3. Design of Experiments

3.1. Introduction

In Chapter 2, we assumed that the sample locations are fixed and studied how to fit a good polynomial response surface (PRS) through the given samples. We showed that different basis functions can affect the accuracy of the prediction, which was evaluated using prediction variance. We also studied a method of eliminating unimportant basis functions to improve the accuracy of the prediction. However, the choice of sample locations where experiments (whether numerical or physical) are performed has very large effects on the quality of the surrogate. A partial result was shown in **Example 2-10** and Exercise Problem 28 of Chapter 2, where the prediction variance changes significantly depending on the locations of the samples. As the purpose of surrogate modeling is to approximate a quantity of interest (QoI) using a set of samples, the goal is to make the surrogate model as accurate as possible using as few samples as possible. To achieve this goal, the first important task is to determine how many samples to use and where to locate the samples.

In this chapter, we will explore methods for selecting a good set of sample locations for carrying out experiments. The selection of these sample locations is known as the design of experiments (DoE) or experimental designs. This terminology was developed because the PRS was initially designed to approximate the QoI from experiments. It is generally accepted that DoE is as important as the surrogate itself because the accuracy of the surrogate depends on the number and location of samples. In addition, there is no single DoE scheme that is the best for all surrogates. Various DoE techniques cater to different sources of errors, in particular, errors due to noise in samples or errors due to an improper surrogate model.

Although DoE is used to find sample locations to build a surrogate in this text, DoE was originally developed in agricultural applications to identify the effect of input variables (factors) on the output QoI [22]. DoE is defined as a branch of applied statistics that deals with planning, conducting, analyzing, and interpreting controlled tests to evaluate the factors (i.e., input variables) that control the value of a parameter or group of parameters. It is used (a) to determine if input variables have an effect on QoIs, (b) to determine if multiple inputs interact in their effect on the QoIs, (c) to model the behavior of the QoI as a function of input variables, and (d) to optimize the QoI. It is noted that DoE is used for defining sample locations for surrogate modeling in this text.

It would be beneficial to understand common DoE terms and concepts before we study detailed DoE methods. The most commonly used terms in the DoE include input variables, uncontrollable parameters, and output QoI. Figure 3-1 illustrates the relationship between these terms. Input variables are those variables that can be varied in experiments or numerical simulations (the process in the figure). In DoE literature, input variables are referred to as factors. It is assumed that input variables are deterministic and the users have full control of them. Input variables include dimensions of test specimens, applied loads, mass, time, etc. Output QoI is the response of the model or measurement from the experiment. It is assumed that the output QoI varies depending on input variables. Since the QoI is the output from an experiment or simulation, it may include measurement error or simulation error. It is also assumed that the output QoI is a continuous and smooth function of input variables. Uncontrollable parameters are those parameters that cannot be controlled by the user, such as ambient temperature during the experiment. In practice, variation of these parameters during the process can cause random noise in the output QoI. Even if these parameters affect the output QoI, their contribution is not modeled as a functional relationship in the surrogate. The process represents physical experiments or numerical simulations that can produce output QoI for given input variables and uncontrollable parameters. In some sense, the process is

considered a black box to the user. That means the user does not need to know the complex process within the experiment or simulation. As long as the user can provide different input variables and obtain output QoI, the surrogate model can be generated. In this sense, the surrogate model is considered a data-driven approach as it can be applied to any physical model or process. The main purpose of DoE is to determine input variables such that their effects on the output QoI can be identified in the best way.

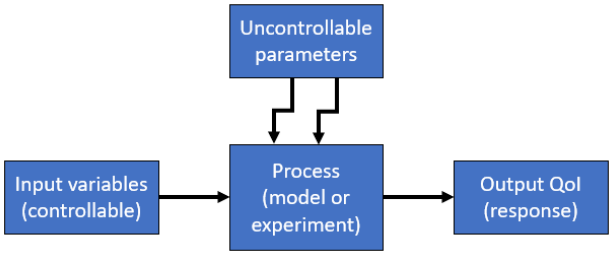


Figure 3-1: The effect of input variables on the quantity of interest in a numerical model or experiment.

As mentioned before, in this textbook, DoE is used for the purpose of generating sample locations to build a surrogate. In that sense, DoE is inherently a multi-objective optimization problem. We would like to select points so that we maximize the accuracy of the information that we get from the experiments. As shown in Chapter 2, however, the accuracy can be improved by increasing the number of samples as well. On the other hand, we also would like to minimize the number of experiments, because they are expensive. In some cases, the objective of the experiments is to estimate some physical characteristics, and in these cases, we would like to maximize the accuracy of these characteristics. However, in the design applications, which are of primary interest in this textbook, we would like to construct a surrogate that could be used to predict the performance at unsampled locations. In this case, our primary goal is to choose the points for the experiments to maximize the predictive capability of the model. Therefore, we focus on how to reduce/minimize the prediction variance at unsampled locations.

Samples do not need to be generated simultaneously. Initially starting from a small number of samples, additional samples can be added sequentially to improve the accuracy of surrogate prediction. The strategy of adding additional samples is called adaptive sampling, sequential design, or active learning. Usually, the adaptive sampling strategies require a criterion to determine the next sample locations. Many criteria have been proposed, such as D-optimal design and minimum bias design. These criteria are often based on a trade-off between minimizing bias (i.e., error) and/or minimizing variance (i.e., uncertainty).

When a surrogate model is flexible enough, a better surrogate can be obtained with more samples. With a fixed size of design space, the distance between samples decreases as the number of samples increases. Therefore, the meaning of the number of samples is equivalent to the distance between the samples. It is easy to imagine that if the distance between samples is short, the surrogate approximate can be accurate because the samples can capture the trend of functional behavior well. At the same time, for a fixed number of samples, it is better to locate the samples such that they are more or less uniformly distributed to the entire sampling space, which is called ‘space-filling design’. In order to have a space-filling design, some sampling strategies minimize the maximum distance between samples. Or, some strategies uniformly divide the sampling space and put a sample in each segment. Section 3.4 introduces two important space-filling DoEs: Latin hypercube sampling and orthogonal arrays.

A lot of work has been done on experimental designs in regular design domains. Such domains occur when each design variable is bounded by simple lower and upper limits so that the design domain is box-like. Occasionally, spherical domains are also considered. Sometimes each design variable can take only

two or three values, often called levels. These levels are termed low, nominal and high. In other cases, the design space is approximately box-like, but it is possible to carry out experiments with the design variables taking values outside the box for the purpose of improving the properties of the surrogate. In Section 3.2, we will summarize briefly some of the properties of experimental design in box-like domains, and present some of the more popular experimental designs in such domains. The readers are referred to Myers and Montgomery (1995) [4] for the in-depth analysis of this subject.

For design optimization, however, it is common for us to try and create surrogates in irregularly shaped domains. In that case, we have to create our own experimental design. Section 3.3 will discuss several techniques available for finding good designs in a generally shaped domain. These strategies mostly use an optimization technique based on different criteria. Since additional sample locations can be found based on pre-existing sample locations, these strategies are good for an adaptive sampling scheme.

3.2. Design of experiments in boxlike domains

Scaling of input variables

When the design space of input variables is given in a box-like shape, the design space is defined by simple lower and upper limits on each of the input variables

$$x_i^l \le x_i' \le x_i^u, \qquad i = 1, \cdots, n \tag{3.1}$$

where x_i^l and x_i^u are the lower and upper bounds, respectively, of the input variable x_i'. The prime indicates that the design variable has not been normalized. For convenience, we scale the design variable as

$$x_i = \frac{2x_i' - x_i^l - x_i^u}{x_i^u - x_i^l} \tag{3.2}$$

The normalized variables are then all bound in the cube

$$-1 \le x_i \le 1, \qquad i = 1, \cdots, n \tag{3.3}$$

Such normalization is useful when different input variables have different dimensions with different orders of magnitude. For example, the thickness of a plate may vary in $x_1' \in [0.5, 1.5] \times 10^{-3} m$, while Young's modulus of the material may vary in $x_2' \in [190, 210] \times 10^9 Pa$. Therefore, using the original variables can cause numerically ill-conditioning of the moment matrix during the regression process. The normalization in Eq. (3.2) should not affect the PRS with linear regression because of the linear relationship between the original and normalized variables.

Although the main purpose of surrogate models in this text is optimization, they are also frequently used for uncertainty quantification, where input variables follow a specific probability distribution, and the probability distribution of output QoI needs to be evaluated. In such a case, it would be better to scale the input variables based on their distribution. For example, when the user wants the design space to cover six times the standard deviation, input variables can be scaled by

$$x_i = \frac{x_i' - \mu_{x_i}}{\sigma_{x_i}} \tag{3.4}$$

where μ_{x_i} and σ_{x_i} are, respectively, the mean and standard deviation of the input variable x_i'. Then, the scaled variables are all bound in the cube

$$-6 \le x_i \le 6, \qquad i = 1, \cdots, n \tag{3.5}$$

When the mean and standard deviation of input variables are not available, the mean and standard deviation of samples can also be used as an approximation.

It is also possible that the output QoI can be normalized in a similar way to the input variables. Unfortunately, the output QoI is not what the user can control, it is not possible to establish the lower and upper bounds of the output QoI. Instead, the minimum and maximum of samples can be used to normalize the output QoI, as

$$y_i = \frac{y_i' - y_{min}}{y_{max} - y_{min}} \tag{3.6}$$

where y_{min} and y_{max} are, respectively, the minimum and maximum samples.

Example 3-1

Consider the quadratic PRS in **Example 2-5** with five samples. Normalize the input variable based on Eq. (3.2) and the QoI based on Eq. (3.6) and show that the resulting surrogate is identical to the one given in **Example 2-5** after conversion.

Solution:

With the original samples before scaling, the quadratic PRS was

$$\hat{y}'(x) = -0.1071 + 0.925x' + 0.0536x'^2$$

Note that the notation $'$ is used for the variables in the original domain. Since the lower and upper bounds of input variables are not given in the example, the minimum and maximum values of the samples are used as the two bounds: $x^l = -2, x^u = 2, y_{min} = -1.5, y_{max} = 1.75$. Then the input variables and output QoI are converted to

$$x = \frac{1}{2}x', \qquad y_i = \frac{y_i' + 1.5}{3.25}$$

Then, the five samples are scaled by $(-1,0), (-0.5, 0), (0, 6/13), (0.5, 11/13), (1, 1)$. The PRS is of the form $\hat{y}(x) = b_1 + b_2 x + b_3 x^2$, and we can define the following matrices and vectors for regression:

$$\mathbf{X} = \begin{bmatrix} 1 & -1 & 1 \\ 1 & -0.5 & 0.25 \\ 1 & 0 & 0 \\ 1 & 0.5 & 0.25 \\ 1 & 1 & 1 \end{bmatrix}, \quad \mathbf{y} = \begin{Bmatrix} 0 \\ 0 \\ 6/13 \\ 11/13 \\ 1 \end{Bmatrix}, \quad \mathbf{X}^T\mathbf{X} = \begin{bmatrix} 5 & 0 & 2.5 \\ 0 & 2.5 & 0 \\ 2.5 & 0 & 2.125 \end{bmatrix}, \quad \mathbf{X}^T\mathbf{y} = \begin{Bmatrix} 2.3077 \\ 1.4213 \\ 1.2115 \end{Bmatrix}$$

By solving the regression equation in Eq. (2.14), the unknown regression coefficients are identified as $b_1 = 0.4286, b_2 = 0.5692$, and $b_3 = 0.0659$. Therefore, with the normalized variables, the quadratic PRS can be written as

$$\hat{y}(x) = 0.4286 + 0.5692x + 0.0659x^2$$

Now, after applying the conversion relationship, the above quadratic PRS can be converted into

$$\hat{y}(x) = \frac{\hat{y}'(x) + 1.5}{3.25} = 0.4286 + 0.5692\left(\frac{1}{2}x'\right) + 0.0659\left(\frac{1}{2}x'\right)^2$$

$$\hat{y}'(x) = -0.1071 + 0.925x' + 0.0536x'^2$$

Therefore, the same surrogate model is obtained after converting it to the original variables.

Interpolation, extrapolation, and prediction variance

The simplest experimental design for the cube of design space is one experiment at each one of the 2^n vertices. This design is called a two-level full factorial design, where the word 'factorial' refers to 'factor', a synonym for design variable, rather than the factorial function. This means that two samples are generated in each design variable at its minimum and maximum. Such two-level DoE is good when the functional relationship between the design variables and the QoI is linear with possible interactions (i.e., including $x_i x_j, i \neq j$ terms). If three samples are used in each design variable, it is called a three-level. A three-level full factorial design would require 3^n numbers of samples. A three-level DoE is good when the functional relationship is quadratic.

For a small number of design variables (a low dimensional problem), the number of samples 2^n or 3^n can be a manageable number of experiments. However, for larger values of n, we usually cannot afford the full factorial design. For example, for $n = 10$, the number of samples becomes $2^n = 1{,}024$ for two-level, and $3^n = 59{,}049$ for three-level DoE. Therefore, the number of samples increases exponentially proportional to the number of dimensions, which is called the 'curse of dimensionality.' This is considered a major bottleneck of surrogate modeling for applications with many input variables. Therefore, full factorial design is limited only to low-dimensional problems ($n \leq 4$). For high values of n, it may be necessary to consider fractional factorial designs, which do not include all the vertices. Different DoE methods are available depending on how to choose sample locations in fractional factorial designs.

If we want to fit a linear PRS to samples, it certainly appears that we will not need anywhere near 2^n samples for a good fit. For example, for $n = 10$, the linear PRS has 11 unknown coefficients to fit, and using $1{,}024$ experiments to fit these 11 coefficients may appear excessive even if we could afford that many experiments. Theoretically, 11 samples should be good enough if we know that the functional relationship is linear and samples do not include measurement noise or error. In practice, however, we do not know the exact functional relationship, and therefore, we need more samples than the number of unknown coefficients. In general, it is acceptable that the number of samples is about two or three times more than that of the unknown coefficients.

However, with fewer samples, we lose an important property of using the surrogate as an interpolation tool rather than as a tool for extrapolation. To understand that, we will first define what we mean by interpolation and extrapolation. Intuitively, we say that a surrogate will interpolate samples at a point if that point is 'completely surrounded' by sample points. In one-dimensional space, as shown in Figure 3-2(a), this means that there is a sample to the right of the interpolated point, as well as a sample to the left of it. In two-dimensional space, as shown in Figure 3-2(b), we would like the point to be surrounded by at least three samples so that it falls inside the triangle defined by these three samples. Similarly, in three-dimensional space, we would like the point to be surrounded by at least four samples; that is, the point lies inside the tetrahedron defined by these four samples. In n-dimensional space, we would like the point to be surrounded by $n + 1$ sample points, or in other words, fit inside the simplex defined by the $n + 1$ samples. A simplex is the generalization of a triangle and a tetrahedron; a shape in n-dimensional space defined by linearly $n + 1$ independent points.

Given a set of $n + 1$ points in n-dimensional space, $x_1, x_2, \cdots, x_{n+1}$, the simplex defined by these points includes all the points that can be obtained by a convex sum of these points. That is, it includes any point $\mathbf{x}$, which may be written as

$$\mathbf{x} = \sum_{i=1}^{n+1} \alpha_i \mathbf{x}_i \tag{3.7}$$

with

$$\sum_{i=1}^{n+1} \alpha_i = 1 \tag{3.8}$$

and $\alpha_i \geq 0, i = 1, \cdots, n+1$. Given a set of n_y sample points, the set of points where the surrogate performs interpolation is the union of these simplices, which is called the convex hull of the sample points.

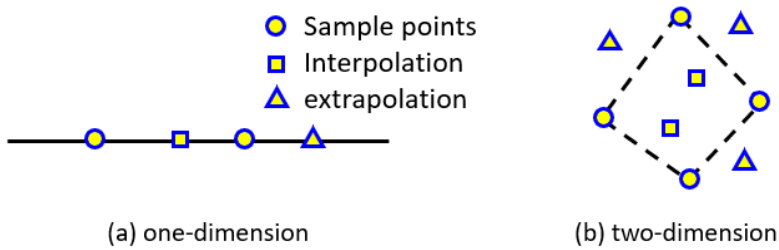


Figure 3-2: Interpolation versus extrapolation in one- and two-dimensional design space.

In general, surrogate prediction is more accurate in the interpolation region than in the extrapolation region. As shown in **Example 2-10**, when three samples were used, the standard error of prediction at the extrapolated point $(1,1)$ was highest $\sigma_y = \sqrt{3}\hat{\sigma}$. On the other hand, the centroid of the three samples has the lowest standard error $\sigma_y = \hat{\sigma}/\sqrt{3}$. Therefore, it is best to locate samples such that the convex hull of samples can cover the design space as much as possible. However, as shown in Figure 1-5, the volume of the convex hull of samples becomes very small compared to that of the design space as the dimension of the design space increases. We showed that in 10-dimensional space if two samples were generated at the center of each orthant ($x_i = -0.5, 0.5$ in the normalized space), 1,024 samples from the two-level full factorial design can only cover 0.1% volume of the design space. Therefore, most prediction points belong to the extrapolation region, where the prediction accuracy of the surrogate deteriorates quickly.

