

Adaptive Kriging Method for Uncertainty Quantification of the Photoelectron Sheath and Dust Levitation on the Lunar Surface

Xinpeng Wei, first author

Department of Mechanical and Aerospace Engineering, Missouri University of Science and Technology

400 W. 13th St., Rolla, MO 65409, USA

weixinp@mst.edu

Jianxun Zhao, second author

Department of Mechanical and Aerospace Engineering, Missouri University of Science and Technology

400 W. 13th St., Rolla, MO 65409, USA

jz32k@mst.edu

Xiaoming He, third author

Department of Mathematics and Statistics, Missouri University of Science and Technology

400 W. 12th St., Rolla, MO 65409, USA

hex@mst.edu

Zhen Hu, fourth author

Department of Industrial and Manufacturing Systems Engineering, University of Michigan-Dearborn

2250 HPEC, Dearborn, MI 48128, USA

zhennhu@umich.edu

Xiaoping Du, fifth author

Department of Mechanical and Energy Engineering, Indiana University - Purdue University Indianapolis

723 W. Michigan St., Indianapolis, IN 46202, USA

dux@iu.edu

ASME Membership

Daoru Han, sixth author¹

Department of Mechanical and Aerospace Engineering, Missouri University of Science and Technology
400 W. 13th St., Rolla, MO 65409, USA
handao@mst.edu

ABSTRACT

This paper presents an adaptive Kriging based method to perform uncertainty quantification (UQ) of the photoelectron sheath and dust levitation on the lunar surface. The objective of this study is to identify the upper and lower bounds of the electric potential and that of dust levitation height, given the intervals of model parameters in the 1-D photoelectron sheath model. To improve the calculation efficiency, we employ the widely used adaptive Kriging method (AKM). A task-oriented learning function and a stopping criterion are developed to train the Kriging model and customize the AKM. Experiment analysis shows that the proposed AKM is both accurate and efficient.

¹ Corresponding author, Department of Mechanical and Aerospace Engineering, Missouri University of Science and Technology, 400 W. 13th St., Rolla, MO 65409, USA
handao@mst.edu

1. INTRODUCTION

The Moon is directly exposed to solar radiation and solar wind plasma (drifting protons and electrons) lacking an atmosphere and a global magnetic field. Consequently, the lunar surface is electrically charged by the bombardment of solar wind plasma and emission/collection of photoelectrons. Near the illuminated lunar surface, the plasma sheath is dominated by photoelectrons, thus usually referred to as “photoelectron sheath”. Additionally, dust grains on the lunar surface may get charged and levitated from the surface under the influence of the electric field within the plasma sheath as well as gravity. This work is motivated by the high computational cost associated with uncertainty quantification (UQ) analysis of plasma simulations using high-fidelity kinetic models such as particle-in-cell (PIC). The main quantities of interest (QoI) of this study is the vertical structure of the photoelectron sheath and its effects on levitation of dust grains with different sizes and electric charges.

Both the electric potential (ϕ) and the electric field (E) on lunar surface are determined by many parameters, such as solar wind drifting velocity (v_d), electron temperature (T_e), photoelectron temperature (T_p), density of ions at infinity ($n_{i,\infty}$), and density of photoelectrons (n_p), etc. Due to uncertain factors in lunar environment, the electric potential, electric field, and the dust levitation height, etc., are also uncertain. While many sources uncertainty may exist, they are generally categorized as either aleatory or epistemic. Uncertainties are characterized as epistemic, if the modeler sees a possibility to reduce them by gathering more data or by refining models. Uncertainties are categorized as aleatory if the modeler does not foresee the possibility of reducing them [1]. An example of the aleatory uncertainty in lunar environment is the solar wind parameters, and an example of the epistemic uncertainty is the photoelectron

temperature which is obtained by limited measurement data from Apollo missions. For lunar landing missions, one needs to take into consideration the uncertainties of the electrostatic and dust environment near the lunar surface. For example, the upper and lower bounds of the electric field and dust grain levitation heights in the photoelectron sheath should be considered when determining whether it is safe for a certain area to land a spacecraft.

