Design and Analysis of Complex Computer Models

Jeevan Jankar, Hongzhi Wang, Lauren Rose Wilkes, Qian Xiao, and Abhyuday Mandal

Abstract This chapter presents a review of some state-of-the-art statistical techniques for analyzing real computer experiments which play a significant role in various scientific research and industrial applications. In computer experiments, emulators (i.e. surrogate models) are often used to rapidly approximate the outcomes and reduce the computational expense. Gaussian process (GP) models, also known as Kriging, are a common choice of emulators, and optimal experimental designs should be used to improve their accuracy. Specifically, space-filling designs are widely used in the literature, which proved to be efficient under GP models. In this chapter, we review different types of GP models as well as various kinds of space-filling designs. We further provide a practical tutorial on how to construct space-filling designs and fit GP emulators to analyze real computer experiments.

Keywords Computer experiments · Gaussian process models · Space-filling designs · Latin hypercube designs

1 Introduction

A computer experiment is a system of complex computer codes simulating a physical process. They are implemented like a function, taking inputs to produce the outputs. This automation can reduce the cost, time, and/or management compared to a traditional lab experiment (see, for example, [20]). Computer experiments are often deterministic (specified inputs will always produce the same output), making the results more stable and less prone to random errors compared to traditional lab experiments. Researchers can manipulate the code to systematically adjust a wide range of inputs and generate outputs based on what they are trying to study. They are instrumental in cases where a physical experiment would be impossible, such as modeling black holes [29]. Due to these characteristics, computer experiments

J. Jankar · H. Wang · L. R. Wilkes · Q. Xiao · A. Mandal (✉)
University of Georgia, Athens, GA 30602, Georgia
e-mail: abhyuday@uga.edu; amandal@stat.uga.edu

R. Srinivas et al. (eds.), *Advances in Computational Modeling and Simulation*,
Lecture Notes in Mechanical Engineering,
https://doi.org/10.1007/978-981-16-7857-8_2

become very popular in various scientific research and industrial applications (see, for more examples, [12, 20]). For example, [8] created a 3D mixed finite element model to study flexoelectric material. The Flexoelectric Effect is where strain gradients polarize electric fields. This process is complicated to study, especially in a practical context, so the finite element method is a numerical approach, i.e. computer experiment, used to study this effect. Mixed finite elements simplify this task further using an alternative way of handling higher order derivatives.

Computer experiments are often computationally intensive, though computing power has increased in recent years. To rapidly generate many outcomes and reduce the computational expenses, emulators (i.e. surrogate models) are needed which are often fitted with only a few data points. Emulators should also allow uncertainty quantification to measure how accurate the model is for predictions. If a good emulator is selected, it may be more useful than the underlying physical process as it eliminates noise. The Gaussian Process (GP) model is a widely used emulator [20, 43]. The GP assumes all observations following a multivariate normal distribution, which is characterized by a mean vector μ and a covariance matrix Σ. The GP model would interpolate the observations, which is desirable for computer experiments having deterministic outputs. It also allows for accurate uncertainty quantification. By specifying different types of covariance functions, researchers may further add prior knowledge about the shape of the response surface.

The GP model has been applied to many computer experiments in Chemistry, Computational Biology, Robotics and others [30]. As an illustration, it has accurately simulated the collision dynamics of complex molecules [6], the spread of COVID-19 [52], flagging suspicious Internet claims [63] and autonomous learning in robots [7]. Data scientists at Microsoft introduced a framework that enables the application of GP models to data sets containing millions of data points [23]. As pictured in Fig. 1, a Bayesian framework is used for human body pose tracking [10]. Instead, a GP experiment can be used to take in a description of a human silhouette as inputs and outputs to identify human pose [68]. One useful application of GP in Astronomy is modeling the collision of two black holes. Researchers cannot create black holes to observe and experiment with, so computer experiments offer a veritable way to simulate the outcome of black hole collisions. Figure 2 illustrates

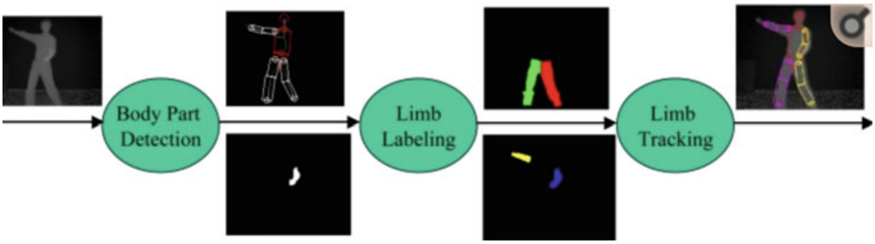

Fig. 1 An example of Bayesian framework for human pose tracking *Source* https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3292173/ [68]

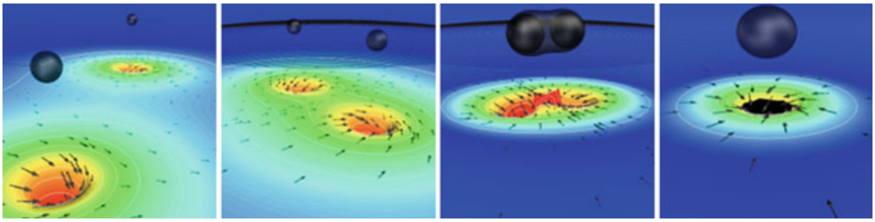

Fig. 2 Computer simulation of two black holes colliding *Source* https://www.black-holes.org/code/SpEC.html

that computer models and GP emulators are created based on the known properties of black holes and the surrounding system of space and are compared to naturally observed black hole movement in order to test how accurate they are [58]. Another interesting application of GP is on car crash simulation to study the damage on the car. Here, models are validated by comparing simulation results with an actually controlled crash. Figure 3 depicts some results from a finite element method.