One may argue that what if we generate samples at all vertices ($x_i = -1.0, 1.0$ in the normalized space) instead of the center of each orthant. In such a case, the entire design space becomes the interpolation region. However, in such a choice of sample locations, the distance between samples increases rapidly. In two-dimensional space, if all four samples are located at vertices, the maximum distance between samples is $2\sqrt{2}$. In three-dimensional space, it becomes $2\sqrt{3}$. Therefore, the distance between samples increases along with the dimension of the design space. Even if surrogate prediction in the interpolation is better than in the extrapolation region, the prediction deteriorates quickly as the distance between samples increases. Instead of discussing the accuracy of surrogate prediction qualitatively, it would be necessary to define a quantitative measure to assess the quality of prediction accuracy of a surrogate.

One measure that we can use to estimate the loss of prediction accuracy incurred when we use extrapolation is the prediction variance. In the case of PRS, we developed the prediction variance in Chapter 2. In summary, the PRS assumes that (a) the true function is described by a linear combination of monomials with unknown coefficients, (b) samples are obtained by adding random noise to the true function, where the noise follows a Gaussian distribution$\sim N(0, \sigma^2)$, and (c) noises at different sample points have the same standard deviation and are not correlated. Under these assumptions, the noise standard deviation (standard error of noise) was estimated by

$$\hat{\sigma} = \sqrt{\frac{\mathbf{e}^T\mathbf{e}}{n_y - n_\beta}} \tag{3.9}$$

where n_y and n_β are, respectively, the number of samples and the number of model parameters, and $e_i = y_i - \hat{y}_i$ is the residual at ith sample location.

Recall that the linear regression PRS model that we used in Eq. (2.10) can be written as $\hat{y}(\mathbf{x}) = \boldsymbol{\xi}(\mathbf{x})^T\mathbf{b}$, where $\boldsymbol{\xi}(\mathbf{x})$ is the vector of monomial basis functions and $\mathbf{b}$ is the vector of regression coefficients. With noise in the samples, $\mathbf{b}$ has some uncertainty to it, while $\boldsymbol{\xi}(\mathbf{x})$ is deterministic. At prediction point $\mathbf{x}$, the prediction variance is given as $V[\hat{y}(\mathbf{x})] = \hat{\sigma}^2\boldsymbol{\xi}(\mathbf{x})^T(\mathbf{X}^T\mathbf{X})^{-1}\boldsymbol{\xi}(\mathbf{x})$. The square root of the prediction variance is called the standard error of prediction, defined as

$$\sigma_y(\mathbf{x}) = \hat{\sigma}\sqrt{\boldsymbol{\xi}(\mathbf{x})^T(\mathbf{X}^T\mathbf{X})^{-1}\boldsymbol{\xi}(\mathbf{x})} \tag{3.10}$$

The standard error gives us an estimate of the sensitivity of the surrogate prediction at different points. From a statistical perspective, when the surrogate prediction at unsampled point $\mathbf{x}$ is $\hat{y}(\mathbf{x})$, the true function at that point is unknown (i.e., uncertain) and the probability of the true function can be given as $\sim N(\hat{y}, \sigma_y^2)$. In the extreme case, if $\sigma_y(\mathbf{x}) = 0$ at a point, it means the surrogate prediction $\hat{y}(\mathbf{x})$ is exact and there is no uncertainty. Therefore, a surrogate prediction is considered accurate when $\sigma_y(\mathbf{x})$ is small. Note that $\sigma_y(\mathbf{x})$ varies at different locations.

The main goal of DoE is that we would like to select sample locations to make the prediction variance as small as possible anywhere in the domain where we would like to estimate the QoI. Intuitively, it appears that this would be helped if the standard error did not vary much from one point to another. This property is called stability, which is defined as the ratio, $\sigma_y^{max}/\sigma_y^{min}$. The following example demonstrates the effect of using extrapolation on the stability of the standard error.

Example 3-2

Consider the problem of fitting a linear PRS $\hat{y}(\mathbf{x}) = b_1 + b_2x_1 + b_3x_2$ to samples in the square domain $-1 \le x_1, x_2 \le 1$. Compare the maximum value of the prediction variance for two cases: (a) a full factorial design (samples at all four vertices), and (b) a fractional factorial design including three vertices, obtained by omitting the vertex $(1,1)$.

Solution:

Full factorial design: We number the four sample locations as $\mathbf{x}_1 = [-1,-1]^T$, $\mathbf{x}_2 = [-1,1]^T$, $\mathbf{x}_3 = [1,-1]^T$, and $\mathbf{x}_4 = [1,1]^T$. For this case, the moment matrix can be obtained as

$$\boldsymbol{\xi}(\mathbf{x}) = \begin{Bmatrix} 1 \\ x_1 \\ x_2 \end{Bmatrix},\ \mathbf{X} = \begin{bmatrix} 1 & -1 & -1 \\ 1 & -1 & 1 \\ 1 & 1 & -1 \\ 1 & 1 & 1 \end{bmatrix},\ \mathbf{X}^T\mathbf{X} = \begin{bmatrix} 4 & 0 & 0 \\ 0 & 4 & 0 \\ 0 & 0 & 4 \end{bmatrix},\ (\mathbf{X}^T\mathbf{X})^{-1} = \frac{1}{4}\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

The standard error of prediction in Eq. (3.10) is given as

$$\sigma_y = \hat{\sigma}\sqrt{\boldsymbol{\xi}^T(\mathbf{X}^T\mathbf{X})^{-1}\boldsymbol{\xi}} = \hat{\sigma}\sqrt{0.25(1 + x_1^2 + x_2^2)}$$

Therefore, the minimum standard error occurs at the origin ($x_1 = x_2 = 0$), $\sigma_y = \hat{\sigma}/2$, while the maximum occurs at the vertices, $\sigma_y = \sqrt{3}\hat{\sigma}/2$. This case represents a fairly stable variation of the standard error of only $\sqrt{3}$ between the smallest and highest value in the domain of interest.

Fractional factorial design: Without sample at (1, 1), we have

$$\mathbf{X}=\begin{bmatrix}1 & -1 & -1\\ 1 & -1 & 1\\ 1 & 1 & -1\end{bmatrix},\ \mathbf{X}^T\mathbf{X}=\begin{bmatrix}3 & -1 & -1\\ -1 & 3 & -1\\ -1 & -1 & 3\end{bmatrix},\ (\mathbf{X}^T\mathbf{X})^{-1}=\frac{1}{4}\begin{bmatrix}2 & 1 & 1\\ 1 & 2 & 1\\ 1 & 1 & 2\end{bmatrix}$$

Therefore, the standard error of prediction is given as

$$\sigma_y=\hat{\sigma}\sqrt{0.5(1+x_1+x_2+x_1^2+x_1x_2+x_2^2)}$$

At the origin, the standard error is $\sigma_y=\hat{\sigma}/\sqrt{2}$, which is increased from the full factorial design. At the three vertices with samples, the standard error of prediction is the same as the standard error of noise: $\sigma_y=\hat{\sigma}$. This is expected, with only three sample points, the PRS passes through the samples so that the error at the sample points should be the same as the measurement error. Therefore, the standard error increases slightly at the origin and the sample locations. However, at the fourth vertex $(1,1)$, where the PRS represents extrapolation, the standard error becomes $\sigma_y=\sqrt{3}\hat{\sigma}$, much increased from the case of full factorial design. This is because this is the farthest extrapolation point from the convex hull of the three samples. By setting the derivatives of the prediction error to zero, we easily find that the minimum error is at the centroid of the three sample points, at $(-1/3,-1/3)$. At the centroid $\sigma_y=\hat{\sigma}/\sqrt{3}$. Now the ratio between the smallest and highest standard errors is 3, with the highest errors in the region of extrapolation. Therefore, the full factorial design is more stable than the fractional factorial design.

It can be checked that when we use a full factorial design for a linear PRS with n variables, the moment matrix becomes diagonal, $\mathbf{X}^T\mathbf{X}=2^n\mathbf{I}$, where $\mathbf{I}$ is a unit matrix of order $n+1$. Therefore, the standard error of prediction becomes

$$\sigma_y=\frac{\hat{\sigma}}{2^{n/2}}\sqrt{1+x_1^2+x_2^2+\cdots,x_n^2} \tag{3.11}$$

In such a case, the maximum prediction error (achieved at any vertex) is $\hat{\sigma}\sqrt{(n+1)/2^n}$. That is, the quality of the fit becomes very good with increasing n. This reflects the fact that we use 2^n points to calculate $n+1$ coefficients so that we filter out the effect of noise. This estimate is misleading in actual situations, however, because rarely do we have a true linear model. When the response we measure is not linear, we will have modeling errors, also called bias errors which are not averaged out.

On the other hand, if the number of samples becomes the same as $n+1$ unknown coefficients, it is called a saturated design. In that case, the standard error progressively increases along with the dimension n, because the portion of design space covered by the simplex containing the sample points becomes progressively smaller. For example, in **Example 3-2** the three points used for the saturated design form a triangle covering half of the design domain. For a three-dimensional cube, four vertices will span a tetrahedron with a volume of one-sixth of the volume of the enclosing cube. For the n dimensional case, the full-factorial design is the vertices of a cube of volume 2^n, while $n+1$ vertices obtained by perturbing one variable at a time from one vertex span a simplex of volume $2^2/n!$. Therefore, the fraction of the extrapolation region rapidly increases with the dimension. As the extrapolation region increases so does the prediction variance. For example, for a three-dimension design space, the maximum prediction error with a full-factorial design is $\sqrt{0.5}\sigma$, while the maximum prediction error for the saturated four-point fractional factorial design is $\sqrt{7}\sigma$ (see Exercise Problem 3).

Designs for linear polynomial response surfaces

For fitting linear PRS, we typically use designs with only two levels for each design variable, and the most popular fractional designs are the so-called orthogonal designs. An orthogonal design means that

different coefficients in linear regression are uncorrelated; that is, $CoV(b_i, b_j) = 0$. It means that the uncertainty in one coefficient is not related to the uncertainty in other coefficients, which is a desirable property as the effect of different basis functions can be determined individually. An orthogonal design is one where the moment matrix $\mathbf{X}^T\mathbf{X}$ is diagonal, which is equivalent to the inner product of two different columns of the design matrix being zero, which means they are orthogonal. Since each column of the design matrix represents the basis function, the orthogonality means that the coefficient of each basis function can be estimated independently from other coefficients. The popularity of orthogonal designs is partly based on the following theorem (see Myers and Montgomery, 1995 p. 284 [4]):

> *For the first-order model (linear PRS) and fixed sample size, if all variables lie between* -1 *and* 1*, then the variance of the coefficients is minimized if the design is orthogonal, and all the variables are at their outer positive or negative limits (i.e.,* -1 *or* $+1$*).*

It is easy to check that the full factorial design is orthogonal, but it is not trivial to produce orthogonal designs with a smaller number of samples. Various orthogonal designs can be found in books on the design of experiments (see Myers and Montgomery, 1995 [4]). Important orthogonal designs will be presented as a part of the orthogonal array in Section 3.4. To demonstrate the beneficial properties of orthogonal designs, we will consider the two-dimensional case that we have studied in **Example 3-2**. In that example, we fitted a two-variable linear PRS first with a full factorial design (four samples at corners) and then with only three samples (simplex in two-dimension). Fitting a linear PRS in n variables on the basis of $n + 1$ points requires the points to be linearly independent, so that they form a simplex. There is no redundancy in the design, in that the number of points is equal to the number of coefficients, and this is called a saturated design. In order to get an orthogonal design with three samples in two-dimensional space, we have to give up on having the variables only at the corners. Instead, a perfect simplex is used, where the distances between all points are the same.

Example 3-3

Consider the equilateral triangle which results in a scalar matrix (a scalar matrix is a scalar multiple of the unit matrix) $\mathbf{X}^T\mathbf{X}$. It includes the points $(\sqrt{3/2}, -1/\sqrt{2})$, $(-\sqrt{3/2}, -1/\sqrt{2})$, $(0, \sqrt{2})$. Check for the stability of the prediction variance and its maximum value in the unit square for the linear model $\hat{y}(\mathbf{x}) = b_1 + b_2x_1 + b_3x_2$.

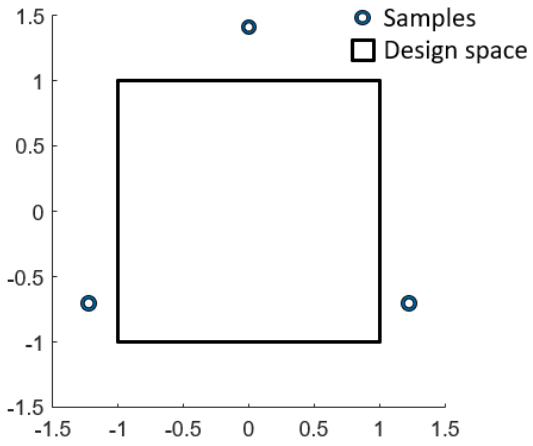


Figure 3-3: Design space and sample locations for **Example 3-3**.

Solution:
First, it is noted that the three samples are the corners of an equilateral triangle whose centroid is the same as that of the design space; i.e., the origin. In order to calculate the standard error of prediction in Eq. (3.10), the inverse of the moment matrix needs to be calculated first using the three samples.

$$\mathbf{X} = \begin{bmatrix} 1 & \sqrt{3/2} & -1/\sqrt{2} \\ 1 & -\sqrt{3/2} & -1/\sqrt{2} \\ 1 & 0 & \sqrt{2} \end{bmatrix}, \quad \mathbf{X}^T\mathbf{X} = \begin{bmatrix} 3 & 0 & 0 \\ 0 & 3 & 0 \\ 0 & 0 & 3 \end{bmatrix}, \quad (\mathbf{X}^T\mathbf{X})^{-1} = \frac{1}{3}\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Therefore, the three samples proved to be an orthogonal design. Then, the standard error of prediction in Eq. (3.10) becomes

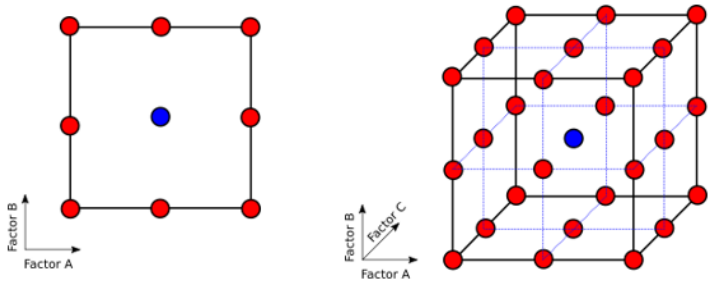
$$\sigma_y(\mathbf{x}) = \hat{\sigma}\sqrt{(1 + x_1^2 + x_2^2)/3}$$

Therefore, the standard error of prediction becomes $\sigma_y = \hat{\sigma}/\sqrt{3}$ at the origin and $\sigma_y = \hat{\sigma}$ at the vertices. That is, the stability ratio is $\sigma_y^{max}/\sigma_y^{min} = \sqrt{3}$, which is an improvement from 3 in **Example 3-2**. Also, σ_y^{max} is reduced from $\sqrt{3}\hat{\sigma}$ to $\hat{\sigma}$. However, this has come at the price of obtaining the samples outside the unit square as shown in Figure 3-3.

As will be shown later, this reduces the variance error but increases the so-called bias error. A bias error is the error introduced when the model that we try to fit is different from the true function. For example, if the model is linear and the true function is quadratic, the simplex model that we have used in this example is likely to increase the error rather than decrease it.

Designs for quadratic polynomial response surfaces
Quadratic PRS with n variables have $n_\beta = (n+1)(n+2)/2$ coefficients as shown in Eq. (2.16). To fit quadratic PRS, we need at least that many samples, and at least three levels for each design variable in order to capture a quadratic functional change. For $n > 3$ it is possible to have the requisite number of samples with only two levels. For example, a quadratic PRS with four variables has 15 coefficients, and a full factorial design in two levels has $2^4 = 16$ points. Therefore, theoretically, it is possible to identify unknown coefficients with a two-level full factorial design. However, if we let only one design variable vary at a time, we can easily check that we are left with three coefficients and we need three different levels of that design variable. That is, a quadratic PRS with $n = 2$ can be defined as $\hat{y}(x_1, x_2) = b_1 + b_2x_1 + b_3x_2 + b_4x_1^2 + b_5x_1x_2 + b_6x_2^2$. When x_1 is fixed, the PRS becomes one-dimension and can be simplified as $\hat{y}(x_2) = \tilde{b}_1 + \tilde{b}_2x_2 + \tilde{b}_3x_2^2$ with three coefficients. Therefore, it would make a sense to have three levels in each design variable.

We can use a full-factorial three-level design for a quadratic PRS, which will have 3^n samples, as shown in Figure 3-4. In addition to samples in all vertices (either -1 or $+1$ location in the normalized design space), the full-factorial three-level design has a sample at the center of each design variable (i.e., 0 location). In most cases, however, we cannot afford such many samples even for fairly small values of n. For example, for $n = 6$, we require $3^6 = 729$ samples, which is too many to identify $n_\beta = 28$ coefficients. In general, in order to have a stable estimation, the number of samples needs to be two or three times more than that of the unknown coefficients. Therefore, 80 samples should be good enough instead of 729. Many alternative DoEs are proposed that can still capture a quadratic change of the function with a much smaller number of samples than the full-factorial design.



(a) Two-dimension (b) Three-dimension

Figure 3-4: Sample locations for full factorial designs.