Determining the bounds of the electric potential, electric field, and dust levitation height, however, is computationally expensive, because the particle-based kinetic models such as particle-in-cell simulations are time-consuming to evaluate. To address this issue, we develop an adaptive Kriging method (AKM) which can determine those bounds with a small number of calculations of the model. It is straightforward to train and obtain an accurate Kriging model [2] to replace the actual model and then calculate the bounds with the model. However, it is not necessary for the Kriging model to be accurate everywhere in its input space, because it will need more training samples and hence decrease the efficiency. Since the objective is to determine those bounds, we only need the Kriging model to be partially accurate near the regions of interest, as long as it can help find those bounds accurately. This way, we can save more computational efforts. To this end, we develop a task-oriented learning function and a stopping criterion to adaptively train the Kriging model. We start with an analytic model for the 1-D photoelectron sheath near the lunar surface [3, 4]. This model is computationally cheap and hence the accurate results can be obtained by brute force. With the accurate results, we can test the accuracy of the proposed method. It is noted here that the ultimate application of this method is not the simple 1-D problem presented in this work, but more complicated or computationally expensive models such as 3-D fully kinetic particle-in-cell plasma simulations.

The rest of this paper is organized as follows. Section 2 presents the 1-D photoelectron sheath and dust levitation problem on lunar surface, as well as the 1-D analytic model. Section 3 briefly introduces the Kriging method and general AKM. Section 4 presents the proposed AKM. Section 5 presents the results. Conclusions are given in Section 6.

2. PROBLEM STATEMENT

2.1. 1-D Photoelectron Sheath Model on the Lunar Surface

We employ the recently derived 1-D photoelectron sheath model for the lunar surface [3, 4]. As given in detail in [3, 4], there are three types of electric potential profiles [3-6] in the photoelectron sheath: Type A, Type B, and Type C, as shown in Fig. 1, where ϕ is the electric potential and Z is the vertical coordinate. In this study, we focus on Type C sheath profile as it is expected at the polar regions of the Moon, where the next lunar landing mission will likely occur.

Both the electrical potential ϕ and corresponding electric field E are functions of Z with a series of parameters $\mathbf{P} = (v_d, T_e, T_p, n_{i,\infty}, n_p)$. To obtain $\phi(Z; \mathbf{P})$ and $E(Z; \mathbf{P})$, we need to solve an ordinary differential equation (ODE) [3]. Once the potential profile ϕ is obtained, it is straightforward to calculate electric field E by

$$E(Z; \mathbf{P}) = -\frac{d\phi(Z; \mathbf{P})}{dZ} \quad (1)$$

A typical Type C sample curve of $E(Z; \mathbf{P})$ is shown in Fig. 2. Note that both ϕ and E converge to zero at large values of Z where it is used as the electric potential reference (zero potential and zero field).

2.2. Dust Levitation

Subjected to the electric field force, a charged dust on lunar surface may be levitated [7, 8]. Above the lunar surface, there is a position where the upward electric field force balances the downward gravity [4]. This position is referred to as equilibrium levitation height, denoted as Z^* . Z^* can be solved through the following equation of static equilibrium of a charged dust in an electric field:

$$qE(Z; \mathbf{P}) = mg \quad (2)$$

where q is the dust charge, m is the mass of the dust, and $g = 1.62 \text{ m/s}^2$ is the gravity acceleration on lunar surface [9]. With the assumption of spherical dust grains, m is given by

$$m = \frac{4}{3} \pi r^3 \rho \quad (3)$$

where r is the radius of the lunar dust grain, and $\rho = 1.8 \text{ g/cm}^3$ is the mass density of dust grains [10].

For simplicity, Eq. (2) is rewritten as

$$E(Z; \mathbf{P}) = w \quad (4)$$

where $w = mg/q$. Once both $E(Z; \mathbf{P})$ and w have been given or determined, a root-finding scheme is employed to solve Eq. (4) for Z^* . Fig. 3 shows an example of how to obtain Z^* graphically.

2.3. Objective

Due to the lack of information, it is almost impossible to obtain the distribution functions of \mathbf{P} . The bounds of \mathbf{P} , however, are much easier to obtain. In some work designs on

lunar surface, we need to determine the bounds of $\phi(Z; \mathbf{P})$ and/or $E(Z; \mathbf{P})$, given the bounds of \mathbf{P} . In this study, all the parameters in \mathbf{P} are modeled as interval variables, whose domain is denoted as Ω . For a given realization \mathbf{p} of \mathbf{P} , both $\phi(Z; \mathbf{p})$ and $E(Z; \mathbf{p})$, $Z \in [Z_{\min}, Z_{\max}]$ are obtained by solving the ODE.