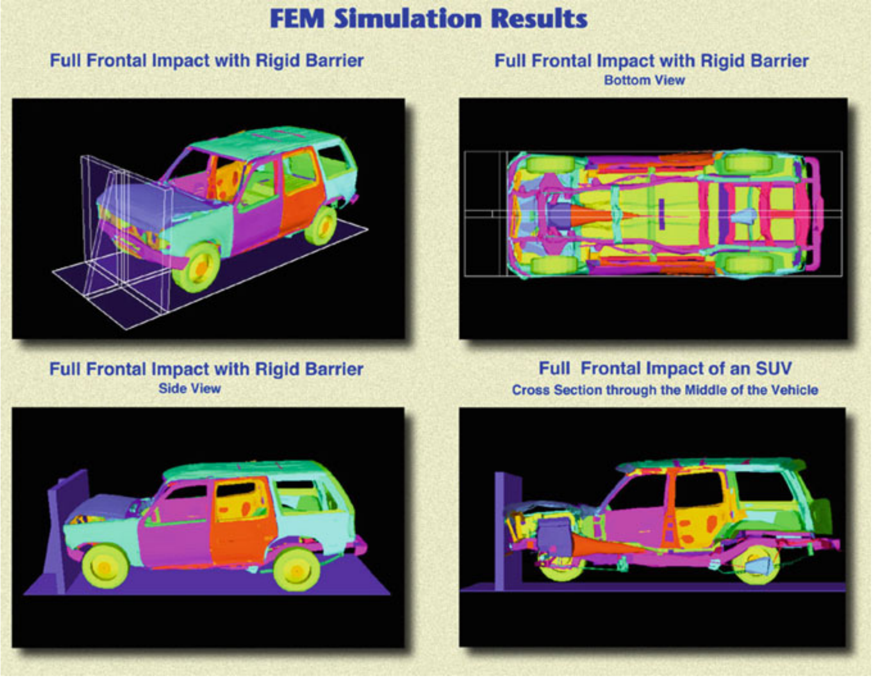
FEM Simulation Results
Full Frontal Impact with Rigid Barrier
Full Frontal Impact with Rigid Barrier
Bottom View
Full Frontal Impact with Rigid Barrier
Side View
Full Frontal Impact of an SUV
Cross Section through the Middle of the Vehicle

Fig. 3 An example of Gaussian Process experiment in car crash simulation *Source* https://www.csm.ornl.gov/SC98/car.html

The remainder of this chapter is organized as follows. In Sect. 2, we systematically review the GP models. Specifically, we discuss the ordinary and universal GP in Sect. 2.1, their model estimations and uncertainty quantification in Sect. 2.2 and methods for including qualitative inputs in Sect. 2.3. In Sect. 3, we review popular experimental designs used in computer experiments, and we conclude this chapter in Sect. 4.

2 The Gaussian Process Model

In this section, we aim to understand GP as a flexible nonparametric regression for surrogate modeling in computer experiments. GP is widely used in many statistical and probabilistic modeling enterprises. GP is a very generic term, and all it means is that any finite collection of realizations is modeled as having a multivariate normal (MVN) distribution. That means, a finite collection of n observations can be completely characterized by their mean vector μ and covariance matrix Σ.

Let $y(\mathbf{x_i})$ be the output which is assumed to be a deterministic real-valued function of the d-dimensional variable $\mathbf{x_i} = (x_{i1}, \dots, x_{id})^T \in D \subset \Re^d$, for $i = 1, 2, \dots, n$. Let $(Y_\mathrm{x})_{\mathrm{x} \in D}$ be a square-integrable random field and y be a realization of $(Y_\mathrm{x})_{\mathrm{x} \in D}$. Let $\mathbf{X} = \{\mathbf{x_1}, \dots, \mathbf{x_n}\}$ be the points where their responses have been observed, which is denoted by $\mathbf{y} = (y(\mathbf{x_1}), \dots, y(\mathbf{x_n}))^T$. The aim of GP is to optimally predict Y_x by a linear combination of the observations $\mathbf{y}$, for any $\mathbf{x} \in D$.

2.1 Model Formulation

Ordinary GP, also known as ordinary Kriging, has the form

$$y(\mathbf{x_i}) = \mu + Z(\mathbf{x_i}), \tag{1}$$

where μ is the mean vector and $Z(\mathbf{x_i})$ is a GP such that $Z(\mathbf{x_i}) \sim GP(0, \sigma^2\Sigma)$. In the above model, $Z(\mathbf{x_i})$ is GP with zero mean, and the covariance function $\phi(\cdot) = \sigma^2\Sigma(\cdot|\theta)$, where $\theta = (\theta_1, \dots, \theta_d)^T$ is the vector of unknown correlation parameters with all $\theta_k > 0$ $(k = 1, \dots, d)$ and Σ is a stationary correlation function that determines the correlation between inputs with parameters θ. The mean of the GP controls the trend, whereas the correlation function controls the smoothness of its sample paths. Power-exponential, Gaussian and Matérn correlation functions are the most widely used ones in the literature.

In the power-exponential correlation structure, the (i, j)th element in the correlation matrix is defined as follows:

$$\Sigma\left(\mathbf{x}_i, \mathbf{x}_j \mid \boldsymbol{\theta}\right) = \prod_{k=1}^{d} \exp\left\{-\theta_k \left|x_{ik} - x_{jk}\right|^{p_k}\right\} \quad \text{for all } i, j, \tag{2}$$

with two inputs $\mathbf{x}_i = (x_{i1}, \ldots, x_{id})^T$ and $\mathbf{x}_j = \left(x_{j1}, \ldots, x_{jd}\right)^T$ and smoothness parameters $p_1, \ldots, p_d$, which lie between 0 and 2, with 0 giving the most rough results and 2 giving the most smooth. If we take $p_k = 2$ for all $k = 1, \ldots, d$, then it results in the popular Gaussian correlation function:

$$\Sigma\left(\mathbf{x}_i, \mathbf{x}_j \mid \boldsymbol{\theta}\right) = \prod_{k=1}^{d} \exp\left\{-\theta_k \left|x_{ik} - x_{jk}\right|^{2}\right\} \quad \text{for all } i, j. \tag{3}$$

The correlation functions of Matérn family is given by

$$\Sigma(\mathbf{h} \mid \boldsymbol{\theta}) = \prod_{k=1}^{d} \frac{1}{\Gamma(v)2^{v-1}} \left(\frac{2\sqrt{v}\,|h_k|}{\theta_k}\right)^{v} K_v\left(\frac{2\sqrt{v}\,|h_k|}{\theta_k}\right), \tag{4}$$

where $v > 0$ is a smoothness parameter, $\Gamma(\cdot)$ is the Gamma function and $K_v(\cdot)$ is the modified Bessel function of order v. Two commonly used orders are $v = 3/2$ and $v = 5/2$.