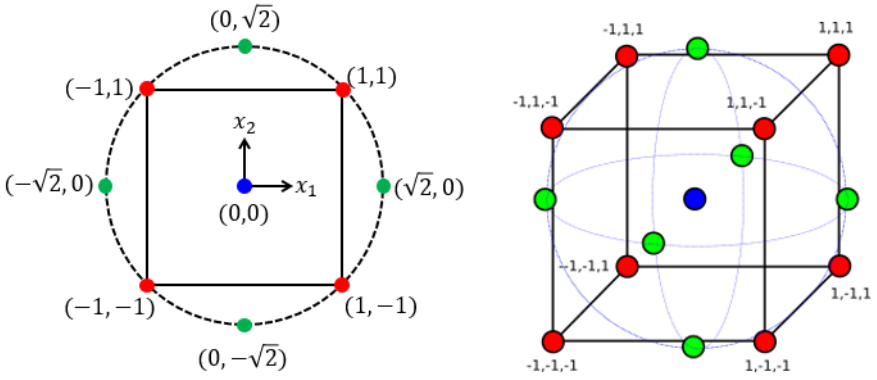
A popular compromise that reduces the number of samples to close to the two-level full factorial design is the central composite design (CCD). The CCD is a fractional factorial design with center points, augmented with a group of axial points that allows for estimating curvature. The CCD is composed of the 2^n points of the full-factorial two-level design, with all the variables at their extremes, plus a number of repetitions n_c at the center of design space, and the $2n$ points obtained by changing one design variable at a time by a distance $\alpha \geq 1$, which are referred to as axial points. Therefore, the total number of samples for CCD is $n_y = 2^n + 2n + n_c$. In determining the range of design space, the center point is often close to or near the local optimum. The replicate center points are used to test if the curvature near the design point is significant, which is important for design exploration. For this reason, the two-level factorial design is augmented with the design center to capture the second-order behavior of QoI. In addition to the center points, it is necessary to have some axial points in order to construct the second-order model and to estimate the parameters related to each second-order term. Without the axial points, the variation due to the second-order terms cannot be orthogonally decomposed into the variations of each second-order term. If quadratic coefficients b_{ii} turn out to be all negative or positive, then we can ensure that the center point is in the vicinity of the local optimum. If the center point is the current operating condition, the estimates of the second-order term give information on whether the region of exploration is close to the local optimum (maximum or minimum). The unique feature of the CCD is that the $2n$ axial samples at the distance $\alpha \geq 1$ are out of the design space. This can be a limitation of the CCD because, in some situations, we may not allow generating samples out of the design space.

Figure 3-5 shows the central composite design for $n = 2$ and $n = 3$. The value of α chosen in the figures are such that all the points outside the origin are of the same distance from the origin so that we have a spherical design. This placement of the points is at the higher end of the typical choice for $\alpha = \sqrt{n}$. A more popular choice is based on the concept of rotatability. The property of rotatability requires that the prediction variance is dependent only on the distance from the origin and not on the orientation with respect to the coordinate axes [23].

It can be shown that for the central composite design, the rotatability requirement will be satisfied for

$$\alpha = 2^{n/4} \tag{3.12}$$

This equation gives $\alpha = \sqrt{2}$ for $n = 2$, which is the same as the spherical design, however, for $n = 3$ we get $\alpha = 1.682$, which is slightly smaller than the equal-distance of $\sqrt{3}$. According to Myers and Montgomery (2002) [4], it is not necessary to have exact rotatability in a second-order design in the practical sense.



(a) Two-dimension (b) Three-dimension

Figure 3-5: Central composite designs for two- and three-dimensional design spaces.

Example 3-4

When input variables are temperature and pressure in the injection-molding process of plastic material, generate a table of input variables using the central composite design. The temperature varies $\in [190°, 210°]$, while the pressure varies $\in [50\text{MPa}, 100\text{MPa}]$. Use $n_c = 5$.

Solution:

Including four vertices, four faces, and five repetitions at the center, the total number of samples is 13. The following table shows sample locations. First, 13 sample locations are determined in the normalized design space, and then, they are transformed to the original design space using the inverse relationship of Eq. (3.2), as

$$x_i' = \frac{1}{2}(1 + x_i)x_i^u + \frac{1}{2}(1 - x_i)x_i^l$$

Sample ID	Temperature	Pressure
1	190°	50MPa
2	210°	50MPa
3	210°	100MPa
4	190°	100MPa
5	185.9°	75MPa
6	214.1°	75MPa
7	200°	39.6MPa
8	200°	110.4MPa
9, 10, 11, 12, 13	200°	75MPa

It may sound like we waste many samples by using n_c repeated samples at the origin. The reason that repeated samples are used at the origin is related to reducing the prediction variance. When we choose sample locations, the objective is to maximize the prediction accuracy; that is, to minimize prediction variance everywhere in the design space. If a single center point is used in the CCD, the prediction variance at the origin is highest. This is unfortunate because normally we choose the center of the design space as the best candidate for optimization, where we need the most accuracy. We want the prediction to

be reliable throughout the region, especially near the center since we hope the optimum is near the center of the design space. By picking five to six center points, the prediction variance at the center is the smallest. At the same time, the prediction variance on the edge is also reduced compared to a single sample at the center. If one or two center points are used in CCD, then the prediction variance at the origin is higher than that of the edges. The prediction variance increases again beyond the edge of the design space. Therefore, the main reason for repeated samples at the origin is to balance the precision at the edge of the design relative to the center. With either a spherical design or a rotatable one, we find that we need a number of replicate center points (points at the origin) to obtain good prediction variance stability.

Example 3-5

In a two-dimensional quadratic PRS with CCD, calculate the minimum and maximum standard error of prediction in Eq. (3.10) by gradually increasing the number of samples at the center from $n_c = 1$ to $n_c = 5$. Plot the contour plots of standard error of prediction and discuss the difference between $n_c = 1$ and $n_c = 5$.

Solution:

With two input variables, the CCD uses four corner points, $(-1,-1), (1,-1), (1,1), (-1,1)$, and four-axial points, $\left(-\sqrt{2},0\right), \left(\sqrt{2},0\right), \left(0,-\sqrt{2}\right), (0,\sqrt{2})$, and n_c repetitions of the center point $(0,0)$. For a quadratic PRS, the moment matrix with these sample locations becomes

$$\mathbf{X}^T\mathbf{X} = \begin{bmatrix} 8+n_c & 0 & 0 & 8 & 0 & 0 \\ 0 & 8 & 0 & 0 & 0 & 0 \\ 0 & 0 & 8 & 0 & 0 & 0 \\ 8 & 0 & 0 & 12 & 0 & 4 \\ 0 & 0 & 0 & 0 & 4 & 0 \\ 8 & 0 & 0 & 4 & 0 & 12 \end{bmatrix}$$

Therefore, the effect of repeated samples at the center occurs at the first diagonal element of the moment matrix. The following Matlab code is used to calculate the standard error of prediction at a 21×21 grid and plot the contours. The users can remove the Matlab comment '%' in the design matrix in order to change the number of repetitions. The code also calculates the maximum and minimum standard errors of prediction along with their ratios (stability).

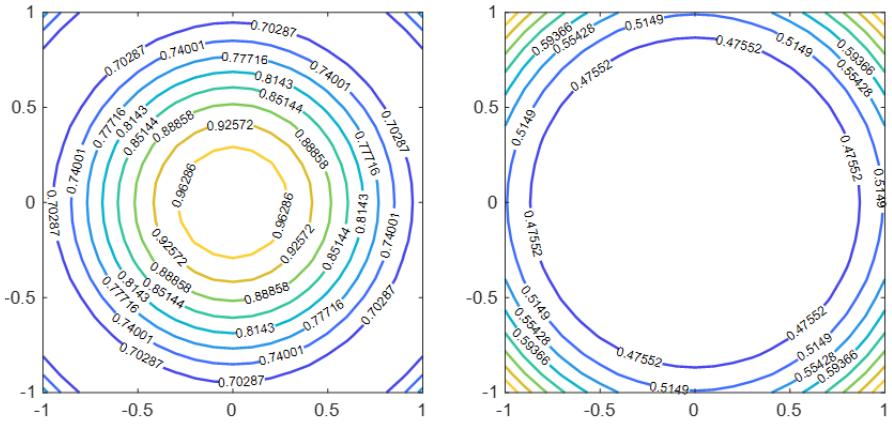
```
a=sqrt(2); b=2;
X=[1 -1 -1  1  1  1;
   1  1 -1  1 -1  1;
   1  1  1  1  1  1;
   1 -1  1  1 -1  1;
   1  a  0  b  0  0;
   1 -a  0  b  0  0;
   1  0  a  0  0  b;
   1  0 -a  0  0  b;
%   1  0  0  0  0  0;
%   1  0  0  0  0  0;
%   1  0  0  0  0  0;
%   1  0  0  0  0  0;
   1  0  0  0  0  0];
XTX=X'*X;
XTXi=inv(XTX);
[X, Y]=meshgrid(-1:.1:1, -1:.1:1);
```

```
Z=zeros(size(X));
[n, m]=size(X);
for i=1:n
    for j=1:m
        xi=[1 X(i,j) Y(i,j) X(i,j).^2 X(i,j).*Y(i,j) Y(i,j).^2];
        Z(i,j)=sqrt(xi*XTXi*xi');
    end
end
v=linspace(min(min(Z)),max(max(Z)),10);
[C,h]=contour(X,Y,Z,v);
clabel(C,h)
Zmax=max(max(Z))
Zmin=min(min(Z))
stability=Zmax/Zmin
```

The following table shows the minimum and maximum of the standard error of prediction along with their ratios.

n_c	σ_y^{min}	σ_y^{max}	$\sigma_y^{max}/\sigma_y^{min}$
1	0.6657	1.0000	1.5021
2	0.5825	0.7906	1.3572
3	0.5216	0.7906	1.5157
4	0.4743	0.7906	1.6667
5	0.4361	0.7906	1.8127

It is interesting to note that the maximum standard error $\sigma_y^{max} = 1.0$ occurs at the center of the design space when $n_c = 1$. When $n_c \geq 2$, the maximum standard error occurs at the corners $\sigma_y^{max} = 0.7906$, and a further increase of n_c does not change the maximum standard error. On the other hand, the increase in n_c reduces the minimum standard error. Therefore, even if the stability ratio $\sigma_y^{max}/\sigma_y^{min}$ increases, the prediction accuracy is improved as the maximum is fixed, while the minimum is reduced. Figure 3-6 shows contour plots of prediction variance with one central point and five central points. It is obvious that more center samples can increase the area of the region that has a low standard error of prediction.



(a) $n_c = 1$ (b) $n_c = 5$

Figure 3-6: Contour plots of $\sigma_y/\hat{\sigma}$ for $n = 2$, $\alpha = \sqrt{2}$, (a) $n_c = 1$ and (b) $n_c = 5$.

The sampling strategy of replicating central points with rotatable CCD can cause a problem with numerical simulations that give exactly the same answer when the simulation is repeated at the same point. This is because the source of error in numerical simulation is modeling error, not random noise. Fortunately, however, with $\alpha = 1$ there is no need for repeated central points. The case of $\alpha = 1$, which is called the face-center central composite design, is very attractive in many applications because it does not require using any other levels except $(-1,0,1)$. Therefore, all samples are within the design space. Figure 3-7 shows the contours of the standard error of prediction for the face-center CCD in a two-dimensional design space. In the case of a two-dimensional space, the face-center CCD becomes identical to a three-level full factorial design. As shown in the figure, the standard error of prediction at the center of the design space is $\sigma_y(0,0) = 0.7454$, while at the four corners $\sigma_y(\pm 1, \pm 1) = 0.8975$, which is the maximum value (see Exercise Problem 11). The minimum standard error of prediction occurs at $\sigma_y\left(\pm\sqrt{0.4}, \pm\sqrt{0.4}\right) = 0.5980$. The stability of face-center CCD is $\sigma_y^{max}/\sigma_y^{min} = 1.50$. Therefore, while the design is not rotatable, the stability is quite good. This performance is better than the CCD with $n_c = 1$ in **Example 3-5** in terms of the maximum standard error of prediction.

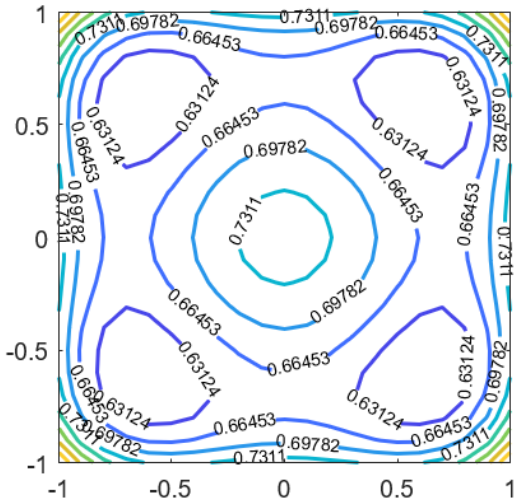


Figure 3-7: Contour plots of $\sigma_y/\hat{\sigma}$ for face-center central composite design for a single center point.

Even if CCD requires a smaller number of samples than the full factorial design, it is still no longer practical for high dimensional design space, because the number of samples increases too fast. CCD is popular for $3 \le n \le 6$, where it gives reasonable ratios between the number of points and the number of coefficients of a quadratic polynomial. For example, when $n = 10$, the number of samples from CCD is $n_y = 2^{10} + 2 \times 10 + n_c = 1{,}044 + n_c$, while the number of unknown coefficients for a quadratic PRS is $n_\beta = 66$. Therefore, $n_y \approx 200$ should be enough. One possible solution is to keep the $2n$ axial samples that perturb a single variable but have a fractional factorial design to replace the 2^n vertices. However, while the number of vertices increases as 2^n, the number of polynomial coefficients increases as $n_\beta = (n+1)(n+2)/2$. Consequently, the type of fractional design used has to be modified as n increases.

There is a block design, first introduced by Box and Behnken (1960) [24], where the number of samples increases at the same rate as the number of polynomial coefficients. The two-variable block design is based on perturbing only two variables from the nominal value. That is, at each point, we have a

pair $(i;\ j)$, such that $|x_i| = 1, |x_j| = 1$, and $x_k = 0$ for all $k \neq i, j$. The two variables are perturbed in all four combinations of ± 1. For example, for $n = 3$ we will have the following sample locations:

x_1	x_2	x_3
-1	-1	0
-1	1	0
1	-1	0
1	1	0
-1	0	-1
-1	0	1
1	0	-1
1	0	1
0	-1	-1
0	-1	1
0	1	-1
0	1	1
0	0	0

where the last point is the central point, which may be repeated. Figure 3-8 shows the sample locations of two-variable block designs for three-dimensional design space. Unfortunately, the block design cannot work for two-dimensional design space because two-dimensional block design yields a total of five samples (four edges points and one center point), while there are six unknown coefficients in a quadratic PRS.

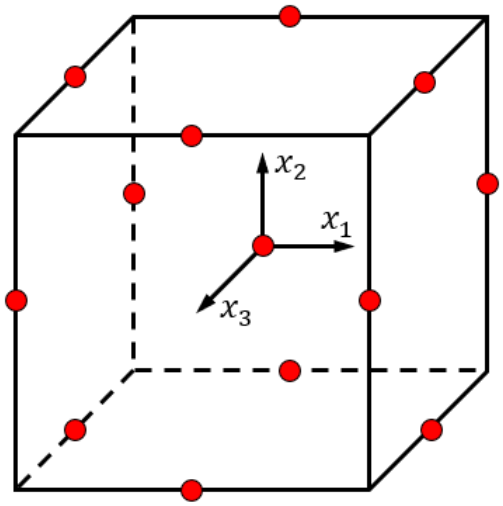


Figure 3-8: Sample locations of two-variable block design for three-dimensional space.

For the general case, we can select two variables out of n in $n(n-1)/2$ ways. For each such combination of two variables, we have four design points, with each one of the variables taking the values of ± 1. Therefore, the total number of points in this block design is $n_y = n_c + 2n(n-1)$. For large values of n this tends asymptotically to be 4 times larger than the number of coefficients that we need to fit. However, this happens for very large values of n. For example, for $n = 10$ the number of coefficients is 66, while the number of points in the block design with one center point is 181 (for comparison, the number of points in the CCD is 1,045). Block designs are spherical, in that all the points are at the same distance from the origin. For example, all the two-variable block design points are at a distance of $\sqrt{2}$ from the origin. For large values of n, this distance can be much smaller than the distance of the vertices (which is $\sqrt{n}$). Therefore, extrapolating to the vertices based on the block design may be risky.

Example 3-6

Although the box design does not work for a quadratic PRS in two-dimension, it is still possible for a linear PRS. Calculate the minimum and maximum standard error of prediction for the two-variable block design in a two-dimensional design space and compare it with that of the full factorial design in **Example 3-2**.