The upper bound $\overline{\phi}(Z)$ of the electric potential is defined as

$$\overline{\phi}(z) = \max_{\mathbf{p} \in \Omega} \phi(z; \mathbf{p}) \quad (5)$$

where z is a given value of variable Z . Note that the entire upper bound curve $\overline{\phi}(Z)$ is not necessarily determined by a specific \mathbf{p} . In other words, at different values of z , $\overline{\phi}(z)$ may be determined by different realizations of \mathbf{P} . Similarly, the lower bound $\underline{\phi}(Z)$ of the electric potential, the upper bound $\overline{E}(Z)$ of the electric field, and the lower bound $\underline{E}(Z)$ are defined as

$$\underline{\phi}(z) = \min_{\mathbf{p} \in \Omega} \phi(z; \mathbf{p}) \quad (6)$$

$$\overline{E}(z) = \max_{\mathbf{p} \in \Omega} E(z; \mathbf{p}) \quad (7)$$

$$\underline{E}(z) = \min_{\mathbf{p} \in \Omega} E(z; \mathbf{p}) \quad (8)$$

Since \mathbf{P} are modeled as interval variables and the intervals (lower and upper bounds) of output are desired, we in fact cope with interval propagation problems in this work. The most straightforward method to determine $\overline{\phi}(Z), \underline{\phi}(Z), \overline{E}(Z)$ and $\underline{E}(Z)$ is through Monte Carlo Simulation (MCS) [11] in the following steps. First, evenly generate a large number N_{MCS} of samples of \mathbf{P} . For convenience, we denote those samples as \mathbf{p}^{MCS} . Second, obtain corresponding N_{MCS} samples of $\phi(Z; \mathbf{P})$ and $E(Z; \mathbf{P})$ by solving the ODE N_{MCS} times. Finally, calculate $\overline{\phi}(Z), \underline{\phi}(Z), \overline{E}(Z)$ and $\underline{E}(Z)$ using the N_{MCS} samples of $\phi(Z; \mathbf{P})$ and $E(Z; \mathbf{P})$:

$$\overline{\phi}(z) = \max_{\mathbf{p} \in \mathbf{p}^{\text{MCS}}} \phi(z; \mathbf{p}) \quad (9)$$

$$\underline{\phi}(z) = \min_{\mathbf{p} \in \mathbf{p}^{\text{MCS}}} \phi(z; \mathbf{p}) \quad (10)$$

$$\overline{E}(z) = \max_{\mathbf{p} \in \mathbf{p}^{\text{MCS}}} E(z; \mathbf{p}) \quad (11)$$

$$\underline{E}(z) = \min_{\mathbf{p} \in \mathbf{p}^{\text{MCS}}} E(z; \mathbf{p}) \quad (12)$$

However, this method is too expensive or even unaffordable. One reason is that solving the ODE a large number N_{MCS} of times is time-consuming, even when the analytic solution to the ODE is available for the 1-D problem. Another reason is that there is no analytic solution to complex 2-D or 3-D problems where kinetic particle-in-cell simulations are usually employed to solve the electrostatic field through Poisson's equation.

The objective of this study is to develop a method to determine $\overline{\phi}(Z)$, $\underline{\phi}(Z)$, $\overline{E}(Z)$ and $\underline{E}(Z)$ accurately and then calculate Z^* of dust grains. It is noted here that the ultimate application of this method is not the relatively simple 1-D problem presented in this work, but more complicated or computationally expensive models such as 3-D fully kinetic particle-in-cell plasma simulations. For computationally expensive models, evaluating the model consumes the majority of computational resource, so we will use the number N_{ODE} of ODEs that we need to solve as a measure of the computational cost.

3. INTRODUCTION TO KRIGING MODEL AND AKM

Before presenting the proposed method, we briefly introduce the Kriging model [12, 13] and AKM [13-28], on which the proposed method is based.