Different correlation functions mentioned above impose different characteristics for function draws, allowing for different properties when modeling computer models. For example, when using the power-exponential function, all sample paths are infinitely differentiable when $p_k = 2$. For the Matérn correlation function, when we have $d = 1$, all sample paths are $\lceil v \rceil - 1$ differentiable. Hence, v is viewed as a smoothness parameter.

In the literature, two important assumptions are often imposed on the ordinary GP model to effectively analyze computer experiment. One assumption is that the GP is separable [9], which means finite-dimensional distributions can determine sample path properties of function draws which are usually infinite-dimensional. The second important assumption is that the model is stationary. Consider $\{\mathbf{x_1}, \ldots, \mathbf{x_n}\} \in D$ and any $h \in \Re^d$, then a GP model is said to be stationary if the random vectors $(Y(\mathbf{x_1}), \ldots, Y(\mathbf{x_n}))$ and $(Y(\mathbf{x_1} + \mathbf{h}), \ldots, Y(\mathbf{x_n} + \mathbf{h}))$ follow the same distribution. This means that both these random vectors should have the same mean and covariance.

The second assumption is restrictive, and we may need more flexibility while modeling computer experiments. One popular approach is to extend the above ordinary GP model to incorporate a global trend function for the mean part. This extended model is known as *Universal Kriging* which has the form:

$$y(\mathbf{x}) = \mu(\mathbf{x}) + Z(\mathbf{x}), \tag{5}$$

with $\mu(\mathbf{x}) = \mathbf{f}(\mathbf{x})^T \boldsymbol{\beta} = \sum_{s=1}^{m} \beta_s f_s(\mathbf{x})$, where $\mathbf{f}$ is a m-dimensional known function and $\boldsymbol{\beta} = (\beta_1, \ldots, \beta_m)^T$ is a vector of unknown parameters. The idea is to rely on functions in $\mathbf{f}(\mathbf{x})$ to de-trend the process and then model any residual variation as zero mean stationary GP. Taking constant mean $\mathbf{f}(\mathbf{x}) \equiv 1$ results in the ordinary GP model discussed above. The stationary correlation functions discussed above in Eqs. (2) and (4) can also be applied here, that is,

$$Cov\,(Z(\mathbf{x}+\mathbf{h}), Z(\mathbf{x})) = \sigma^2 \Sigma(\mathbf{h}),$$

where correlation function $\Sigma(\mathbf{h})$ is a positive semidefinite function with $\Sigma(\mathbf{0}) = 1$ and $\Sigma(\mathbf{h}) = \Sigma(-\mathbf{h})$.

2.2 Estimation and Uncertainty Quantification

In this section, we present equations used for predicting and quantifying uncertainty on $y(\mathbf{x})$ given observed responses $\mathbf{y} = (y(\mathbf{x_1}), \ldots, y(\mathbf{x_n}))^T$. The question we are trying to answer is: given examples of function in pairs $(\mathbf{x_1}, y(\mathbf{x_1})), \ldots, (\mathbf{x_n}, y(\mathbf{x_n}))$, what random function realizations could explain or could have generated those observed values? In other words, we want to calculate the conditional distribution $(Y(\mathbf{x_1}), \ldots, Y(\mathbf{x_n}))\,|\{(\mathbf{x_1}, y(\mathbf{x_1})), \ldots, (\mathbf{x_n}, y(\mathbf{x_n}))\}$.

Before we calculate the *predictive distribution*, we need to address the key question of how the parameters $\boldsymbol{\beta}$, σ^2 and $\boldsymbol{\theta}$ are estimated from the data $(\mathbf{x}_i, y(\mathbf{x}_i))_{i=1}^n$. The most popular approach for parameter estimation is *maximum likelihood estimation*, and the log-likelihood function under the above assumed GP model can be written as

$$l\left(\boldsymbol{\beta}, \sigma^2, \boldsymbol{\theta}\right) = -\frac{1}{2}\left[n \log \sigma^2 + \log\det \Sigma_\theta + \frac{1}{\sigma^2}\left(\mathbf{y} - \mathbf{F}\boldsymbol{\beta}\right)\Sigma_\theta^{-1}\left(\mathbf{y} - \mathbf{F}\boldsymbol{\beta}\right)\right], \quad (6)$$

where $\det \Sigma_\theta$ is the determinant of the matrix $\Sigma_\theta = \left[\Sigma(\mathbf{x_i}, \mathbf{x_j})\right]_{i=1}^{n}\,{}_{j=1}^{n}$ and $\mathbf{F} = [f_s(\mathbf{x_i})]_{i=1}^{n}\,{}_{s=1}^{m}$. Hence, the MLEs for $(\boldsymbol{\beta}, \sigma^2, \boldsymbol{\theta})$ are the parameter estimates that maximize the above log-likelihood function. ML estimates of $(\boldsymbol{\beta}, \sigma^2)$ for fixed value of $\boldsymbol{\theta}$ can be easily obtained as follows:

$$\hat{\boldsymbol{\beta}}_{\boldsymbol{\theta}} = \left(\mathbf{F}^T \Sigma_{\boldsymbol{\theta}}^{-1} \mathbf{F}\right)^{-1} \mathbf{F}^T \Sigma_{\boldsymbol{\theta}}^{-1} \mathbf{y} \quad (7)$$

and

$$\hat{\sigma}_\theta^2 = \frac{1}{n}\left(\mathbf{y} - \mathbf{F}\hat{\boldsymbol{\beta}}_{\boldsymbol{\theta}}\right)^T \Sigma_{\boldsymbol{\theta}}^{-1}\left(\mathbf{y} - \mathbf{F}\hat{\boldsymbol{\beta}}_{\boldsymbol{\theta}}\right). \quad (8)$$

Substituting these ML estimates back into Eq. (6), we get the profile likelihood function as follows:

$$l\left(\hat{\boldsymbol{\beta}}, \hat{\sigma}^2, \boldsymbol{\theta}\right) = -\frac{1}{2}\left[n \log \hat{\sigma}^2 + \log \det \Sigma_\theta + n\right], \tag{9}$$

where the MLE of $\boldsymbol{\theta}$ is one that maximizes the above function in Eq. (9). This optimization problem does not enjoy a closed-form solution, so numerical methods, e.g. quasi-Newton algorithms [40] are used for solving the problem.