Solution:

In a two-dimensional design space, the sample locations for the block design are $\mathbf{x}_1 = (-1,0), \mathbf{x}_2 = (1,0), \mathbf{x}_3 = (0,-1), \mathbf{x}_4 = (0,1), \mathbf{x}_5 = (0,0)$. Therefore, the design matrix and the moment matrix can be defined as

$$\mathbf{X} = \begin{bmatrix} 1 & 1 & 0 \\ 1 & -1 & 0 \\ 1 & 0 & 1 \\ 1 & 0 & -1 \\ 1 & 0 & 0 \end{bmatrix}, \mathbf{X}^T\mathbf{X} = \begin{bmatrix} 5 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 2 \end{bmatrix}, (\mathbf{X}^T\mathbf{X})^{-1} = \begin{bmatrix} 1/5 & 0 & 0 \\ 0 & 1/2 & 0 \\ 0 & 0 & 1/2 \end{bmatrix}$$

Since the moment matrix is diagonal, the box design yields an orthogonal design. The standard error in Eq. (3.10) can be calculated as

$$\sigma_y(\mathbf{x}) = \hat{\sigma}\sqrt{\frac{1}{5} + \frac{1}{2}(x_1^2 + x_2^2)}$$

The minimum standard error of prediction occurs at the center of the design space $\sigma_y(0,0) = 1/\sqrt{5} = 0.4472$, while the maximum standard error or prediction occurs at the four corner points $\sigma_y(\pm 1, \pm 1) = \sqrt{6/5} = 1.0954$. Therefore, stability becomes $\sigma_y^{max}/\sigma_y^{min} = 2.4495$. Compared to the full factorial design, the block design has a lower minimum standard error, but the maximum standard error increases significantly. This happens because the corner points become an extrapolation region for the block design. Figure 3-9 shows the contour plot of the standard error of prediction for block design. Note that the block design satisfies the property of rotatability, where the prediction variance is dependent only on the distance from the origin and not on the orientation with respect to the coordinate axes.

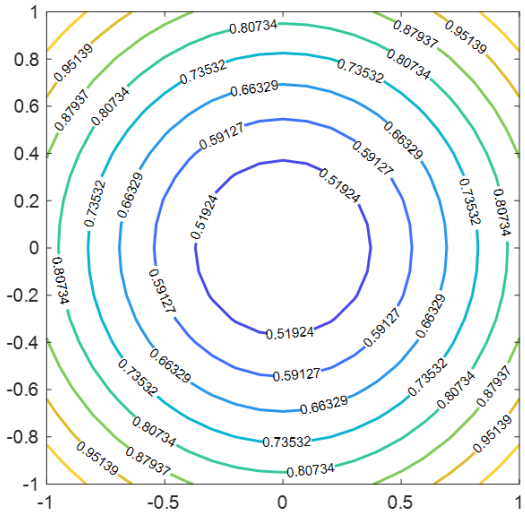


Figure 3-9: Contour plot of the standard error of prediction for block design.

Matlab software provides some useful functions for full and fractional factorial designs. Matlab function `ff2n(n)` generates samples from a two-level full factorial design of n input variables. The output is $n_y \times n$ matrix, where each row represents the sample location in normalized space. The total number of samples will be $n_y = 2 \times 2 \times \cdots \times 2 = 2^n$. Since the two-level full factorial design has sample location only at lower and upper bounds, the matrix is composed of zero (lower bound) and one (upper bound). For a more general full factorial design, the Matlab function `fullfact([l1 … ln])` can be used, where each variable can have different levels. That is, variable x_1 has l_1 levels, x_2 has l_2 levels, etc. The total number of samples will be $n_y = l_1 \times l_2 \times \cdots \times l_n$. The output is $n_y \times n$ matrix, where each row represents the level of samples.

For central composite design, Matlab function `ccdesign(n,'Name',value)` can be used for n design variables with options in `'Name'` and `value`. The number of repeated center points can be specified as `'center'`, n_c. In addition, three different types of CCD can be specified: `'circumscribed'`, `'inscribed'`, and `'faced'`. Figure 3-10 shows these three types of CCD. It is noted that the last option `'faced'` yields face-centered CCD. The following Matlab code can generate samples using CCD and plot them:

```
dCC = ccdesign(2,'type','circumscribed');
plot(dCC(:,1),dCC(:,2),'ro','MarkerFaceColor','b')
X = [1 -1 -1 -1; 1 1 1 -1];
Y = [-1 -1 1 -1; 1 -1 1 1];
line(X,Y,'Color','b')
axis square equal off
```

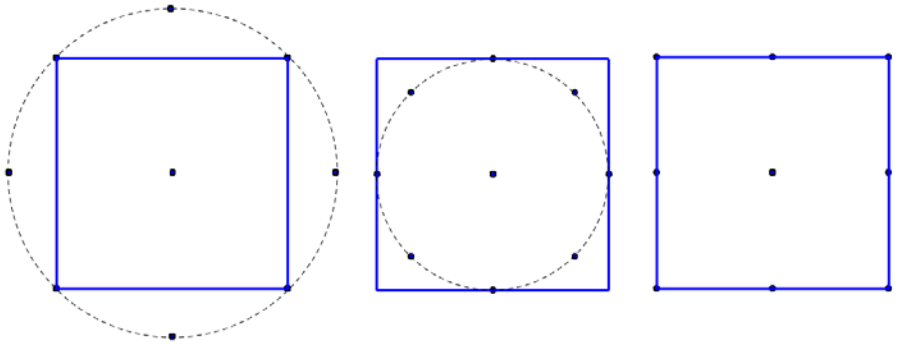


Figure 3-10: Three different central composite designs from Matlab.

For block design, Box-Behnken design [24] is implemented in the Matlab function `bbdesign(n, 'Name', value)`. The number of repeated center points can be specified as `'center'`, n_c. Matlab command `dBB = bbdesign(3)` will produce sample locations using block design as shown in Figure 3-8.

3.3. Optimal design of experiments

The various DoE methods that we considered in the previous section, as well as other DoE methods available in the literature, may perform satisfactorily if the design space is in a box-like domain. After normalizing design variables, the design space will be a square in two-dimension and a cube in three-dimension, etc. When the shape of the design space is fixed, it is possible to pre-determine the number and location of samples that can balance the number of samples and prediction variance. For example, the

full-factorial design is good for lowering prediction variance but requires a large number of samples for high dimensions. On the other hand, the box design can provide a reasonable number of samples for high dimensions but has a high prediction variance because of a large extrapolation region.

For design optimization, however, an optimization problem normally comes with various constraints, which limit the design variables in a certain region, which is referred to as a feasible design. For a given number of samples, the accuracy of PRS is improved as the volume of design space becomes smaller. Therefore, it is desirable to invoke as many constraints as we can to reduce the volume of the design domain. The difficulty is that the standard DoE methods in the previous section cannot be applied to irregular design spaces. Therefore, it would be necessary to develop non-standard DoE methods.

Not only an irregular design space but also a box-like design space may take an advantage of using a non-standard DoE Method. This is because, in the standard DoE methods, the users cannot choose an arbitrary number of samples. Different DoE methods have their own required number of samples. Therefore, if the users want to make a specific number of samples, there may not be any standard DoE methods that allow the same number of samples. In practice, the users may want to choose the best DoE method for a given amount of resources, which is equivalent to a given number of samples. In these cases, the users may have to create their own DoE; that is, to select the best set of sample locations for the given design space and the number of samples.

For a fixed number of samples n_y, non-standard DoE methods basically try to find the locations of samples to satisfy a certain criterion, which is often posed as an optimization problem. That is, the best locations of samples are sought by minimizing a criterion, which represents the quality of the surrogate prediction. Because these types of DoE require solving an optimization problem, they are often referred to as optimal DoE.

Since the optimality criterion of most optimal DoE is based on some function of the moment matrix, the 'optimality' of a given design is model-dependent. That is, the users must specify a model for the design (linear or quadratic PRS, etc.) and the number of samples n_y desired before the 'optimal design' can be generated. Therefore, the generated DoE is 'optimal' only for the given model, the given criterion, and the given number of samples.

The optimization problem of optimal DoE has a considerable challenge in practice. In the case of determining the location of n_y samples in n dimensional space, there are $n \times n_y$ number of variables. Finding an optimum $n \times n_y$ variables can easily be impractical as the number of variables increases quickly. For example, in the case of a six-dimensional design space with 20 samples, there will be 120 variables to optimize. Therefore, traditional optimization algorithms are not a good choice for optimal DoE. Instead, optimal DoE problems are commonly treated as combinatorial problems. A pool of candidate points is defined first, where the QoI can possibly be evaluated. Then, out of the pool, the 'best' n_y points are selected. For example, 3^n samples from the full factorial design can be defined as the pool of candidate points, and then, $n_y = 3 \times n_\beta$ best samples can be selected from them.

As mentioned before, since optimal DoE is an optimization problem, different optimal DoE can be obtained when different criteria are used. In the case of PRS, most of them are related to the moment matrix $\mathbf{X}^T\mathbf{X}$, because linear regression relies on it. The D-optimal design seeks to maximize $|\mathbf{X}^T\mathbf{X}|$, the determinant of the moment matrix. This criterion results in minimizing the variance of the PRS coefficients. The A-optimal design seeks to minimize the trace of the inverse of the moment matrix. This criterion results in minimizing the average variance of the parameter estimates. The G-optimal design seeks to minimize the maximum prediction variance, i.e., minimize the maximum of $\boldsymbol{\xi}(\mathbf{x})^T(\mathbf{X}^T\mathbf{X})^{-1}\boldsymbol{\xi}(\mathbf{x})$, over a specified set of design points. There are other optimal designs that will not be covered in this text. Interested readers are referred to Atkinson et al. [25]

D-optimal design

In Chapter 2, we discussed different types of uncertainty in linear regression. From the assumption that the model form of PRS is correct but samples have random noise, different realizations of noise can cause different fitting results, which is the source of uncertainty. This uncertainty first causes the uncertainty of the vector of regression coefficients $\mathbf{b}$ in the form of covariance matrix as shown in Eq. (2.38): $\mathbf{\Sigma_b} = \hat{\sigma}^2[\mathbf{X}^T\mathbf{X}]^{-1}$. Then, the uncertainty in the coefficients can cause the uncertainty in prediction, which is given in Eq. (3.10): $\sigma_y(\mathbf{x}) = \hat{\sigma}\sqrt{\boldsymbol{\xi}(\mathbf{x})^T(\mathbf{X}^T\mathbf{X})^{-1}\boldsymbol{\xi}(\mathbf{x})}$. Most optimal design methods try to find sample locations to reduce these two types of uncertainty. Since the standard deviation of noise $\hat{\sigma}$ depends on the samples, the users cannot control it. It is aleatory uncertainty, and we have to live with it. The vector of basis functions $\boldsymbol{\xi}(\mathbf{x})$ is fixed when the model form of PRS is given. Therefore, the only term that can vary is the inverse of the moment matrix $(\mathbf{X}^T\mathbf{X})^{-1}$, which depends on the number and locations of samples. Therefore, most optimal DoE methods try to reduce $(\mathbf{X}^T\mathbf{X})^{-1}$ so that the standard error of prediction can be reduced.

The idea behind the D-optimal design is that since the model form is accurate, if there is no uncertainty in the regression coefficients, the prediction will also be accurate. Therefore, a good design makes the covariance matrix $\mathbf{\Sigma_b}$ small, which is equivalent to make $(\mathbf{X}^T\mathbf{X})^{-1}$ small or to make the moment matrix $\mathbf{X}^T\mathbf{X}$ large. Since $\mathbf{X}^T\mathbf{X}$ is a matrix, the magnitude of the moment matrix is represented by the determinant of the matrix. Therefore, D-optimal DoE is to find $(\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_{n_y})$, such that

$$\text{maximize } D = |\mathbf{X}^T\mathbf{X}| \tag{3.13}$$

The determinant of the moment matrix can be shown to be directly related to the confidence region of the coefficients of the PRS (a confidence region for a coefficient is the region where the coefficient will lie with a given probability). In fact, it is inversely proportional to the square of the volume of the confidence region of the coefficients. Therefore, maximizing the determinant increases our confidence in the coefficients. Unlike traditional DoEs, D-optimal designs do not require orthogonal design matrices, and as a result, parameter estimates may be correlated. Parameter estimates may also be locally, but not globally, D-optimal.

The details of the D-optimal algorithm can be found in Atkinson et al. [25]. First, since the optimal designs depend on the surrogate model, it is necessary to specify an approximate mathematical model which defines the functional form of the relationship between the QoI and the independent variables (the design variables). Next, generate a set of possible candidate points based on this model. Finally, from these candidates select the subset that maximizes the determinant of the moment matrix $|\mathbf{X}^T\mathbf{X}|$. The number of possible designs grows rapidly as the complexity of the model increases. This number is usually so large that an exhaustive search of all possible designs for a given sample size is not feasible. The D-optimal algorithm begins with a randomly selected set of points. Points in and out of the current design are exchanged until no exchange can be found that increases the determinant of the moment matrix. To cut down on the running time, the number of points considered during any iteration may be limited.

Unfortunately, this method does not guarantee that the global maximum can be found. To overcome this, the algorithm is repeated several times in hopes that at least one iteration leads to the global maximum. For example, the algorithm can repeat 50 or 100 times by starting with a random initial n_y set of samples. However, since the criterion is not convex and since the dimension is high, there is no guarantee to obtain the global maximum of the determinant.

Finding the D-optimal set of points from a given set of points is often a difficult combinatorial problem. For example, if we need to find 10 D-optimal points out of 50 sample locations, we can have ${}_{10}C_{50} \approx 10^{10}$ possible combinations. Therefore, the number of combinations becomes huge even for moderate-size problems. Solution algorithms can rarely find the D-optimal set and usually settle on a

suboptimal but good set. Some solution algorithms are based on replacing one point at a time and taking advantage of inexpensive expressions for updating the determinant when one point is changed. Genetic algorithms have also been used to find a good design based on D-optimality.

Example 3-7
In order to fit a linear PRS $\hat{y} = b_1 x_1 + b_2 x_2$, two sample locations were given as $(0,0)$ and $(1,0)$. Find the third sample location using a D-optimal design in the design space $x_1, x_2 \in [0,1]$.

Solution:
Let the third sample location be (p, q). Then the design matrix and the moment matrix are defined as

$$\mathbf{X} = \begin{bmatrix} 0 & 0 \\ 1 & 0 \\ p & q \end{bmatrix}, \qquad \mathbf{X}^{\mathrm{T}}\mathbf{X} = \begin{bmatrix} 1+p^2 & pq \\ pq & q^2 \end{bmatrix}, \qquad \left|\mathbf{X}^{\mathrm{T}}\mathbf{X}\right| = q^2$$

For a given design space, the maximum of $\left|\mathbf{X}^{\mathrm{T}}\mathbf{X}\right|$ occurs at $q = 1$, while p is arbitrary. Therefore, based on the D-optimal design, the third sample point is selected at $(p, 1)$ with p being arbitrary.

It would be interesting to check how the D-optimal design actually reduces the uncertainty in the regression coefficients. Based on Eq. (2.38), the covariance matrix of regression coefficients can be calculated as

$$\boldsymbol{\Sigma}_{\mathbf{b}} = \hat{\sigma}[\mathbf{X}^{\mathrm{T}}\mathbf{X}]^{-1} = \hat{\sigma}\begin{bmatrix} 1 & -\frac{p}{q} \\ -\frac{p}{q} & \frac{1}{q^2} + \frac{p^2}{q^2} \end{bmatrix}$$

Therefore, it is obvious that $q = 1$ gives lower variances (diagonal terms) and lower correlations (off-diagonal terms). Although the D-optimal design concludes that p can be arbitrary, the covariance matrix shows that when $p = 0$, the uncertainty in the coefficients is minimum. For example, when $p = 0$, the correlation between the two coefficients is zero, and the variance of b_1 is minimum. Therefore, the D-optimal design may not capture the true minimum of the covariance matrix. This is because a single scalar measure may not fully represent the complex behavior of the entire matrix.

Matlab provides various functions for D-optimal designs. Since D-optimal designs depend on the PRS model, either the users specify the PRS model or provide the design matrix. Also, as mentioned before, it is possible that the users generate candidate samples first and the Matlab function can choose the best sample locations out of all candidate locations. Otherwise, it is also possible that n_y sample locations are randomly generated first, and then, they are moved to different locations to improve the D-optimality criterion. In order to specify the PRS model, the following options are available: '`linear`', '`interaction`', '`quadratic`', '`purequadratic`'. '`linear`' and '`quadratic`' are the standard linear and quadratic PRS surrogates, respectively. '`interaction`' only includes interaction terms, such as x_1x_2, x_1x_3, etc. without squared terms, while '`purequadratic`' only includes squared terms, such as x_1^2, x_2^2, etc., without interaction terms.

Matlab function `rlist = candexch(C,nrows)` uses a row-exchange algorithm to select `nrows` number of D-optimal designs from the candidate set. The matrix `C` is the candidate design matrix, and the output `rlist` is the list of selected rows. Possible options are how to choose initial `nrows` samples, the maximum number of iterations, and the number of times to try to generate a design from new starting locations. In order to find a better local optimum, the algorithm can repeat multiple times with

random starting locations. This function has the advantage of using non-standard basis functions (e.g., without a constant term), while the disadvantage is that the users need to generate the design matrix at all candidate points.

Matlab function `[dRE,X] = rowexch(n,ny,model)` uses a row-exchange algorithm to generate a D-optimal design with `ny` samples for a linear additive model with `n` design variables. The output `dRE` is a $n_y \times n$ matrix of sample locations, and `X` is the design matrix. The `rowexch` function first generates candidate samples automatically and then utilizes `candexch` function to find the D-optimal design. The only difference is that `rowexch` can only use specific polynomial forms that are specified in the `model`. Both `candexch` and `rowexch` can only find D-optimal design out of candidate sample locations.