3.1. Overview of Kriging Model

Kriging model makes regression to a black-box function (BBF) using a training sample set, or a design of experiment (DoE). The main idea of Kriging is to treat the BBF as a realization of a Gaussian random field indexed by the input variables of the BBF. The theoretical foundation of Kriging model is exactly the Bayesian inference [29]. From the perspective of Bayesian interface, a prior Gaussian random field is trained by the DoE and hence a posterior Gaussian random field is generated. Then the mean value function, also indexed by the input variables of the BBF, of the posterior random field is the Kriging prediction to the BBF. Besides, the variance function, also indexed by the input variables of the BBF, of the posterior random field quantifies the local prediction uncertainty or prediction error.

The randomness, or uncertainty, of the posterior random field mainly comes from the fact that only a limit number of samples of the BBF are used to train the prior random field. In other words, only part of the information of the BBF is available, and the missing part of information leads to the epistemic uncertainty in the random field. Generally, the more training samples we use, the less epistemic uncertainty will result and with stronger confidence will we predict the BBF.

3.2. Formulation for Kriging Model

A simple yet widely used prior random field is the stationary Gaussian random field given by

$$K(\mathbf{X}) = \mu + \eta(\mathbf{X}; \xi^2, \boldsymbol{\theta}) \quad (13)$$

where μ is an unknown parameter representing the mean value of the random field $K(\mathbf{X})$ and $\eta(\mathbf{X}; \xi^2, \theta)$ is a zero-mean stationary Gaussian random field indexed by \mathbf{X} , the input variables of a BBF $k(\mathbf{X})$. Both the variance parameter ξ^2 and correlation parameters θ of $\eta(\mathbf{X}; \xi^2, \theta)$ are unknown. The parameters μ , ξ^2 and θ fully define the prior random field $K(\mathbf{X})$. A DoE, or a training sample set, of $k(\mathbf{X})$ is used to train $K(\mathbf{X})$ and then determine those parameters.

The correlation function $C(\mathbf{x}^{(i)}, \mathbf{x}^{(j)})$ of $\eta(\mathbf{X}; \xi^2, \theta)$ is given by

$$C(\mathbf{x}^{(i)}, \mathbf{x}^{(j)}) = \xi^2 R(\mathbf{x}^{(i)}, \mathbf{x}^{(j)}; \theta) \quad (14)$$

where $R(\mathbf{x}^{(i)}, \mathbf{x}^{(j)}; \theta)$ is the correlation coefficient function of $\eta(\mathbf{X}; \xi^2, \theta)$ at two points $\mathbf{x}^{(i)}$ and $\mathbf{x}^{(j)}$ of \mathbf{X} . There are many models for $R(\mathbf{x}^{(i)}, \mathbf{x}^{(j)}; \theta)$. A widely used model is known as the Gaussian model, or squared exponential model, given by

$$R(\mathbf{x}^{(i)}, \mathbf{x}^{(j)}; \theta) = \prod_{d=1}^D \exp \left[-\theta_d \left(x_d^{(i)} - x_d^{(j)} \right)^2 \right] \quad (15)$$

where D is the dimension of \mathbf{X} , $x_d^{(i)}$ is the d^{th} component of $\mathbf{x}^{(i)}$, and θ_d is the d^{th} component of θ .

For a BBF $k(\mathbf{X})$, the Kriging model predicts $k(\mathbf{x})$ as $\tilde{k}(\mathbf{x})$, which is a normal variable whose mean value and variance are $\hat{k}(\mathbf{x})$ and $\sigma^2(\mathbf{x})$, respectively. Note that $\sigma^2(\mathbf{x})$ is also termed as mean squared error (MSE). Generally, $\hat{k}(\mathbf{x})$ is regarded as the deterministic prediction to $k(\mathbf{x})$, since a deterministic prediction is usually needed. $\sigma^2(\mathbf{x})$ measures the prediction uncertainty, or prediction error, and therefore we can validate a Kriging model simply using $\hat{k}(\mathbf{x})$ and $\sigma^2(\mathbf{x})$ without employing traditional validation methods, such as the cross validation [30]. Because of this advantage, many algorithms have been proposed to adaptively train a Kriging model for expensive BBFs [14-27, 31-36]. When sufficient training

samples have been used for training, $\sigma^2(\mathbf{x})$ converges to 0 and the normal variable $\tilde{k}(\mathbf{x})$ degenerates to a deterministic value, i.e., the exact value of $k(\mathbf{x})$.