Once we have estimates of parameters, we can calculate the conditional distribution as mentioned above. Let $\left(\hat{\boldsymbol{\beta}}, \hat{\sigma}^2, \hat{\boldsymbol{\theta}}\right)$ denote the ML estimates of unknown parameters for the given GP model. Then for a new input $\mathbf{x}^* \in \Re^d$, the mean and variance of random variable $Y(\mathbf{x}^*|\mathbf{y})$ are as follows:

$$\hat{y}\left(\mathbf{x}^*\right) = \mathbb{E}\left[Y\left(\mathbf{x}^*\right) \mid \mathbf{y}\right] = \mathbf{f}^T\left(\mathbf{x}^*\right)\hat{\boldsymbol{\beta}} + \mathbf{r}_{\hat{\boldsymbol{\theta}}}^T\left(\mathbf{x}^*\right)\Sigma_{\hat{\boldsymbol{\theta}}}^{-1}\left(\mathbf{y} - \mathbf{F}\hat{\boldsymbol{\beta}}\right), \tag{10}$$

$$s\left(\mathbf{x}^*\right)^2 = \mathrm{Var}\left[Y\left(\mathbf{x}^*\right) \mid \mathbf{y}\right] = \hat{\sigma}^2\left(1 - \mathbf{r}_{\hat{\boldsymbol{\theta}}}^T\left(\mathbf{x}^*\right)\Sigma_{\hat{\boldsymbol{\theta}}}^{-1}\mathbf{r}_{\hat{\boldsymbol{\theta}}}\left(\mathbf{x}^*\right)\right), \tag{11}$$

where the covariance vector $\mathbf{r}_{\hat{\boldsymbol{\theta}}}\left(\mathbf{x}^*\right) = \left[\Sigma_{\hat{\boldsymbol{\theta}}}\left(\mathbf{x}^*, \mathbf{x}_1\right), \Sigma_{\hat{\boldsymbol{\theta}}}\left(\mathbf{x}^*, \mathbf{x}_2\right), \ldots, \Sigma_{\hat{\boldsymbol{\theta}}}\left(\mathbf{x}^*, \mathbf{x}_n\right)\right]^{\mathrm{T}}$.

When some observed data points are very close to each other, the covariance matrix $\Sigma_{\hat{\boldsymbol{\theta}}}$ may become nearly singular, making it difficult to obtain a stable inverse matrix $\Sigma_{\hat{\boldsymbol{\theta}}}^{-1}$. This is a common issue for GP models, when the run and/or factor sizes are large. One way to deal with this problem is to add a positive scalar λ, called the *nugget* parameter, to the diagonal elements in $\Sigma_{\hat{\boldsymbol{\theta}}}$, i.e. replacing $\Sigma_{\boldsymbol{\theta}}$ with $\Sigma_{\boldsymbol{\theta}} + \lambda\mathbf{I}$, where $\mathbf{I}$ is an identity matrix. Adding λ is analogous to adding the ridge parameter in ridge regression, which helps in moving the smallest eigenvalue of $\Sigma_{\boldsymbol{\theta}}$ away from zero, thus stabilizing the calculation of its inverse.

For large data sizes, the estimation of GP models can be very time-consuming, mainly due to the matrix inverse calculation of order $O(n^3)$. To deal with this problem, [21] proposed a localize GP (LaGP) approach. Based on a local subset of the data, they provide a family of local sequential design schemes that defines the support points of a GP predictor. The idea is to make sure that for a given choice of covariance structure, the data points far from the target location $\mathbf{x}^*$ will have little effect on the prediction. Hence, it is not unwise to calculate the inverse of the full covariance matrix, as the elements corresponding to "far away" points will contribute very little to predicting $y(\mathbf{x}^*)$. Interested readers may refer to [21] for further details.

The notion of calibration and sensitivity analysis is important in the context of physical and computer experiments. In practice, we only observe response y_{Field} instead of observing real physical response y_{Real}. And, we use the above computer models to approximate y_{Real} as y_{Model}. Now, as we saw in the earlier sections apart from input variables, computer models also use some more parameters known as calibration parameters to fine-tune the model. Covariance parameters $\boldsymbol{\theta}$ are one such example of calibration parameters. A Bayesian framework was proposed by [28] to address this as follows:

$$y_{Real}(\mathbf{x}) = y_{Model}(\mathbf{x}, \boldsymbol{\theta}) + \mathrm{b}(\mathbf{x})$$
$$y_{Field}(\mathbf{x}) = y_{Model}(\mathbf{x}, \boldsymbol{\theta}) + \mathrm{b}(\mathbf{x}) + \epsilon,$$

where $\mathrm{b}(\mathbf{x})$ is a bias and ϵ is the normal error. Reference [28] used Bayesian methods to estimate the bias correction function and unknown calibration parameter $\boldsymbol{\theta}$ under a GP prior. Iterative history matching algorithm as one proposed by [53] for calibrating a galaxy formation model called GALFORM is an alternative to this Bayesian approach. Recently, [1] used this algorithm for calibrating hydrological time-series models.

2.3 GP with Qualitative Inputs

The above-mentioned GP model is valid only with quantitative inputs, but there are many situations in real life where inputs can be both quantitative and qualitative. One straightforward way to adapt GP models with qualitative inputs is to construct separate GP models for each level combination of the qualitative factors. Yet, when there are many high-level qualitative factors, such an approach would require many observations to fit a large number of GP models. In the current literature, many integrated GP models for both quantitative and qualitative factors are proposed [22, 41, 50, 65, 66].