Matlab function `[dCE,X] = cordexch(n,ny,'model')` is different from `rowexch` in the sense that it does not choose a D-optimal design from candidate locations. Instead, it starts with `ny` initial sample locations that are generated randomly and it moves the previous locations to new locations to increase the determinant of the moment matrix. At each step, the coordinate-exchange algorithm exchanges a single element of design matrix **X** with a new element evaluated at a neighboring point in the design space. Since the algorithm searches a small neighborhood of the current sample location, the algorithm is more likely to become trapped in a local minimum.

Example 3-8
Use Matlab `cordexch` function to generate 6 and 12 samples for two-dimensional quadratic PRS using D-optimal design. Discuss the sample locations and the ratio of uncertainty between the two sample sets.

Solution:
Since two-dimensional quadratic PRS has six regression coefficients, a minimum of six samples are required. The following Matlab code can calculate D-optimal designs for six and twelve samples.

```
ny=6;                       %With 6 samples:
[dce,X]=cordexch(2,ny,'quadratic');
scatter(dce(:,1),dce(:,2),200,'filled')
D6=det(X'*X)

ny=12;                     %With 12 samples:
[dce,X]=cordexch(2,ny,'quadratic');
scatter(dce(:,1),dce(:,2),200,'filled')
D12=det(X'*X)
```

With six samples, the determinant of the moment matrix was $|\mathbf{X}^T\mathbf{X}| = 256$, while that of twelve samples was $|\mathbf{X}^T\mathbf{X}| = 30{,}320$. Therefore, it is expected that the uncertainty in the regression coefficient is much lower for the twelve samples. In terms of the ratio between the two determinants, the twelve samples may have less than 1% of uncertainty than the six samples. The sample locations are shown in Figure 3-11, and the individual locations are shown in the following tables.

Six sample locations:

Sample No.	1	2	3	4	5	6
x_1	1	-1	-1	1	0	0
x_2	-1	1	-1	1	0	-1

Twelve sample locations:

Sample No.	1	2	3	4	5	6	7	8	9	10	11	12
x_1	1	-1	0	-1	1	-1	1	1	1	-1	0	0
x_2	-1	-1	-1	1	1	0	1	0	-1	1	1	0

It is noticed that all samples are on the edge of the design space, except for the center locations. In the case of six samples, four samples were located in the corners, and the fifth sample is at the center. The last sample was located at $(-1, 0)$, but it could have been positioned in either $(1,0)$, $(0, -1)$, or $(0, 1)$. In the case of twelve samples, the nine samples were located in two-level full factorial designs. The remaining three samples actually overlapped with some of these nine samples (samples 7, 9, and 10).

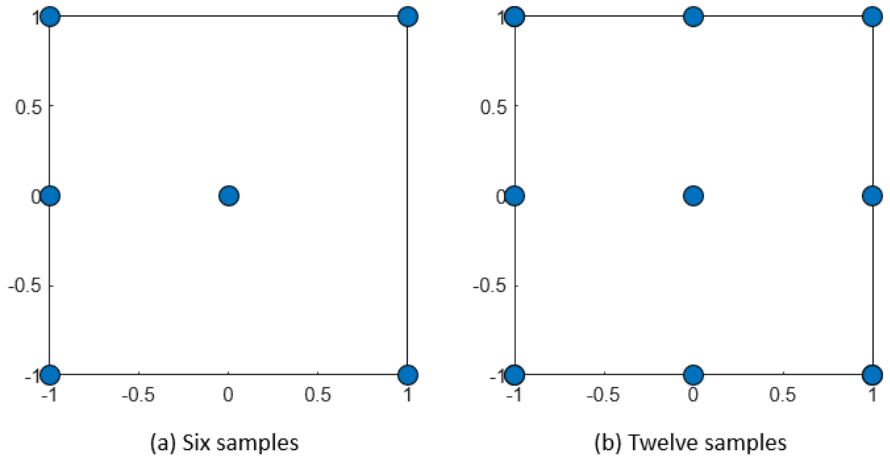


Figure 3-11: D-optimal DoEs for six and twelve samples.

A-optimal design

A-optimal design is similar to D-optimal design in the sense that both focus on the variance of regression coefficients. In order to reduce the covariance matrix of coefficients, the D-optimal design maximizes the determinant of the moment matrix. The A-optimal design tries to reduce the variance of individual coefficients. The variance of each coefficient relies on the diagonal terms of the inverse of the moment matrix $(\mathbf{X}^T\mathbf{X})^{-1}$. The A-optimal design seeks to minimize the sum of these elements, that is the trace of $(\mathbf{X}^T\mathbf{X})^{-1}$. Therefore, the A-optimal design is to find $(\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_{n_y})$, such that

$$\text{minimize} \sum_{i=1}^{n_\beta} (\mathbf{X}^T\mathbf{X})_{ii}^{-1} \tag{3.14}$$

Jones et al. [26] showed that the A-optimal design is more consistent with the screening objective than the D-optimal design. An A-optimal design minimizes the average variance of the parameter estimates, which is directly related to that goal. While there are many cases where A- and D-optimal designs coincide, the A-optimal designs tend to have better statistical properties when the A- and D-optimal designs differ. In such cases, A-optimal designs generally have more uncorrelated columns in their model matrices than D-optimal designs. Also, even though A-optimal designs minimize the average variance of the parameter estimates, various cases exist where they outperform D-optimal designs in terms of the variances of all individual parameter estimates. A-optimal designs can also substantially reduce the worst prediction variance compared with D-optimal designs.

Example 3-9
Repeat **Example 3-7** with an A-optimal design.

Solution:
From the covariance matrix given in **Example 3-7**, the sum of diagonal terms can be written as

$$\sum_{i=1}^{2}(\mathbf{X}^T\mathbf{X})_{ii}^{-1} = \left(1 + \frac{1+p^2}{q^2}\right)$$

The above term has a minimum when $p^2 = 0$ and $q^2 = 1$. Therefore, the third sample location using the A-optimal design becomes $(p, q) = (0,1)$. Note that in the D-optimal design p was arbitrary, while the A-optimal design found the best location for minimizing the variance of the two coefficients.

G-optimal design
D-optimal and A-optimal designs try to reduce uncertainty in the covariance matrix of coefficients, which can indirectly reduce the prediction variance. However, it makes more sense if an optimal design tries to reduce prediction variance directly, which is called the G-optimal design. Since prediction variance is a function of the prediction point, the G-optimal design minimizes the maximum prediction variance as

$$\text{minimize } \max v(\mathbf{x}) = \boldsymbol{\xi}(\mathbf{x})^T(\mathbf{X}^T\mathbf{X})^{-1}\boldsymbol{\xi}(\mathbf{x}) \tag{3.15}$$

It can be shown that under the standard statistical assumptions about the error that the maximum prediction variance in the domain defined by the sample points is always larger or equal to the number of terms in the response surface, n_β, (see Myers and Montgomery, 1995, p. 367). Therefore, with a G-optimal design, we have a target to shoot for. We would like to seek a set of points that will bring the maximum close to n_β/n_y. This value is achieved by a two-level full-factorial design for a linear model, see **Example 3-2**, where the prediction variance was $3\hat{\sigma}/4$ (four samples with three coefficients).

If all sample locations are sought, the total number of design variables would be $n_y \times n$, which becomes impractical for a large number of samples or high-dimensional design space. In adaptive sampling, some sample locations are fixed, and additional k samples can be sought. In such a case, the number of design variables would be $k \times n$.

In general, finding n_y sample locations such that the maximum prediction variance is minimized as in Eq. (3.15) is a two-level optimization problem. In the inner level, for a given set of sample locations, we need to find the location where the prediction variance is maximum and the maximum prediction variance value. This optimization problem is formulated as follows: find $\mathbf{x} = \{x_1, \cdots, x_n\}^T$ to satisfy

$$\begin{aligned} &\min f(\mathbf{x}) = -\boldsymbol{\xi}(\mathbf{x})^T(\mathbf{X}^T\mathbf{X})^{-1}\boldsymbol{\xi}(\mathbf{x}) \\ &\text{such that } \; -1 \le \mathbf{x} \le 1 \end{aligned} \tag{3.16}$$

In this equation, it is assumed that the design matrix is known (or the sample locations are known). The maximum prediction variance is then given as $-f$. The outer-level optimization is to find the sample locations; i.e., $\{\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \cdots, \mathbf{x}^{(n_y)}\}$ such that the maximum prediction variance is minimized. The optimization problem is formulated as

$$\min g\left(\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \cdots, \mathbf{x}^{(n_y)}\right) = \max\left(-\boldsymbol{\xi}^T(\mathbf{X}^T\mathbf{X})^{-1}\boldsymbol{\xi}\right) \tag{3.17}$$

$$|\mathbf{X}^T\mathbf{X}| > 0$$

Last condition on the determinant of the function $|\mathbf{X}^T\mathbf{X}|$ is imposed to avoid the repetition of the points, which makes the matrix singular.

The optimization problem is difficult to solve because the lower-level optimum is not a smooth function of the upper-level design variables (the position of the sample points). This is because as the data points change, the position of the maximum variance can jump. Therefore, most gradient-based optimization ends up with a local optimum. Gradient-free global search algorithms, such as a genetic algorithm, can be computationally expensive. In practice, the inner-level optimization problem, finding the maximum prediction variance, is often replaced by grid evaluation, where the design space is discretized by a grid and the maximum prediction variance is sought among the grid.

Example 3-10

Repeat **Example 3-7** with a G-optimal design. Compare the maximum standard error of prediction for D-optimal, A-optimal, and G-optimal designs.

Solution:

Since Eq. (3.15) is a min-max problem, it would be difficult to solve it analytically. We will solve the problem heuristically with some reasonings. Using the inverse of the moment matrix in **Example 3-7**, the prediction variance term in Eq. (3.15) can be written as

$$v(\mathbf{x}) = \boldsymbol{\xi}(\mathbf{x})^T(\mathbf{X}^T\mathbf{X})^{-1}\boldsymbol{\xi}(\mathbf{x}) = \left(x_1 - 2\frac{p}{q}x_2\right)^2 + \frac{1+p^2}{q^2}x_2^2$$

First, since q only occurs in the denominator, if $q = 0$, the variance term goes to infinity. Therefore, it is obvious that q must be in its upper bound to reduce the variance. Therefore, we heuristically determine $q = 1$. Then, the variance can be simplified as

$$v(\mathbf{x}) = (x_1 - 2px_2)^2 + (1+p^2)x_2^2$$

Since the maximum variance normally occurs at corner points, we evaluate the variance at four corners:

$$v(0,0) = 0$$
$$v(1,0) = 1$$
$$v(0,1) = 1 + p^2$$
$$v(1,1) = 2 - 2p + p^2$$

It is obvious that the maximum may occur either at $(0,1)$ or $(1,1)$, depending on the value of p. When $p \geq 0.5$, the maximum occurs at $(0,1)$, and the minimum of the maximum occurs at $p = 0.5$. When $p \leq 0.5$, the maximum occurs at $(1,1)$, and the minimum of the maximum occurs at $p = 0.5$. Therefore, the G-optimal design yields $(p, q) = (0.5,1)$, which is the middle of the upper bound.

Note that the sample location of the D-optimal design in **Example 3-7** is not fixed. For the purpose of comparison, however, we choose the third sample location of the D-optimal design at (1,1), just to make it different from the other two designs. With the third sample at $(0.5,1)$, the maximum standard error of prediction of G-optimal design becomes $\sigma_y^{max} = \sqrt{5}\hat{\sigma}/2$. It is interesting to note that the maximum standard error of prediction from D-optimal and A-optimal designs are $\sigma_y^{max} = \sqrt{2}\hat{\sigma}$. Therefore, G-optimal design yields the sample location that has the smallest maximum standard error of prediction.

Figure 3-12 compares the location of the third sample and the standard error of prediction for D-optimal, A-optimal, and G-optimal designs. When the third sample locates in one corner, the maximum standard error occurs in the other corner as this is the farthest extrapolation point. On the other hand, since the G-optimal design locates the third sample in the middle of the edge, the distance to the farthest

extrapolation point is shorter than the other two designs. It is noted that the minimum standard error of all three designs is zero at the origin. This happens because the linear PRS $\hat{y} = b_1 x_1 + b_2 x_2$ does not have a constant term. The following Matlab code is used to plot the contour of the standard error of prediction:

```
X=[0 0;1 0;1 1];     % D-optimal design
%X=[0 0;1 0;0 1];     % A-optimal design
%X=[0 0;1 0;0.5 1];   % G-optimal design
%
XTX=X'*X;
XTXi=inv(XTX);
[X, Y]=meshgrid(0:.1:1, 0:.1:1);
Z=zeros(size(X));
[n, m]=size(X);
for i=1:n
    for j=1:m
        xi=[X(i,j) Y(i,j)];
        Z(i,j)=sqrt(xi*XTXi*xi');
    end
end
v=linspace(min(min(Z)),max(max(Z)),10);
[C,h]=contour(X,Y,Z,v);
clabel(C,h)
Zmax=max(max(Z))
Zmin=min(min(Z))
```

It would be interesting to compare the covariance matrix of the three designs. Remember that D-optimal and A-optimal designs are for minimizing the covariance matrix. The covariance matrices of the three designs are summarized in the following table:

Designs	D-optimal	A-optimal	G-optimal
Covariance matrix	$\mathbf{\Sigma_b} = \hat{\sigma}\begin{bmatrix} 1 & -1 \\ -1 & 2 \end{bmatrix}$	$\mathbf{\Sigma_b} = \hat{\sigma}\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$	$\mathbf{\Sigma_b} = \hat{\sigma}\begin{bmatrix} 1 & -0.5 \\ -0.5 & 1.25 \end{bmatrix}$

It turns out that the variance of regression coefficients is the smallest for the A-optimal design. In addition, the A-optimal design shows uncorrelated coefficients. Even if the D-optimal design is for minimizing the determinant of the moment matrix, it shows the largest variance among the three and also a significant correlation. G-optimal design is good for maximum prediction variance, but it has a relatively large variance and correlation between coefficients.

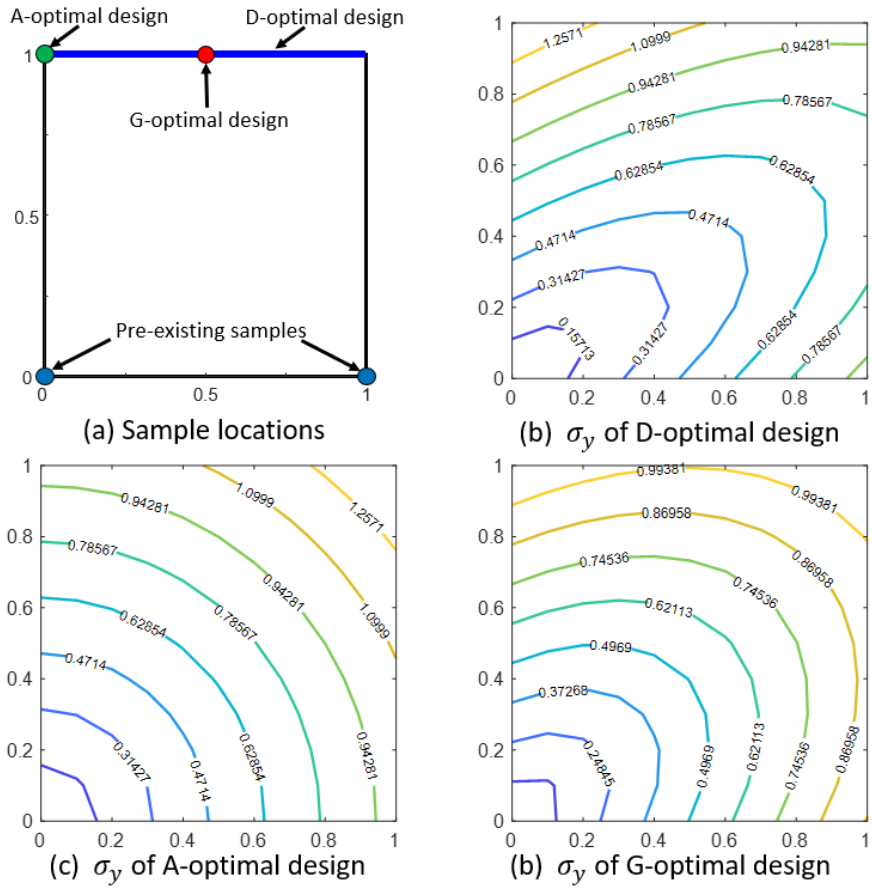


Figure 3-12: Comparison of sample locations and standard error of prediction for D-optimal, A-optimal, and G-optimal designs.

Minimum bias design

The three optimal designs that we discussed in the previous subsections only focus on reducing uncertainty in regression coefficients or prediction. Unfortunately, reducing uncertainty does not always guarantee prediction accuracy. It has an assumption that first, the model form is accurate, and second, the mean of regression coefficients is identical to the true coefficients. When the model form is not accurate or the regression process cannot estimate the coefficients accurately, there exists an error in surrogate prediction. This error is called modeling error by engineers and bias error by statisticians.