3.3. An Example of Kriging Model

Fig. 4 shows a 1-D example of Kriging model. In total five initial training samples are used to train the Kriging. The vertical distance between $\hat{k}(x) \pm \sigma(x)$ graphically quantify the prediction error at x . The larger the distance, the larger the prediction error. On interval $[0, 2]$, the training samples are denser than that on $[2, 10]$. Consequently, the prediction error is smaller on $[0, 2]$ than that on $[2, 10]$. It is noted that the prediction error is not only dependent on the density of the training samples but also the nonlinearity of the BBF. With the prediction error shown in Fig. 4, it is obvious that in order to improve the prediction accuracy, we need to add training samples somewhere near $x = 4$ and $x = 8$. Fig. 5 shows the updated Kriging model with one more training sample added at $x = 8$. The overall prediction accuracy is improved significantly.

3.4. AKM

The main idea of AKM is to adaptively add training samples to update the Kriging model iteratively until an expected accuracy is achieved. Fig. 6 shows a brief flowchart of AKM. The QoI is what we aim to calculate, such as $\overline{\phi}(Z)$ and $\underline{\phi}(Z)$. Since the QoI is calculated through the Kriging model instead of the BBF itself, there is an inevitable error caused by the Kriging model. The error metric is used to measure the error. The stopping criterion, which is based on the

error metric, is used to determine when to stop adding training samples. Once the error of QoI is sufficiently small, it is reasonable to return the QoI and stop the algorithm. If the error is large in an iteration, we must add one or more training samples to update the Kriging model. How to determine new training samples is the task of the learning function. A good learning function should be robust and lead to a high convergence rate.

Given a specific engineering problem, the key of employing an AKM is to make good use of all available information, such as the features of the BBF and QoI, and then design a customized or task-oriented error metric, stopping criterion and learning function.

In the UQ community, a great number of AKMs have been developed to solve various kinds of problems, such as reliability analysis [15, 17-24, 26, 31-33, 36], robustness analysis [14], sensitivity analysis [34], robust design [25, 35], and reliability-based design [16, 27], etc.

4. THE PROPOSED METHOD

In this section, we present detailed procedures of calculating $\overline{\phi}(Z)$ and $\underline{\phi}(Z)$. Similar procedures can also apply to calculate $\overline{E}(Z)$ and $\underline{E}(Z)$.

4.1. Overview of the Proposed Method

The main idea of the proposed method is to employ the framework of AKM and customize it to calculate $\overline{\phi}(Z)$ and $\underline{\phi}(Z)$ (as well as $\overline{E}(Z)$ and $\underline{E}(Z)$). Fig. 7 shows the brief flowchart of the proposed method. In Step 1, we evenly generate N_{in} initial samples of \mathbf{P} . Generally, N_{in} is much smaller than N_{MCS} . Details of this step will be given in Subsection 4.2. In

Step 2, the ODE (1-D Poisson's equation) is solved N_{in} times, with the N_{in} samples of \mathbf{P} , to obtain N_{in} samples of $\phi(Z; \mathbf{P})$. In Step 3, the samples of $\phi(Z; \mathbf{P})$ are used to build a Kriging model $\hat{\phi}(Z; \mathbf{P})$. Both Z and \mathbf{P} are treated as input variables so the dimension of $\hat{\phi}(Z; \mathbf{P})$ is $1 + 5 = 6$. In Step 4, $\bar{\phi}(Z)$ and $\underline{\phi}(Z)$ are estimated through

$$\bar{\phi}(z) = \max_{\mathbf{p} \in \mathbf{p}^{\text{MCS}}} \hat{\phi}(z; \mathbf{p}) \quad (16)$$

$$\underline{\phi}(z) = \min_{\mathbf{p} \in \mathbf{p}^{\text{MCS}}} \hat{\phi}(z; \mathbf{p}) \quad (17)$$

In Step 5, an error metric is developed to measure the error of $\bar{\phi}(Z)$ and $\underline{\phi}(Z)$ estimated by Eq. (16) and Eq. (17). Step 6 is about a stopping criterion. Details about Steps 5 and 6 will be given in Subsection 4.4. The learning function involved in Step 7 will be given in Subsection 4.3. The implementation of the proposed method will be given in Subsection 4.5.