Reference [60] proposed a new method called EzGP to deal with such problems. Let the kth input of the computer emulator be $\mathbf{w}_k = \left(\mathbf{x}_k^T, \mathbf{z}_k^T\right)^T$, where $\mathbf{x}_k = \left(x_{k1}, \dots, x_{kp}\right)^T$ is the continuous part of input as mentioned in the previous sections and $\mathbf{z}_k = \left(z_{k1}, \dots, z_{kq}\right)^T \in \mathbb{N}^q$ is the qualitative part of the input, where $k = 1, \dots, n$. The EzGP method is inspired by the idea of Analysis of Variance (ANOVA) where quantitative and qualitative inputs are jointly modeled as follows:

$$y(\mathbf{w}) = \mu + Z_{\mathbf{z}}(\mathbf{x}), \tag{12}$$

which suggests that for any given level combination of qualitative factors, $y(\mathbf{w})$ is a GP. Specifically, they considered the following additive model structure:

$$Z_{\mathbf{z}}(\mathbf{x}) = Z_0(\mathbf{x}) + Z_{z^{(1)}}(\mathbf{x}) + \cdots + Z_{z^{(q)}}(\mathbf{x}), \tag{13}$$

where Z_0 and $Z_{z^{(h)}}$ for $h = 1, \dots, q$ are independent GPs with mean zero and some covariance functions. Here, Z_0 plays the role of base GP which takes only quantitative inputs reflecting the intrinsic relation between y and $\mathbf{x}$, and other GPs $Z_{z^{(h)}}$ are the adjustments made to the base GP to reflect the impact of each qualitative factor $z^{(h)}$ for $h = 1, \dots, q$. The EzGP method can easily deal with heterogeneity in computer models with multiple qualitative factors. Two variants in EzGP are proposed to fit data with high dimensionality or large run sizes, which can achieve high computational efficiency.

3 Designs for Computer Experiments

Computer codes generate outputs in a deterministic manner in computer experiments, meaning the same input returns the same output (no random errors). Latin hypercube designs (LHDs, [38]) are the most popular experimental designs in computer experiments. An n runs and k factors LHD is an $n \times k$ matrix with each column being a random permutation of numbers $1, \ldots, n$. LHDs do not have replicates in each one-dimensional projection. There are various types of optimal LHDs for practical needs, including space-filling LHDs, maximum projection LHDs and orthogonal LHDs.

When we have little or no information about the response surface, it is desirable to have design points as scattered out as possible in the design space for better exploration. Despite LHDs having a uniform one-dimensional projection property, random LHDs may have poor space-filling properties over the entire design space. Figure 4 is an illustrative example with two LHD designs. The LHD in the left panel is concentrated almost entirely on the diagonal, which clearly does not explore the input space sufficiently. The design points in the right panel are scattered out over the entire design space, so this design may provide more reliable information. The maximin distance criterion [25] is a widely used metric for measuring the space-filling property of LHDs. It aims to maximize the minimum distances between design points. Let $\mathbf{X}$ denote an LHD matrix, where the L_q-distance between two runs x_i and x_j of $\mathbf{X}$ is given by $d_q(x_i, x_j) = \left\{\sum_{k=1}^{m} |x_{ik} - x_{jk}|^q\right\}^{1/q}$, where q is an integer. Two popular choices are $q = 1$ (i.e. the Manhattan distance) and $q = 2$ (i.e. the Euclidean distance). The maximin L_q-distance design has the maximized minimum L_q-distance, i.e. $\max \min d_q(x_i, x_j)$, where $1 \leq i < j \leq n$. Reference [24, 39] further proposed a scalar value to evaluate the maximin distance criterion:

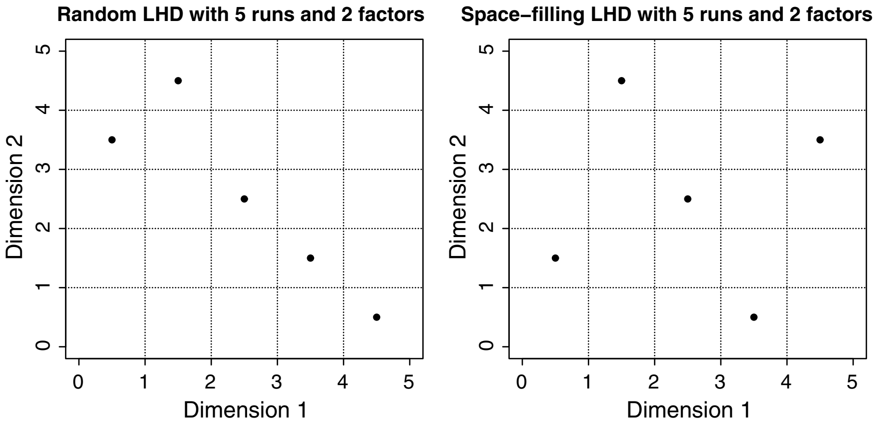

Fig. 4 Latin hypercube designs for size $n = 5$ and $k = 2$

$$\phi_p = \left\{ \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} d_q(x_i, x_j)^{-p} \right\}^{1/p}, \tag{14}$$

where p is a tuning parameter. As $p \to \infty$, the ϕ_p criterion in Eq. (14) is asymptotically equivalent to the Maximin distance criterion, and $p = 15$ is usually sufficient in practice. The LHDs that minimize the ϕ_p criterion are called the maximin distance LHDs.

In the literature, both algebraic constructions [56, 67] and search algorithms [3, 24, 27, 31, 32, 39] are proposed to construct maximin distance LHDs. Algebraic constructions usually require very little computational cost to generate optimal LHDs, which are very attractive for large design sizes. Yet, they are only available for certain design sizes. Search algorithms can generate optimal designs of flexible sizes, but they often require more computation resources to identify optimal LHDs. As there are $(n!)^{k-1}$ possible LHDs with n runs and k factors, search algorithms could become very costly when n and k are large. Here, we will briefly survey some popular construction methods; see [55] for a survey.