Bias measures the difference between surrogate prediction $\hat{y}(\mathbf{x})$ of QoI and the true QoI $y(\mathbf{x})$. Since the true QoI is often unknown and measurements have random noise, the average of measurements is commonly considered the true QoI. On the other hand, variance measures how surrogate $\hat{y}(\mathbf{x})$ is affected by a particular dataset. Samples are assumed to be generated from the true QoI by adding random noise. If the surrogate fit is perfect, it filters out the random noises in the samples, and different datasets should yield the same surrogate prediction. However, this can be achieved when the number of samples approaches infinity. With a finite number of samples, different datasets will yield different surrogate predictions. This uncertainty in prediction is measured in terms of variance.

Unfortunately, variance and bias are competing objectives in DoE. The bias error can be reduced by fitting the surrogate close to samples (e.g., by adding more basis functions), but this cause a large variance. On the other hand, the variance error can be reduced by smoothing the surrogate model (e.g., by adding a penalty term as in Eq. (1.6)), but this smoothing penalty inevitably increases the bias error.

Therefore, it is necessary to balance the bias and variance errors by a trade-off. In theory, it is possible to improve the bias and variance error simultaneously by adding more samples and more basis functions.

For PRS surrogate models, using the assumption that the true model is also in a polynomial form, Myers and Montgomery [4] proposed minimum bias designs. To consider the model errors, it would be necessary to introduce the concepts of design moments. Let R be the region of interest in terms of predicting QoI, and let μ_i be the first moments of the domain, which is defined as

$$\mu_i = \frac{1}{V}\int_R x_i \, \mathrm{d}R\,, \qquad i = 1, \cdots, n \tag{3.18}$$

where V is the volume of the domain R. Similarly, the second moment μ_{ij} can be defined as

$$\mu_{ij} = \frac{1}{V}\int_R x_i x_\mathrm{j} \, \mathrm{d}R\,, \qquad i, j = 1, \cdots, n \tag{3.19}$$

Higher-order moments can also be defined in a similar way.

Since samples are given at discrete points, the above definition of moments needs to be extended to the case with discrete samples. First, let there are n_y samples in n-dimensional design space. Then $x_{ik}, (i = 1, \cdots, n, k = 1, \cdots, n_y)$ represents the i-th component of the k-th sample. Therefore, the discrete counterpart of the moment is defined as

$$m_i = \frac{1}{n_y}\sum_{k=1}^{n_y} x_{ik}\,, \qquad i = 1, \cdots, n \tag{3.20}$$

Higher-order moments can also be fined in a similar way.

The concept of minimum bias design is based on the equivalence of the discrete moments with the continuous moments. Let us consider a PRS surrogate model given in the form of $\hat{y}(\mathbf{x}) = \boldsymbol{\xi}(\mathbf{x})^{(1)^T}\mathbf{b}^{(1)}$, and it has a bias term. Assuming that the bias term is given in a polynomial form with higher orders, the true function can be defined as $y(\mathbf{x}) = \boldsymbol{\xi}(\mathbf{x})^{(1)^T}\mathbf{b}^{(1)} + \boldsymbol{\xi}(\mathbf{x})^{(2)^T}\mathbf{b}^{(2)}$, where the second term corresponds to the bias. With given n_y samples, the relationship between the samples and regression coefficients can be written as

$$\mathbf{y} = \mathbf{X}^{(1)}\mathbf{b}^{(1)} + \mathbf{X}^{(2)}\mathbf{b}^{(2)} \tag{3.21}$$

where $\mathbf{X}^{(1)}$ and $\mathbf{X}^{(2)}$ are the design matrices using $\boldsymbol{\xi}(\mathbf{x})^{(1)}$ and $\boldsymbol{\xi}(\mathbf{x})^{(2)}$, respectively. The averaged moment matrix of the combined design matrix can be partitioned as

$$\mathbf{M} = \begin{bmatrix} \mathbf{M}_{11} & | & \mathbf{M}_{12} \\ -- & | & -- \\ \mathbf{M}_{21} & | & \mathbf{M}_{22} \end{bmatrix} = \frac{1}{n_y}\begin{bmatrix} \mathbf{X}^{(1)^T}\mathbf{X}^{(1)} & | & \mathbf{X}^{(1)^T}\mathbf{X}^{(2)} \\ ------ & | & ------ \\ \mathbf{X}^{(2)^T}\mathbf{X}^{(1)} & | & \mathbf{X}^{(2)^T}\mathbf{X}^{(2)} \end{bmatrix} \tag{3.22}$$

On the other hand, the moment in the continuous domain can be defined using the basis functions as

$$\boldsymbol{\mu} = \begin{bmatrix} \boldsymbol{\mu}_{11} & | & \boldsymbol{\mu}_{12} \\ -- & | & -- \\ \boldsymbol{\mu}_{21} & | & \boldsymbol{\mu}_{22} \end{bmatrix} = \frac{1}{V}\begin{bmatrix} \int_R \boldsymbol{\xi}(\mathbf{x})^{(1)}\boldsymbol{\xi}(\mathbf{x})^{(1)^T}\mathrm{d}R & | & \int_R \boldsymbol{\xi}(\mathbf{x})^{(1)}\boldsymbol{\xi}(\mathbf{x})^{(2)^T}\mathrm{d}R \\ -------- & | & -------- \\ \int_R \boldsymbol{\xi}(\mathbf{x})^{(2)}\boldsymbol{\xi}(\mathbf{x})^{(1)^T}\mathrm{d}R & | & \int_R \boldsymbol{\xi}(\mathbf{x})^{(2)}\boldsymbol{\xi}(\mathbf{x})^{(2)^T}\mathrm{d}R \end{bmatrix} \tag{3.23}$$

Comparing the discrete moment in Eq. (3.22) with the continuous moment in Eq. (3.23), the regression process is nothing but approximating the integrals of basis functions by a sum over the sample points. Therefore, good samples make this approximation accurate. The idea of minimum bias design is to

choose sample locations such that the mean-squared-bias error of PRS is minimized, which can be achieved by making the discrete moment equal to the continuous moment. If there is enough freedom in selecting sample locations, it is possible to make the difference disappear. If that is not possible, optimization methods can be used to seek the best points to minimize the difference.

Box and Draper [29] proved that minimum bias designs satisfy the following relationship:

$$\mathbf{M}_{11}^{-1}\mathbf{M}_{12} = \boldsymbol{\mu}_{11}^{-1}\boldsymbol{\mu}_{12} \tag{3.24}$$

Box and Draper concluded that the following conditions

$$\mathbf{M}_{11} = \boldsymbol{\mu}_{11}, \qquad \mathbf{M}_{12} = \boldsymbol{\mu}_{12} \tag{3.25}$$

are sufficient conditions for a minimum bias design (Myers and Montgomery [4], p. 411). When the true functional form is known, the vector of basis functions, $\boldsymbol{\xi}(\mathbf{x})^{(2)}$, is available. Without having the true functional form, however, the second condition in Eq. (3.25) cannot be imposed. Therefore, in most cases, the first condition is used to define sample locations with additional constraints. For a given vector of basis function, the continuous moment matrix, $\boldsymbol{\mu}_{11}$, is calculated first. Then, the unknown sample locations are found by matching the discrete moment matrix, $\mathbf{M}_{11}$, with $\boldsymbol{\mu}_{11}$.

In contrast to the minimum variance designs that tend to put points in the periphery of the design domain, minimum bias designs tend to bring them closer to the centroid. In addition, minimum bias designs often have low prediction variance, but the reverse is not true. That is, minimum variance designs tend to have large bias errors. Compromise designs, in which $\mathbf{M}_{11}$ and $\mathbf{M}_{12}$ are slightly larger than $\boldsymbol{\mu}_{11}$, and $\boldsymbol{\mu}_{12}$, respectively, are occasionally used.

Example 3-11

The true function is a quadratic polynomial, given in the form $y(\mathbf{x}) = x_1^2 + x_2^2$, which is only used to generate samples. A linear PRS, $\hat{y}(\mathbf{x}) = b_1 + b_2x_1 + b_3x_2$, will be constructed in the design space $x_1, x_2 \in [-1,1]$. Find four symmetric sample locations using a minimum bias design. Compare e_{RMS} of minimum bias design with that of a two-level full factorial design.

Solution:

Since the true functional form is unknown, we will only use the condition, $\mathbf{M}_{11} = \boldsymbol{\mu}_{11}$. For the linear model with the vector of basis functions, $\boldsymbol{\xi}(\mathbf{x})^{(1)} = \{1, x_1, x_2\}^T$, the continuous moment matrix is

$$\boldsymbol{\mu}_{11} = \frac{1}{V}\int_{-1}^{1}\int_{-1}^{1}\begin{bmatrix} 1 & x_1 & x_2 \\ x_1 & x_1^2 & x_1x_2 \\ x_2 & x_1x_2 & x_2^2 \end{bmatrix} \mathrm{d}x_1\mathrm{d}x_2 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \frac{1}{3} & 0 \\ 0 & 0 & \frac{1}{3} \end{bmatrix}$$

Note that because of the symmetry of the domain, all the integrals with odd powers of either x_1 or x_2 are zero.

The discrete moment matrix $\mathbf{M}_{11}$ can be calculated from its definition in Eq. (3.22)

$$\mathbf{M}_{11} = \frac{1}{n_y}\mathbf{X}^{(1)^T}\mathbf{X}^{(1)} = \frac{1}{n_y}\begin{bmatrix} \sum_{k=1}^{n_y} 1 & \sum_{k=1}^{n_y} x_{1k} & \sum_{k=1}^{n_y} x_{2k} \\ \sum_{k=1}^{n_y} x_{1k} & \sum_{k=1}^{n_y} x_{1k}^2 & \sum_{k=1}^{n_y} x_{1k}x_{2k} \\ \sum_{k=1}^{n_y} x_{2k} & \sum_{k=1}^{n_y} x_{1k}x_{2k} & \sum_{k=1}^{n_y} x_{2k}^2 \end{bmatrix}$$

If we pick four points that are symmetric with respect to both the x_1- and x_2-axis, there are two possibilities. One is the set of samples along the axis of each variable: $(\pm\alpha, 0), (0, \pm\alpha)$, where $0 \le \alpha \le 1$ is a constant. The second is the set of samples along $\pm 45°$ lines: $(\pm\beta, \pm\beta)$, where $0 \le \beta \le 1$ is a constant. Because of the symmetry, the sums involving odd powers will vanish, so that all the zeros in $\mathbf{M}_{11}$ will match the zeroes in $\boldsymbol{\mu}_{11}$. Only the diagonal terms are non-zero. The value of α and β is calculated by setting the diagonal terms in $\mathbf{M}_{11}$ matrix equal to the corresponding component in $\boldsymbol{\mu}_{11}$. That is,

$$\frac{1}{4}\sum_{k=1}^{4} 1 = 1, \qquad \frac{1}{4}\sum_{k=1}^{4} x_{1k}^2 = \frac{1}{3}, \qquad \frac{1}{4}\sum_{k=1}^{4} x_{2k}^2 = \frac{1}{3}$$

The first term is trivial. The remaining two terms are used to calculate α or β.

For the first set $(\pm\alpha, 0), (0, \pm\alpha)$, these equations yield

$$\frac{1}{4}(\alpha^2 + \alpha^2) = \frac{1}{3}, \qquad \rightarrow \alpha = \sqrt{\frac{2}{3}} = 0.8165$$

Therefore, the four sample locations are along each axis: $(\pm 0.8165, 0), (0, \pm 0.8165)$. With these sample locations, the regression coefficients can be calculated as

$$\mathbf{X} = \begin{bmatrix} 1 & -\alpha & 0 \\ 1 & \alpha & 0 \\ 1 & 0 & -\alpha \\ 1 & 0 & \alpha \end{bmatrix}, \qquad \mathbf{y} = \begin{Bmatrix} \alpha^2 \\ \alpha^2 \\ \alpha^2 \\ \alpha^2 \end{Bmatrix}, \qquad \mathbf{b} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y} = \begin{Bmatrix} \alpha^2 \\ 0 \\ 0 \end{Bmatrix}$$

Therefore, the linear PRS becomes $\hat{y}(\mathbf{x}) = b_1 + b_2 x_1 + b_3 x_2 = \alpha^2 = 2/3$, which is a constant.

For the second set $(\pm\beta, \pm\beta)$, these equations yield

$$\frac{1}{4}(\beta^2 + \beta^2 + \beta^2 + \beta^2) = \frac{1}{3}, \qquad \rightarrow \beta = \sqrt{\frac{1}{3}} = 0.5774$$

Therefore, the four sample locations are along each axis: $(\pm 0.5774, \pm 0.5774)$. With these sample locations, the regression coefficients can be calculated as

$$\mathbf{X} = \begin{bmatrix} 1 & -\beta & -\beta \\ 1 & -\beta & \beta \\ 1 & \beta & -\beta \\ 1 & \beta & \beta \end{bmatrix}, \qquad \mathbf{y} = \begin{Bmatrix} 2\beta^2 \\ 2\beta^2 \\ 2\beta^2 \\ 2\beta^2 \end{Bmatrix}, \qquad \mathbf{b} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y} = \begin{Bmatrix} 2\beta^2 \\ 0 \\ 0 \end{Bmatrix}$$

Therefore, the linear PRS becomes $\hat{y}(\mathbf{x}) = b_1 + b_2 x_1 + b_3 x_2 = 2\beta^2 = 2/3$, which is a constant. Note that the two surrogates are identical.

Now let us compare these two sets to the full factorial design $(\pm 1, \pm 1)$ for fitting the function $y(\mathbf{x}) = x_1^2 + x_2^2$. With these sample locations, the regression coefficients can be calculated as

$$\mathbf{X} = \begin{bmatrix} 1 & -1 & -1 \\ 1 & -1 & 1 \\ 1 & 1 & -1 \\ 1 & 1 & 1 \end{bmatrix}, \qquad \mathbf{y} = \begin{Bmatrix} 2 \\ 2 \\ 2 \\ 2 \end{Bmatrix}, \qquad \mathbf{b} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y} = \begin{Bmatrix} 2 \\ 0 \\ 0 \end{Bmatrix}$$

Therefore, for the full factorial design, the surrogate model is also a constant function $\hat{y}(\mathbf{x}) = 2$. Obviously, the surrogate $\hat{y}(\mathbf{x}) = 2/3$ of the minimum bias design is better than that of the full factorial design. Figure 3-13 shows the sample locations of the minimum bias design. The circular markers are for the first design, while the square markers are for the second design.

To appreciate that the fit is optimal, consider fitting the function by a general linear polynomial. Because of the double symmetry of the function, all the linear terms should vanish, so that the response surface should indeed be of the form $\hat{y}(\mathbf{x}) = b_1$. The mean square error over the region is then

$$e_{RMS}^2 = \frac{1}{4}\int_{-1}^{1}\int_{-1}^{1}(x_1^2 + x_2^2 - b_1)^2 \mathrm{d}x_1 \mathrm{d}x_2 = \frac{28}{45} - \frac{4}{3}b_1 + b_1^2$$

Differentiating the error with respect to b_1 and setting it to zero confirms the fact that $b_1 = 2/3$ gives the minimum error, $e_{RMS}^2 = 8/45$. In contrast, $b_1 = 2$ gives $e_{RMS}^2 = 88/45$.

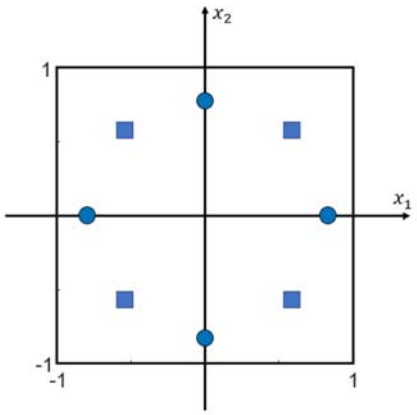


Figure 3-13: Sample locations of minimum bias designs.

If this example appears impressive, note that we did not have any noise at all in the function, and the example was selected to make the minimum bias design look good. In other cases, the results may be less dramatic, and a compromise between minimum bias and minimum variance may be called for.

3.4. Space-filling design of experiments

The optimal designs in the previous section are good for finding sample locations that minimize prediction uncertainty and/or bias error of surrogate prediction. The optimal designs are especially useful for adaptive sampling, where the locations of additional samples are sought in addition to pre-existing samples. The limitations of optimal designs are: (a) it can be computationally challenging if the dimension of design space is high and the number of samples is large, and (b) the optimal designs depend on the specific surrogate model and the number of samples. Especially, most optimal designs in Section 3.3 are based on PRS surrogate, which uses linear regression with polynomial basis functions. Therefore, optimal designs are difficult to generalize for a general surrogate model with an arbitrary number of samples.

DoE methods that are independent of surrogate models are based on the simple fact that surrogate predictions are accurate when the prediction point is close to the sample location. Therefore, it is always a good idea to distribute samples regularly in the design space. But as shown in Section 3.2, it would require too many samples to construct dense full-factorial designs with many levels. Instead, fractional factorial designs are sought in order to reduce the required number of samples, and at the same time, satisfy the uniformity as much as possible. This can be achieved by either maximizing the minimum distance [27] or minimizing correlation measures among samples [28]. Clusters of samples can improve prediction accuracy but can be considered a waste of resources. A large empty space in the interpolation

region or a large extrapolation region can cause poor surrogate predictions. Therefore, a good DoE method means filling the design space as much as possible for a given number of samples. This strategy of sampling is referred to as space-filling DoE in this text.

In addition to the space-filling property, it would be beneficial if samples are randomly distributed. The traditional DoE in Section 3.2 has specific locations of samples. In addition, traditional DoE tends to locate samples on the boundary of the design space, either at corners or on the edges. Therefore, there is a chance that a surrogate might have a good performance at these sample locations but a bad performance at unsampled locations. Considering the fact that the optimal design normally presents at the center of the domain, these boundary samples can cause a bias error in the central region. If a space-filling DoE has randomness in selecting locations, it can provide a robust estimate of prediction accuracy.

The term 'space-filling' might mislead reality. It is appropriate only for low-dimensional spaces. For high-dimensional spaces, we cannot afford to 'fill' the design space. Therefore, it is inevitable to anticipate a large extrapolation region in a high-dimensional design space. In addition, space-filling DoE tends to put samples inside the design space instead of at boundaries. Therefore, in a low-dimensional design space, it would be useful to add additional samples at the corners of the design space. Adding samples at all corners is impractical for high-dimensional design space.

There are many different methods to generate space-filling DoEs. Among them, orthogonal arrays [30] (OA) and Latin Hypercube sampling [31] (LHS) are considered practical options. OA produces uniform designs but can generate particular forms of sample replications, while LHS does not produce replicates but can lack uniformity. As a result, OA-based LHS [32] and other optimal LHS schemes [33, 34] have been proposed. Among many space-filling DoEs, we will only consider Monte-Carlo simulation, LHS, and OA designs in this section.

Monte-Carlo simulation

Monte Carlo simulation (MCS) is a popular method for generating random samples based on a probability distribution. Originally, this is a method to generate random samples for the purpose of uncertainty quantification, but it can be extended for the purpose of DoE. Normally, MCS assumes that a random variable X has a probability density function $f_X(x)$ and generates samples that follow this distribution. The idea to extend MCS for the purpose of DoE is to use a uniform distribution for $f_X(x)$ such that samples are generated uniformly in the design space. Although the idea seems reasonable, in practice, uniformity can be achieved only with a large number of samples. It is likely that some regions will be poorly sampled, and some regions will be overly sampled. In five-dimensional space, for example, it will require $2^5 = 32$ samples if one sample is to be located at each orthant. However, the probability to have one sample at each orthant is

$$\frac{31}{32} \times \frac{30}{32} \times \cdots \times \frac{1}{32} = 1.8 \times 10^{-13}$$

Therefore, it would be very unlikely to have evenly distributed samples.

Example 3-12

Generate 20 samples of $(x_1, x_2) \in [0,1]$ using MCS and plot the marginal histogram. Check if samples are uniformly distributed in the design space.

Solution:

The following Matlab code is used to generate 20 samples from uniform distribution and plot sample locations and marginal histogram of each variable:

```
rng default;
x=rand(20,2);
subplot(2,2,1); plot(x(:,1), x(:,2), 'o');
subplot(2,2,2); hist(x(:,2),20);
subplot(2,2,3); hist(x(:,1),20);
```

Matlab command `rand` generates random samples from a uniform distribution in the interval of $[0, 1]$. Therefore, if the range of input variables is different, it would be necessary to scale the samples appropriately. Figure 3-14 shows the sample locations and marginal histograms. The distribution of points shows both clusterings in the lower-right and upper-left corners and scattering in the lower-left corner. Obviously, the distribution of samples does not look like a uniform distribution. If the samples are uniformly distributed, it is expected that the marginal histograms show a uniform height. The histograms were divided into 20 bins, and therefore, if samples were uniformly distributed, it is expected to have one sample in each bin. However, as shown in the figure, with 20 samples, there is evidence of both clustering and scarcity of samples.

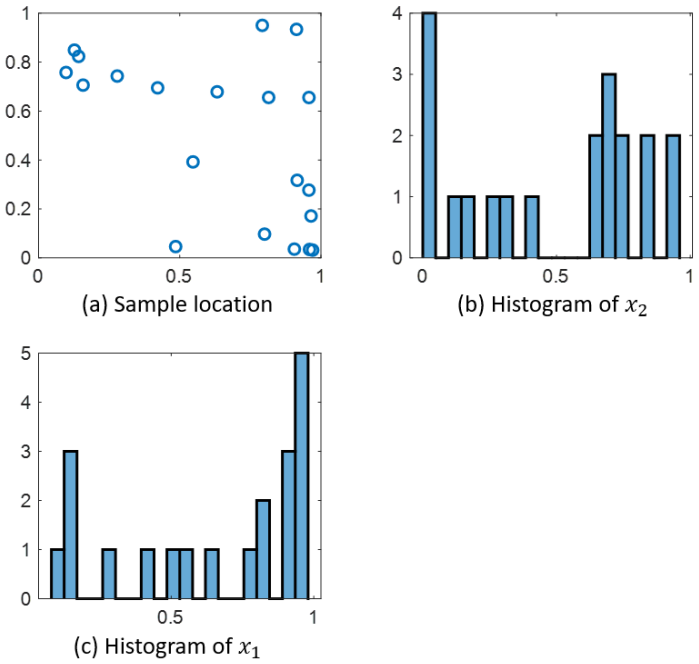


Figure 3-14: Marginal histogram of 20 samples in two-dimensional design space.

Latin hypercube sampling

Latin Hypercube sampling (LHS) is a semi-random sampling scheme, where the design space is gridded, and samples are randomly located within the grid. The design range of each variable is divided by n_y intervals depending on its probability distribution. In the case of space-filling designs (i.e., the design variable is assumed to be uniformly distributed), the design space is divided by uniform intervals of size $1/n_y$. For example, Figure 3-15(a) shows seven LHS samples from a uniform distribution. First, the range of design is divided into seven intervals, and one sample is placed at each interval. However, the sample location is random in the sense that the sample location within an interval is randomly selected.

Although uniform distribution is mostly used here for space-filling DoE, LHS can generate samples from any distribution. For example, Figure 3-15(b) shows ten LHS samples from a normal distribution. In

order to do that, first the range of probability $[0, 1]$ is equally divided by ten intervals (vertical axis). Second, as shown in the figure, the inverse transformation is performed to find the corresponding variable's intervals (horizontal axis). Therefore, all intervals may have different sizes, but they have the same probability. Last, one sample is randomly generated at each interval. If the purpose of the surrogate is to propagate uncertainty from input to output, it makes sense to sample according to the distribution of the input variables. However, for the purpose of optimization, it would be better to fill the design space evenly, and therefore, a uniform distribution makes sense.

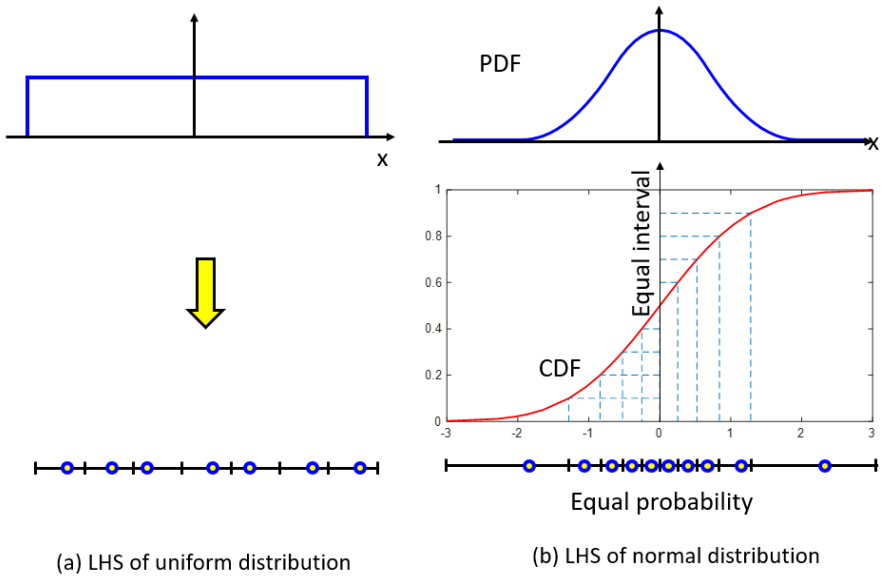


Figure 3-15: Latin hypercube sampling scheme in one dimension.

In the case of a one-dimensional variable, the randomness in LHS is only from the fact that the sample location within an interval is random. However, in the case of multi-dimensional variables, there is another source of randomness in LHS, which is more significant than the first. In the case of two designs with three samples (i.e., $n = 2, n_y = 3$), as shown in Figure 3-16, each variable has three intervals. Because of two variables, the grid has a total of nine cells. Therefore, there is randomness associated with selecting three sample locations out of nine cells. However, this is not a pure combination problem of ${}_3C_9$ because LHS requires one sample at each column or row. More specifically, in the first column of the figure, LHS can select one row out of three. In the second column, LHS can select one row out of two because the row that was selected in the first column cannot be selected. Then for the last column, the row is fixed because two rows are already selected by the previous two columns. Therefore, the possibility of locating samples is $3! = 3 \times 2 \times 1$. This is the major source of uncertainty in LHS.

Often the distribution of sample locations in LHS is defined using a definition table. As shown in Figure 3-16, the first column defines the intervals for x_1, and the second column defines the intervals for x_2 where the sample is located. It is noted that the first column is in a sequence, but the second column is a random permutation of sequences 1,2,3. If there is a third variable, the definition matrix would have one more column with another random permutation of the three numbers. In general, for n_y samples in n variables, the location of the samples will be defined by an $n_y \times n$ definition matrix, with each column a random permutation of $1,2, \ldots, n_y$.

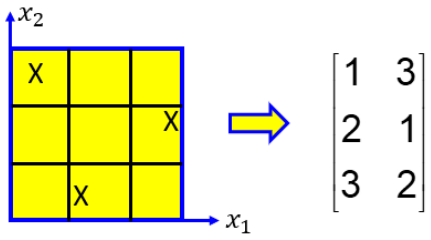


Figure 3-16: Latin hypercube samples and definition matrix ($n = 2, n_y = 3$).

In general, the samples from LHS will be better distributed for each individual variable than that of MCS, but still, there could be a large portion of the design space that is not sampled. An extreme example would be if all the permutations happen to be in a sequence, then all the samples will be aligned on one of the diagonals in the box. Due to the randomness in selecting intervals, there are many possible LHS designs that still satisfy the basic requirement of one sample in each interval of each design variable. Therefore, it makes sense to generate many LHS designs and pick the best one. In order to do that, it would require having criteria to choose the best LHS design. The desirable criteria would be maximizing the minimum size of the empty hyper-sphere (i.e., unsampled region) or minimizing correlation among samples. The first criterion tries to remove clustered samples. Unfortunately, since calculating the minimum size of an empty hyper-sphere is tricky, the minimum distance between samples is often used as an LHS design criterion that is easy and cheap to calculate.

Matlab command `lhsdesign` can either maximize the minimum distance (default option) or minimize correlation. The command iteratively generates samples to improve the criterion. Figure 3-17(a) shows 10 samples using LHS with two variables. The following Matlab code is used to generate samples and plot them. The variable `x` includes samples from the minimum distance option, while the variable `xr` includes samples from the minimum correlation option.

```
x=lhsdesign(10,2);  plot(x(:,1), x(:,2), 'o');
xr=lhsdesign(10,2,'criterion','correlation');
hold on; plot(xr(:,1), xr(:,2), 'r+');
r=corrcoef(x)
%r = 1.0000   -0.6999
%   -0.6999    1.0000
r=corrcoef(xr)
%r = 1.0000   -0.0545
%   -0.0545    1.0000
```

The blue circles in the figure are generated with the default option (maximizing the minimum distance) and have a correlation coefficient of -0.7. The red crosses are obtained by minimizing the correlation option, and it is -0.0545. Note that even though the minimum distance between the circles is larger than between the crosses, the circles have a much larger empty space in the lower-left corner. This is because Matlab uses only five iterations as default for optimizing the design.

Since the five default iterations in Matlab often yield a poor DoE, it makes sense to use more iterations. Figure 3-17(b) shows the results with 5,000 iterations. Maximizing the minimum distance (blue circles) actually reduces the correlation as well, where the correlation is dropped to 0.236. Of course, minimizing the correlation leads to a lower correlation of 0.042. With more iterations, maximizing the minimum distance eliminates the large unsampled region that was present in Figure 3-17(a). On the other hand, even with more iterations, minimizing the correlation did not change the minimum distance or

eliminated the unsampled region. Therefore, for the red crosses, there are still large empty spaces near $(0.45, 0.75)$, $(0,0)$, and $(0,1)$. This explains why the minimum distance is the default criterion in Matlab.

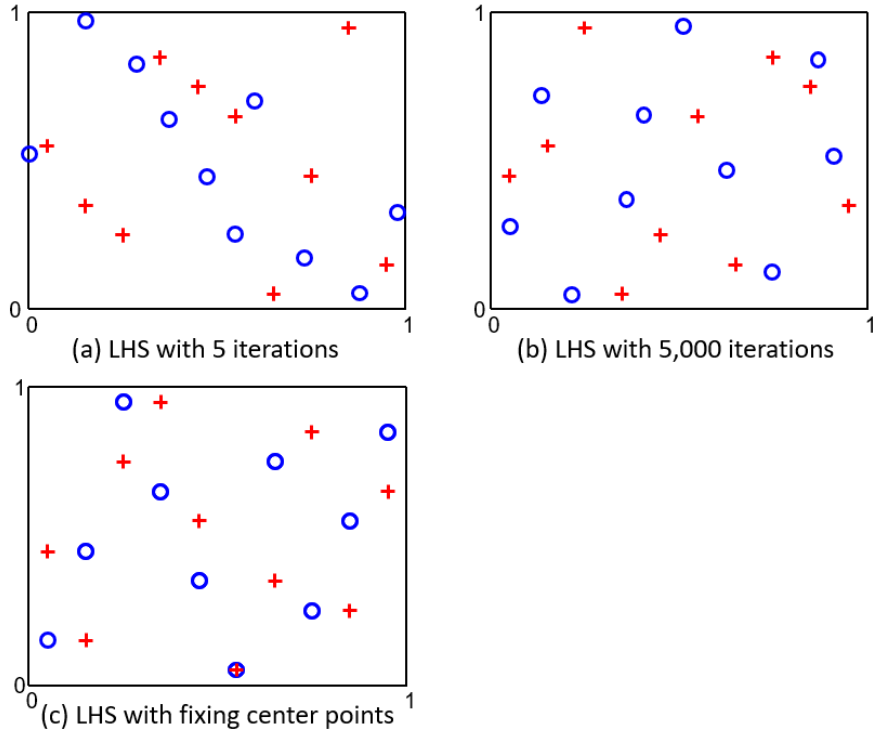


Figure 3-17: Latin hypercube samples with different options ($n = 2, n_y = 10$).

If the sample location within an interval does not have to be random, the randomness of LHS can be reduced by putting the samples at the center of the intervals, using '`smooth`' parameter in `lhsdesign`. Samples in Figure 3-17(c) are generated by the following command: `x=lhsdesign(10, 2, 'iterations',5000,'smooth','off')`. With ten samples, each variable is divided into ten intervals, and the samples are located at the center of these intervals, such as 0.05, 0.15, 0.25, and so on. Therefore, in the figure, the circles (max minimum distance) and crosses (minimum correlation) are aligned, and one point even overlaps.

Orthogonal arrays

Orthogonal arrays are a type of general fractional factorial designs. It is a highly fractional orthogonal design that is based on a design matrix proposed by Dr. Genichi Taguchi [35]. DoE from orthogonal arrays allows an arbitrary number of variables with an arbitrary number of levels. Consider the matrix of samples in Figure 3-18(a), which has four samples $(n_y = 4)$ of three variables $(n = 3)$ with two levels. In the normalized design variables, the two levels mean that a design variable has a value of -1 or $+1$; i.e., the lower- and upper-bound of design space. The notation of an orthogonal array is $\mathrm{OA}_{n_y}(s^n)$, where n_y is the number of samples, n the number of design variables, and s the number of levels. Therefore, the orthogonal array in Figure 3-18(a) can be denoted as $\mathrm{OA}_4(2^3)$. An orthogonal array is written in the form of $n_y \times n$ matrix. For linear PRS, DoE from orthogonal arrays yield an orthogonal design; that is, the moment matrix becomes diagonal.

Experiment No.	Design variable		
	1	2	3
1	-1	-1	-1
2	-1	1	1
3	1	-1	1
4	1	1	-1

(a) $OA_4(2^3)$

Experiment No.	Design variable		
	1	2	3
1	-1	-1	-1
2	-1	0	0
3	-1	1	1
4	0	-1	1
5	0	0	-1
6	0	1	0
7	1	-1	0
8	1	0	1
9	1	1	-1

(b) $OA_9(3^3)$

Figure 3-18: Examples of orthogonal arrays.

Orthogonal arrays are balanced to ensure that all levels of all variables are considered equally. For this reason, the variables can be evaluated independently of each other despite the fractionality of the design. An important property of an orthogonal array is that in every pair of columns each of the possible ordered pairs of elements appears the same number of times. In Figure 3-18(a), for example, in every pair of columns each of the four ordered pairs $(-1,-1)$, $(-1,1)$, $(1,-1)$, and $(1,1)$ appears exactly once. If we choose columns 1 and 2, they are $(-1,-1)$, $(-1,1)$, $(1,-1)$, and $(1,1)$. If we choose columns 1 and 3, they are $(-1,-1)$, $(-1,1)$, $(1,1)$, and $(1,-1)$. Therefore, all possible pairs appear once.