Specifically, [56] proposed to generate maximin distance LHDs via good lattice point (GLP) sets [67] and Williams transformation [59]. They proved that the resulting designs of sizes $n \times (n-1)$ (with n being any odd prime) and $n \times n$ (with $2n + 1$ or $n + 1$ being odd prime) are optimal under the maximin L_1-distance criterion. The construction method starts by generating a GLP design, and then use the Williams transformation [59] to improve a linear permuted GLP design. Reference [51] proposed to construct orthogonal array-based LHDs (OALHDs) from existing orthogonal arrays (OAs). The key idea of this construction is to deterministically replace OA entries with a random permutation of LHD elements. OALHDs inherit the properties of OAs and tend to have better space-filling properties compared to random LHDs. Note that the design sizes of OALHDs rely on the existence of corresponding OAs.

Search algorithms should be used to generate optimal LHDs when no construction methods are available. Reference [39] proposed a simulated annealing (SA) algorithm, which randomly exchanges elements to seek improvements over iterations to identify global best LHDs. Following the work of [39] and [51], [31] proposed to construct orthogonal array-based LHDs (OALHDs) using the SA algorithm. They proposed to exchange elements that share the same original OA entry randomly. Reference [27] proposed a multi-objective criterion and developed a modified SA algorithm to generate optimal LHDs having good space-filling properties as well as orthogonality. This algorithm can lead to many good designs, but it is often computationally heavy, since it calculates all average pairwise correlations and row-wise distances at each iteration. Besides these SA-based algorithms, [32] proposed to use a genetic algorithm (GA) for searching optimal designs, which focuses on global best by exchanging random columns between global best and other candidate solutions. In addition, [3] proposed a version of the particle swarm optimization (PSO) algorithm by gradually reducing the Hamming distances between each particle and its personal best (or the global best). Generally speaking, the PSO is recommended for small design sizes ($n \leq 7$) and the GA has better performance for moderate and large design sizes.

Uniform designs (UDs) [11, 13] are another popular type of space-filling designs. There are various measurements of uniformity proposed in the literature, such as the star L_2-discrepancy [57], modified L_2-discrepancy [14] and the centered L_2-discrepancy [15]. The search algorithms mentioned above can be used for identifying UDs.

Maximin distance LHDs have space-filling properties in the full-dimensional space, but their two to $k-1$-dimensional projections may not be space-filling. Reference [26] proposed the maximum projection LHDs (MaxPro LHDs) which enhance the space-filling properties in all possible dimensional projections. Analogous to [39], [26] defined the maximum projection criterion as

$$\min_{\mathbf{X}} \psi(\mathbf{X}) = \left\{ \frac{1}{\binom{n}{2}} \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} \frac{1}{\Pi_{l=1}^{k} (x_{il} - x_{jl})^2} \right\}^{1/k}. \tag{15}$$

LHDs that minimize the ψ values are called MaxPro LHDs. Reference [26] proposed an SA-based search algorithm to identify MaxPro LHDs.

Orthogonal LHDs (OLHDs) are another type of optimal LHDs which aim to minimize the correlations between factors [16, 45, 48]. Two correlation-based criteria are often used to measure designs' orthogonality: the average absolute correlation criterion and the maximum absolute correlation criterion [16], which are defined as

$$\text{ave}(|q|) = \frac{2\sum_{i=1}^{k-1}\sum_{j=i+1}^{k} |q_{ij}|}{k(k-1)} \text{ and } \max|q| = \max_{i,j} |q_{ij}|, \tag{16}$$

where q_{ij} is the correlation between the ith and jth columns in the design matrix. Orthogonal designs may not exist for all sizes. In practice, designs with small ave($|q|$) or max$|q|$ are preferred.

In the literature, construction methods of OLHDs are widely explored. Specifically, [62] proposed a method to construct OLHDs with run sizes $n = 2^m + 1$ and factor sizes $k = 2m - 2$, where m is any integer no less than 2. Reference [5] extended the work of [62] to accommodate more factors. Reference [45] developed a method based on factorial designs with group rotations for $n = 2^{2^m}$ and $k = 2^m t$, where m is any positive integer and t is the number of rotation groups. Reference [47] improved their earlier work [46] to construct OLHDs with even more flexible run sizes: $n = r2^{c+1}$ or $n = r2^{c+1} + 1$ and $k = 2^c$, where c and r are any two positive integers. Reference [61] proposed to use generalized orthogonal designs to construct OLHDs and nearly orthogonal LHDs (NOLHDs) with $n = 2^{r+1}$ or $n = 2^{r+1} + 1$ and $k = 2^r$, where r is any positive integer. Reference [17] proposed to take advantage of orthogonal matrices and their full fold-overs for constructing OLHDs with $n = 2ak$ runs and k factors, where k is the size of orthogonal matrix and a is any positive integer. Reference [2] implemented the Williams transformation [59] to construct OLHDs with odd prime run-size n and factor-size $k \leq n - 1$. Reference [33] proposed to couple OLHDs or NOLHDs with OAs to accommodate large numbers of factors with fewer runs: n^2 runs and $2fp$ factors, where n and p are design sizes of the OLHDs or NOLHDs and $2f$ is the number of columns in the coupled OA.

4 Discussion

There are many instances in nature where it is either expensive or impossible to conduct a physical experiment. For example, it is prohibitively difficult to conduct a study for investigating the devastation caused by a nuclear explosion. Instances like the formation of a galaxy or the formation of binary black holes cannot be studied through physical experiments. Computer experiments can simulate such phenomena with reasonable accuracy. Although such computer simulators are a lot more desirable than real experiments, they are still computationally expensive. To deal with this problem, scientists use surrogates (emulators) to facilitate the analysis and optimization of complex systems. GPs are widely used as surrogates (or emulators). Space-filling designs, such as LHDs, are often used to reap the benefits of utilizing such surrogates effectively.

Several efficient packages in R are available for fitting the GP model and identifying LHDs. Interested readers can explore different packages for fitting GP: Local Approximate Gaussian Process Regression (laGP) by [19], DiceKriging (Kriging Methods for Computer Experiments) by [42] and GP-fit (Gaussian Processes Modeling) by [36]. For obtaining LHDs with flexible run sizes, packages like Latin Hypercube Designs (LHD) by [54] and Maximin-Distance (Sliced) Latin Hypercube Designs (SLHD) by [44] can be used.