Figure 3-18(b) shows an orthogonal array of $OA_9(3^3)$. In this case, since each variable can have three levels: $-1, 0, +1$, all possible pairs of levels are $(-1,-1), (-1,0), (-1,1), (0,-1), (0,0), (0,1), (1,-1), (1,0), (1,1)$. In every pair of columns, each of the nine ordered pairs appears exactly twice. Therefore, it is an orthogonal array $OA_9(3^3)$.

It is interesting to note that s^n is in fact the number of samples in the s-level full factorial design, while n_y is the actual number of samples in the orthogonal array. Therefore, the ratio (n_y/s^n) is the fraction of full factorial design. For $OA_4(2^3)$ in Figure 3-18(a), the ratio is $4/2^3 = 1/2$ fraction of full factorial design, and for $OA_9(3^3)$ in Figure 3-18(b) the ratio becomes $9/3^3 = 1/3$.

A sub-matrix formed by deleting some columns of an orthogonal array is also an orthogonal array. Thus, by deleting certain columns of a given orthogonal array, it is possible to generate many different plans for multifactor experiments. If the third column in Figure 3-18(a) is deleted, it becomes $OA_4(2^2)$ and it is nothing but the two-level full factorial design. The same is true for Figure 3-18(b), when the third column is removed, it becomes the three-level full factorial design. In practice, since each column represents a design variable, the effect of the orthogonal array will be the same if any single column is removed. In Figure 3-18(a), for example, the resulting orthogonal array by deleting the third column will be identical to that by deleting the first column. They are only different by row indices.

A fractional factorial design that enables uncorrelated estimation of every factorial effect included in the underlying linear model assuming that all other effects are zero is called an orthogonal plan. Fractional factorial designs based on orthogonal arrays irrespective of the degree of fractionation are necessarily orthogonal plans. This is the primary reason for the popularity of fractional factorials based on orthogonal arrays.

Since each row of an orthogonal array represents a sample, exchanging rows would not affect the property of DoE. Also, exchanging columns simply means relabeling variables. It would not affect the orthogonality of DoE either. Lastly, the levels of factors, $-1, 0, +1$, can be changed to different orders, such as $+1, -1, 0$ because all levels will appear the same number of times. Therefore, exchanging levels would not change the property either. In summary, two orthogonal arrays are defined to be equivalent if one can be obtained from the other via the following operations: (1) exchanging rows, (2) exchanging columns, and (3) exchanging labels of levels.

Although orthogonal arrays are a powerful tool to generate effective samples, it has limitations to using them for the purpose of DoE. First, it is not general enough to generate an arbitrary number of samples, variables, and levels. Only a limited number of combinations are available. For a list of available orthogonal arrays, theory and applications, see, for example, Owen [36], Hedayat et al. [37], and references therein. Second, samples in some orthogonal arrays can be repeated. This is undesirable for numerical experiments where the QoI is also repeated. In such a case, it would be necessary to perturb the repeated sample locations slightly.

Example 3-13

Calculate the standard error of prediction for the orthogonal array shown in Figure 3-18(a) when a linear PRS $\hat{y}(x) = b_1 + b_2x_1 + b_3x_2 + b_4x_3$ is fit with the four samples.

Solution:

For the linear PRS, the design matrix and the moment matrix at the four sample locations become

$$\mathbf{X} = \begin{bmatrix} 1 & -1 & -1 & -1 \\ 1 & -1 & 1 & 1 \\ 1 & 1 & -1 & 1 \\ 1 & 1 & 1 & -1 \end{bmatrix}, \qquad \mathbf{X}^T\mathbf{X} = \begin{bmatrix} 4 & 0 & 0 & 0 \\ 0 & 4 & 0 & 0 \\ 0 & 0 & 4 & 0 \\ 0 & 0 & 0 & 4 \end{bmatrix}$$

As expected, since the orthogonal array yields an orthogonal design, the moment matrix becomes diagonal. Using the inverse of the moment matrix, the standard error of prediction can be calculated as

$$\sigma_y(\mathbf{x}) = \hat{\sigma}\sqrt{\boldsymbol{\xi}(\mathbf{x})^T(\mathbf{X}^T\mathbf{X})^{-1}\boldsymbol{\xi}(\mathbf{x})} = \frac{\hat{\sigma}}{2}\sqrt{1 + x_1^2 + x_2^2 + x_3^3}$$

The standard error of prediction has its minimum $\sigma_y^{min} = \hat{\sigma}/2$ at the origin, while its maximum $\sigma_y^{max} = \hat{\sigma}$ at the corner of the design space. It is noted that the standard error of prediction is the same for both sampled or unsampled corners. This happened because the orthogonal array captures the behavior of an individual variable while fixing other variables.

3.5.Review of various designs of experiments

In this chapter, we discussed several different DoE methods. Some DoEs are well structured, while others have randomness embedded in the process. Also, different DoEs have different characteristics, and it is difficult to tell whether one DoE is better than others for all possible configurations. From the perspective of users, it might be necessary to provide a guideline to choose an appropriate DoE for their applications. In this section, different application conditions are considered and DoEs that are useful for such conditions are recommended.

Guideline for selecting designs of experiments

Since the condition of using DoE can be different in terms of the number of samples, the number of variables, and levels. There would not be a single DoE that works best for all possible conditions. In addition, the samples obtained from DoE can have different levels of noise. Therefore, it would be a good idea to find some recommendations for finding a good DoE for the given conditions. The conditions that we need to consider for choosing a good DoE are (a) the level of noise, (b) the number of variables, (c) the number of regression coefficients, and (d) the number of samples. Also, it is possible to consider if all the experiments are conducted simultaneously or sequentially. The latter does not require all sample locations in advance. Sample locations are sequentially determined based on previous sample results.

Low dimension with high noise: The first condition that we want to consider is the case of low dimension with a high level of noise. This corresponds to the case when the QoI depends on two or three input variables, but the experiments include a large level of noise. When the design space is box-like, the full-factorial design or CCD is recommended. This is because the number of required samples is relatively small in low dimensions. Both full factorial design and CCD are good to reduce the prediction variance. When the domain is irregular, either D-optimal or A-optimal design is recommended as these designs are optimized to reduce the uncertainty in coefficients. Since the domain is irregular, an adaptive sampling scheme can be a good strategy to add samples until the uncertainty reaches an acceptable level.

Low dimension with low noise: The second condition is the case of low dimension with a low level of noise. This corresponds to the case when the QoI depends on two or three input variables, but the level of noise in experiments is low. In such a case, the focus is on the accuracy of the surrogate rather than reducing prediction variance. When the design space is box-like, the minimum bias design, LHS design, and orthogonal arrays would be good choices for DoE. The LHS design can fill the low-dimensional space with a small number of samples. The orthogonal arrays can capture the trend of all design variables with a reasonably small number of samples. When the design space is irregular, MCS might be a good choice by removing samples that belong outside of the design space.

High dimension with high noise: This would be considered the most challenging case because it is difficult for samples to fill the design space, and the experimental result at each sample location is not accurate. When the design space is box-like, it would be good to use CCD, block design, and fractional factorial design. CCD would be a good choice if the number of design variables is not too many (i.e., $n = 5\sim7$) because the number of samples would be too many compared to the regression coefficients for large n. For a large number of variables, block design or fractional factorial design are the only practical methods, but both may come with a large extrapolation region. For irregular domains, D-optimal and A-optimal designs are recommended using the adaptive sampling option.

High dimension with low noise: When the number of input variables is large, but simulations and/or experiments are relatively accurate, it is better to use space-filling DoEs, such as LHS design. Especially, it would be good to use maximizing the minimum distance option in LHS design. However, due to the curse of dimensionality, it is inevitable to accept a large portion of the extrapolation region in DoE, which can lead to a large prediction variance.

Good practices of designs of experiments

Although it is possible to choose different DOE methods as we discussed in this section, there are several good practices to follow in general.

1. It is always good to normalize the design space to $-1 \leq x_k \leq 1$ such that the design space is a hyper-cube. Theoretically, it is possible to have an arbitrary range of design variables. However, when the ranges of design variables are significantly different, a numerical difficulty can occur. For example, let us

consider the case that both Young's modulus and Poisson's ratio are design variables. In the MKS unit system, Young's modulus of metal is in the order of 10^{11} Pa, while the Poisson's ratio is $0 < \nu < 0.5$. Such a huge difference makes it difficult for measuring distance between sample locations. Therefore, it would be better to normalize all design variables with the same range using Eq. (3.2). When a design variable does not have lower- and/or upper-bounds, it is still required to select low or high values for the bounds. If a design variable varies different orders of magnitude, it is possible to use logarithmic scale.

2. Similar to the normalization of design variables, it would be better to normalize QoI using Eq. (3.6), where the range of QoI becomes $0 \leq y \leq 1$. If the QoI that is approximated varies over many orders of magnitudes, it would be better to use logarithm of function or similar transformation so that the region of small values of QoI is not ignored due to large values of QoI.

3. If the design space is a box-like domain, it would be better to use a well-known pattern of sampling methods, such as the central composite design or fractional factorial design. These methods are well established along with in-depth error analysis.

4. The most difficult part of design of experiments is the curse of dimensionality. Working with five design variables is much easier than six variables. Therefore, it is recommended to reduce the design domain and the number of design variables as much as possible. Not all variables significantly affect the QoI. Therefore, it would be better to fix those variables that do not change the QoI significantly. Sensitivity analysis or parameter study can be used to identify important/significant variables.

5. As discussed in Chapter 2, the number of samples should be large enough so that the surrogate fitting process satisfies the regression property. In the case of polynomial response surfaces in Chapter 2, it is recommended that the number of samples should be at least two- or three times more than that of unknown model parameters. If the number of samples is not large enough, it is possible that the surrogate fits noise, not the trend. This phenomenon is called over-fitting.

6. The basic assumption of polynomial response surfaces is that the functional form is correct, while samples have noise. Therefore, it would be best to start with an accurate functional form. Any domain knowledge or prior experience would be useful to find an appropriate functional form. Also, it would be a good strategy to start with a high-order polynomial and remove unimportant basis progressively. If the functional form of the surrogate model is significantly different from the true function, the estimated noise standard deviation includes not only the noise in data but also the bias error of the model.

7. When backward elimination process is applied, it would be better to remove those coefficients that have low t-statistics. Also, during the backward elimination process, it would not be a good idea to eliminate more than one coefficient at a time. This is because the other coefficient might be significant after the previous coefficient is eliminated.

3.6.Exercise

1. Consider a surrogate model $\hat{y}(x) = b_1 + b_2(x+1) + b_3(x+2)^2$ with design space $x \in [-2,2]$. Five samples are given to fit the surrogate: $(-2,-1), (-1,-0.05), (0,1), (1,2), (2,3.5)$. (a) Fit the surrogate to the samples as they are. (b) Scale the input variables and output QoIs and then fit the surrogate to the samples. Check if the two surrogates are identical or not.

2. Consider the problem of fitting a linear PRS $\hat{y}(\mathbf{x}) = b_1 + b_2x_1 + b_3x_2 + b_4x_3$ to samples in the square domain $-1 \le x_1, x_2, x_3 \le 1$. Compare the maximum value of the prediction variance of the full factorial design (samples at all eight vertices) with that of two-dimensional linear PRS in **Example 3-2**.
3. Find the maximum prediction variance in the unit cube for a linear PRS, when the samples are given in the four points $(-1,-1,-1), (-1,-1,1), (-1,1,-1), (1,-1,-1)$. Compare this result with that of Exercise Problem 2.
4. In **Example 3-2**, it was shown that the two-level full-factorial design in a two-dimensional problem is an orthogonal design and the prediction variance is minimum at the origin. Show the same conclusion holds for a three-dimensional problem.
5. In a three-dimensional design space, four non-planar samples are the minimum number of samples to define a simplex. Show that the following four samples yield an orthogonal design for a linear PRS and calculate the minimum and maximum standard error of prediction: $\mathbf{x}_1 = (-1,-1,1), \mathbf{x}_2 = (1,-1,-1), \mathbf{x}_3 = (-1,1,-1),\ \mathbf{x}_4 = (1,1,1)$.
6. In one-dimensional linear PRS, show that the two samples $x_1 = -\alpha$ and $x_2 = \alpha$, $\alpha > 0$, produce an orthogonal design. Discuss the value of α when the standard error of prediction can be minimized.
7. In **Example 3-3**, the three samples are given at $(\alpha\sqrt{3/2}, -\alpha/\sqrt{2})$, $(-\alpha\sqrt{3/2}, -\alpha/\sqrt{2})$, $(0, \alpha\sqrt{2})$ with $\alpha \in [0.1, 1.5]$. Show that these samples yield an orthogonal design. Plot the minimum and maximum standard error of prediction as a function of α.
8. For two-dimensional quadratic PRS, use the three-level full-factorial design to calculate the minimum and maximum standard error of prediction and stability ratio.
9. In **Example 3-5**, vary the location of axial point $\alpha \in [1.0,1.7]$ in the increment of 0.1 and plot σ_y^{min} and σ_y^{max} as a function of α when $n_c = 1$.
10. Repeat Problem 9 when $n_c = 5$.
11. For the face-center CCD design shown in Figure 3-7, calculate the minimum and maximum standard error of prediction along with its stability.
12. For the three-dimensional central composite design with $n_c = 1$, calculate the minimum and maximum of the standard error of prediction along with its stability.
13. For the three-dimensional block design shown in Figure 3-8, calculate the minimum and maximum standard error of prediction along with its stability. Compare the results with that of the central composite design in Exercise Problem 12.
14. Find the maximum prediction variance in the unit cube for a linear polynomial, when the data is given in the four points $(1,1,-1), (1,1,1), (1,-1,-1), (-1,1,-1)$.
15. Repeat Problem 14 when samples are given in the four points $(-1,-1,-1), (-1,-1,1), (-1,1,-1), (1,-1,-1)$.
16. When the first data point is given by $(-1,-1)$, find additional two points in the unit square that will minimize the maximum prediction variance in the unit square for a linear response surface.
17. Find the three points in the unit square that will minimize the maximum prediction variance in the unit square for a linear response surface.
18. For **Example 3-11**, find the maximum prediction variance for the minimum bias designs and compare it to that of the full factorial design.
19. Construct a minimum-bias central composite design by finding α that minimizes the maximum standard error of prediction. Use $n_c = 1$.
20. A linear response surface $\hat{y}(\mathbf{x}) = b_1x_1 + b_2x_2$ is fit using the following 4 DOEs: $(-1,-1)$, $(1,-1)$, $(-1,1)$ and $(1,p)$. (a) Determine unknown $-1 \le \mathrm{p} \le 1$ using the D-optimality criterion. (b) Calculate the prediction variance at the center location $(0,0)$.

21. Three LHS samples are given in the form of a definition matrix for two uniformly distributed variables x_1 and x_2. The design space is given as $x_1, x_2 \in [0,1]$. (a) Draw the intervals of the variables and one realization of this design, using the maximum minimum distance criterion to place them well in the cells. (b) Give the coordinates of these points and the value or the criterion.

$$\begin{bmatrix} 1 & 2 \\ 2 & 3 \\ 3 & 1 \end{bmatrix}$$

22. Find the minimum and maximum standard error of prediction for the orthogonal design given in Figure 3-18(b) when a linear PRS is used.
23. Show that the four points: $(\pm 0.1876, \pm 0.7947)$ constitute a minimum bias design for the one-dimensional response surface $y = b_1 + b_2 x + b_3 x^2$ on the interval $(-1,1)$. Assume that the true model is a cubic polynomial.
24. Find the D-optimal design for the one-dimensional response surface $y = b_1 + b_2 x$ on the interval $(-1,1)$ using 3 points. You can assume that 2 of the three points are $x = 1$ and $x = -1$.
25. For a linear model, with tests conducted at three vertices of the unit square $(-1,1), (1,-1), (-1,-1)$, verify that the prediction variance is given as $\mathrm{Var}[y] = \sigma^2(1 + x_1 + x_2 + x_1^2 + x_1 x_2 + x_2^2)/2$. Hint: You do not need to invert the matrix XTX. You can extract that matrix from the expression for the variance so that you will just need to check that it is the correct inverse
26. If the region of interest includes only the triangle defined by the 3 points in Problem 25, formulate the optimization problem of finding the maximum variance in the triangle. Find where the variance is maximal in the triangle and check your answer from the Kuhn-Tucker conditions of the optimization problem you formulated. If you can add one more experiment, where would you add it to reduce the maximum prediction variance in the triangle.
27. You need to fit a linear function $y = b_1 + b_2 x$ to data. You can run experiments in the interval $(-1,1)$, but you are interested only in results in the interval $(-1,0.5)$. For 3 experiments, find the D-optimal design, and find the equations that the coordinates of the points need to satisfy for a minimum bias design. Find the minimum bias design, and compare the two for the function $y = e^x$ in the interval $(-1,0.5)$.
28. A response surface can be constructed for the purpose of predicting the results of future experiments or for the purpose of evaluating some physical constants (that is the coefficients in the response surface could be these constants). In addition, a response surface could be constructed under conditions of high experimental noise or conditions of large modeling errors (the assumed model being different from the true model). Please indicate which optimality criterion you will use for selecting experimental points for each one of the four possible combinations of these conditions: (a) Physical constants, high noise. (b) Physical constants, large modeling error. (c) Predicting future experiments, high noise. (d) Predicting future experiments, large modeling error.