Even though the computing power has increased dramatically over the last few years, handling big data remains a challenging problem. There is an increasing body of literature for computer experiments with large numbers of data points, but the existing literature on large numbers of input variables is still meager. For details, please refer to the review article by [35]. The problem of data reduction is an active area of research among statisticians and computer scientists, and much progress needs to be done in this area. Recent work on this includes techniques like kernel handling [4] and support points [37].

Different Bayesian approaches for analyzing computer experiments have been discussed in the literature, particularly in the context of uncertainty quantification, but most of them are difficult to implement and time-consuming [18, 28]. To solve this problem, we need more advanced techniques. Another topic of active research is to incorporate qualitative input variables. Many practical applications have both quantitative and qualitative inputs, e.g. the data center computer experiment [41] and the study of high-performance computing systems [64]. However, traditional GP modeling is designated for only quantitative inputs, since its covariance function of responses is constructed under the continuous input space with proper distance metrics. More effective techniques and algorithms need to be developed that can accommodate qualitative inputs and one such recent work is [60].

Finally, there is vast existing literature on continuous response, but there are many instances where the response is binary or non-continuous. For example, binary black hole formation [34] or computer experiments with binary time series have non-Gaussian observations [49]. For handling high-dimensional input parameter space, input variables with non-continuous characteristics and non-Gaussian observations, new techniques and algorithms need to be developed.

References

1. Bhattacharjee NV, Ranjan P, Mandal A, Tollner E (2019) A history matching approach for calibrating hydrological models. Environ Ecol Stat 26:87–105
2. Butler NA (2001) Optimal and orthogonal Latin hypercube designs for computer experiments. Biometrika 88(3):847–857
3. Chen R-B, Hsieh D-N, Hung Y, Wang W (2013) Optimizing Latin hypercube designs by particle swarm. Stat Comput 23(5):663–676
4. Chen Y, Welling M, Smola A (2010) Super-samples from kernel herding. In: UAI
5. Cioppa TM, Lucas TW (2007) Efficient nearly orthogonal and space-filling Latin hypercubes. Technometrics 49(1):45–55
6. Cui J, Krems RV (2015) Gaussian process model for collision dynamics of complex molecules. Phys Rev Lett 115(7)
7. Deisenroth M, Fox D, Rasmussen C (2015) Gaussian processes for data-efficient learning in robotics and control. IEEE Trans Pattern Anal Mach Intell 37:408–423
8. Deng F, Deng Q, Shen S (2018) A three-dimensional mixed finite element for flexoelectricity. J Appl Mech 85(3)
9. Doob JL, Doob JL (1953) Stochastic Processes. Wiley, Probability and Statistics Series
10. Ek CH, Torr PH, Lawrence ND (2008) Gaussian process latent variable models for human pose estimation. In: Popescu-Belis A, Renals S, Bourlard H (eds) Machine learning for multimodal interaction. Berlin, Heidelberg, Springer Berlin Heidelberg, pp 132–143
11. Fang K-T (1980) The uniform design: application of number-theoretic methods in experimental design. Acta Math Appl Sin 3(4):9
12. Fang KT, Li R, Sudjianto A (2005) Design and modeling for computer experiments. Chapman and Hall/CRC
13. Fang KT, Li R, Sudjianto A (2005) Design and modeling for computer experiments. CRC Press
14. Fang KT, Lin DK, Winker P, Zhang Y (2000) Uniform design: theory and application. Technometrics 42(3):237–248
15. Fang K-T, Ma C-X, Winker P (2002) Centered L_2-discrepancy of random sampling and Latin hypercube design, and construction of uniform designs. Math Comput 71(237):275–296
16. Georgiou SD (2009) Orthogonal Latin hypercube designs from generalized orthogonal designs. J Stat Plan Inference 139(4):1530–1540
17. Georgiou SD, Efthimiou I (2014) Some classes of orthogonal Latin hypercube designs. Stat Sin 24(1):101–120
18. Gramacy RB, Lee HK (2007) Bayesian treed gaussian process models with an application to computer modeling. J Am Stat Assoc 103:1119–1130
19. Gramacy RB (2019) laGP: large-scale spatial modeling via local approximate gaussian processes in R. J Stat Softw 72(1):1–46
20. Gramacy RB (2020) Surrogates: gaussian process modeling, design and optimization for the applied sciences. Chapman Hall/CRC, Boca Raton, Florida. http://bobby.gramacy.com/surrogates/
21. Gramacy RB, Apley DW (2015) Local gaussian process approximation for large computer experiments. J Comput Graph Stat 24(2):561–578

22. Han G, Santner TJ, Notz WI, Bartel DL (2009) Prediction for computer experiments having quantitative and qualitative input variables. Technometrics 51(3):278–288
23. Hensman J, Fusi N, Lawrence ND (2013) Gaussian processes for big data. In: Technical report UAI-P-2013-PG-282-290, twenty-ninth conference on uncertainty in artificial intelligence (UAI2013)
24. Jin R, Chen W, Sudjianto A (2005) An efficient algorithm for constructing optimal design of computer experiments. J Stat Plan Inference 134(1):268–287
25. Johnson ME, Moore LM, Ylvisaker D (1990) Minimax and maximin distance designs. J Stat Plan Inference 26(2):131–148
26. Joseph VR, Gul E, Ba S (2015) Maximum projection designs for computer experiments. Biometrika 102(2):371–380
27. Joseph VR, Hung Y (2008) Orthogonal-maximin Latin hypercube designs. Stat Sin 171–186
28. Kennedy MC, O'Hagan A (2001) Bayesian calibration of computer models. J R Stat Soc: Series B (Statistical Methodology) 63(3):425–464
29. Kidder LE, Scheel MA, Teukolsky SA, Carlson ED, Cook GB (2000) Black hole evolution by spectral methods. Phys Rev D 62:084032
30. Kruckow M, Tauris T, Langer N, Kramer M, Izzard R (2018) Progenitors of gravitational wave mergers: binary evolution with the stellar grid-based code combine. Mon Not R Astron Soc 481:1908–1949
31. Leary S, Bhaskar A, Keane A (2003) Optimal orthogonal-array-based Latin hypercubes. J Appl Stat 30(5):585–598
32. Liefvendahl M, Stocki R (2006) A study on algorithms for optimization of Latin hypercubes. J Stat Plan Inference 136(9):3231–3247
33. Lin CD, Mukerjee R, Tang B (2009) Construction of orthogonal and nearly orthogonal Latin hypercubes. Biometrika 96(1):243–247
34. Lin L, Bingham D, Broekgaarden F, Mandel I (2021) Uncertainty quantification of a computer model for binary black hole formation
35. Liu H, Ong YS, Shen X, Cai J (2020) When gaussian process meets big data: a review of scalable GPS. IEEE Trans Neural Netw Learn Syst 31:4405–4423
36. MacDonald B, Ranjan P, Chipman H (2015) GPfit: an R package for fitting a gaussian process model to deterministic simulator outputs. J Stat Softw 64(12):1–23
37. Mak S, Joseph VR (2018) Support points. Ann Stat
38. McKay MD, Beckman RJ, Conover WJ (1979) Comparison of three methods for selecting values of input variables in the analysis of output from a computer code. Technometrics 21(2):239–245
39. Morris MD, Mitchell TJ (1995) Exploratory designs for computational experiments. J Stat Plan Inference 43(3):381–402
40. Nocedal J, Wright S (2006) Numerical optimization. Springer series in operations research and financial engineering. Springer, New York
41. Qian PZ, Wu H, Wu CJ (2008) Gaussian process models for computer experiments with qualitative and quantitative factors. Technometrics 50(3):383–396
42. Roustant O, Ginsbourger D, Deville Y (2012) DiceKriging, DiceOptim: two R packages for the analysis of computer experiments by kriging-based metamodeling and optimization. J Stat Softw 51(1):1–55
43. Sacks J, Welch WJ, Mitchell TJ, Wynn HP (1989) Design and analysis of computer experiments. Stat Sci 409–423
44. Brenneman WA, Ba S, Myers WR (2015) Slhd: maximin-distance (sliced) Latin hypercube designs. Technometrics 57(4):479–487
45. Steinberg DM, Lin DK (2006) A construction method for orthogonal Latin hypercube designs. Biometrika 93(2):279–288
46. Sun F, Liu MQ, Lin DK (2009) Construction of orthogonal Latin hypercube designs. Biometrika 96(4):971–974
47. Sun F, Liu MQ, Lin DK (2010) Construction of orthogonal Latin hypercube designs with flexible run sizes. J Stat Plan Inference 140(11):3236–3242

48. Sun F, Tang B (2017) A general rotation method for orthogonal Latin hypercubes. Biometrika 104(2):465–472
49. Sung CL, Hung Y, Rittase W, Zhu C, Jeff Wu CF (2017) A generalized gaussian process model for computer experiments with binary time series. J Am Stat Assoc 115:945–956
50. Swiler LP, Hough PD, Qian P, Xu X, Storlie C, Lee H (2014) Surrogate models for mixed discrete-continuous variables. In: Constraint programming and decision making, Studies in computational intelligence, pp 181–202. Springer
51. Tang B (1993) Orthogonal array-based Latin hypercubes. J Am Stat Assoc 88(424):1392–1397
52. Velásquez RM, Lara JV (2020) Forecast and evaluation of covid-19 spreading in USA with reduced-space gaussian process regression. Chaos Solitons Fractals 136:109924
53. Vernon I, Goldstein M, Bower R (2009) Galaxy formation: a bayesian uncertainty analysis. Bayesian Anal 5:619–669
54. Wang H, Xiao Q, Mandal A (2020) LHD: Latin Hypercube Designs (LHDs) algorithms. R package version 1.2.0
55. Wang H, Xiao Q, Mandal A (2021) Musings about constructions of efficient Latin hypercube designs with flexible run-sizes
56. Wang L, Xiao Q, Hongquan X (2018) Optimal maximin L_1-distance Latin hypercube designs based on good lattice point designs. Ann Stat 46(6B):3741–3766
57. Warnock TT (1972) Computational investigations of low-discrepancy point sets. In: Applications of number theory to numerical analysis, pp 319–343. Elsevier
58. Williams D, Heng IS, Gair J, Clark JA, Khamesra B (2020) Precessing numerical relativity waveform surrogate model for binary black holes: a gaussian process regression approach. Phys Rev D 101:063011. https://journals.aps.org/prd/abstract/10.1103/PhysRevD.101.063011
59. Williams EJ (1949) Experimental designs balanced for the estimation of residual effects of treatments. Aust J Chem 2(2):149–168
60. Xiao Q, Mandal A, Lin CD, Deng X (2021) Ezgp: easy-to-interpret gaussian process models for computer experiments with both quantitative and qualitative factors. SIAM/ASA J Uncertain Quantif 9(2):333–353
61. Yang J, Liu MQ (2012) Construction of orthogonal and nearly orthogonal Latin hypercube designs from orthogonal designs. Stat Sin 433–442
62. Ye KQ (1998) Orthogonal column Latin hypercubes and their application in computer experiments. J Am Stat Assoc 93(444):1430–1439
63. Zhang MM, Dumitrascu B, Williamson SA, Engelhardt BE (2019) Sequential gaussian processes for online learning of nonstationary functions. arXiv:abs/1905.10003
64. Zhang Q, Chien P, Liu Q, Xu L, Hong Y (2020) Mixed-input gaussian process emulators for computer experiments with a large number of categorical levels. J Qual Technol 1–11
65. Zhang Y, Tao S, Chen W, Apley DW (2020) A latent variable approach to gaussian process modeling with qualitative and quantitative factors. Technometrics 62(3):291–302
66. Zhou Q, Qian PZ, Zhou S (2011) A simple approach to emulation for computer models with qualitative and quantitative factors. Technometrics 53(3):266–273
67. Zhou Y, Hongquan X (2015) Space-filling properties of good lattice point sets. Biometrika 102(4):959–966
68. Zhu Y, Fujimura K (2010) A bayesian framework for human body pose tracking from depth image sequences. Sensors (Basel, Switzerland) 10:5280–5293