There are two significant differences between most existing AKMs and the proposed method. First, the former aims at estimating a constant value, such as the structural reliability and robustness, while the latter aims at estimating two functions, i.e., $\bar{\phi}(Z)$ and $\underline{\phi}(Z)$. Second, when given a specific value of input, the output of the BBFs involved in the former methods is a single value. However, in this work, with a given realization \mathbf{p} of \mathbf{P} , the output of solving the ODE is a function $\phi(Z; \mathbf{p})$. With those differences, we cannot use the existing error metrics, stopping criteria or learning functions. Instead, we take into consideration those differences and design a new error metric, stopping criterion and learning function to fit the problem. This is the main contribution of the proposed algorithm.

4.2. Candidate Samples and Initial Training Samples

For numerical computation, we need to evenly discretize Ω into a few points. Suppose P_i , the i^{th} component of \mathbf{P} , is discretized into N_i points, then Ω will be discretized into in total $N_P = \prod_{i=1}^5 N_i$ points. For convenience, we denote the set of those points as \mathbf{p}^{MCS} . Similarly, Z is discretized into N_Z points (denoted as \mathbf{z}^{MCS}) in its range $[Z_{\min}, Z_{\max}]$. Theoretically, any $\mathbf{p} \in \Omega$ could be selected as a training sample for $\hat{\phi}(Z; \mathbf{P})$. However, we do not want any two training samples to be clustering together, because we use the exact interpolation in Kriging and clustered training samples may impact the training and the convergence rate of the proposed AKM. Therefore, we only select training samples of \mathbf{P} from \mathbf{p}^{MCS} and call \mathbf{p}^{MCS} candidate samples or candidate points.

The N_{in} initial training samples \mathbf{p}^{in} of \mathbf{P} are selected such that they are distributed in Ω as even as possible. Commonly used sampling methods include random sampling, Latin hypercube sampling and Hammersley sampling [37]. In this study, we employ the Hammersley sampling method because it has better uniformity properties over a multidimensional space [38]. The Hammersley sampling method firstly generates initial training samples in a 5-dimensional hypercube $[0,1]^5$ and then they are mapped into Ω to get the initial training samples of \mathbf{P} . Note that the five dimensions of the hypercube are assumed to be independent, with the assumption that all variables in \mathbf{P} are independent. Those initial training samples, however, are not necessarily among \mathbf{p}^{MCS} , so we need to round them to the nearest ones in \mathbf{p}^{MCS} . Since the components of \mathbf{P} do not necessarily share the same dimension unit, the distances which we use to find the nearest samples should be normalized. For example, the

distance d between a sample $\mathbf{p}^{(h)}$ generated by Hammersley and a candidate sample $\mathbf{p}^{(c)}$ in \mathbf{p}^{MCS} is given by

$$d(\mathbf{p}^{(h)}, \mathbf{p}^{(c)}) = \sqrt{\sum_{i=1}^5 \left(\frac{p_i^{(h)} - p_i^{(c)}}{P_{i,\max} - P_{i,\min}} \right)^2} \quad (18)$$

where $p_i^{(h)}$ is the i^{th} component of $\mathbf{p}^{(h)}$, $p_i^{(c)}$ is the i^{th} component of $\mathbf{p}^{(c)}$, $P_{i,\max}$ is the maximal value of P_i which is the i^{th} component of \mathbf{P} , and $P_{i,\min}$ is the minimal value of P_i . Then $\mathbf{p}^{(h)}$ is rounded to $\mathbf{p}^* = \arg \min_{\mathbf{p} \in \mathbf{p}^{\text{MCS}}} d(\mathbf{p}^{(h)}, \mathbf{p}^{(c)})$. When all the initial training samples generated by Hammersley have been rounded to the nearest ones in \mathbf{p}^{MCS} , we get the initial training sample set $\mathbf{p}^{\text{in}} \subset \mathbf{p}^{\text{MCS}}$ of \mathbf{P} .

Solving the ODE N_{in} times, each with a sample in \mathbf{p}^{in} , we get N_{in} samples of $\phi(Z; \mathbf{P})$. Note that each sample of $\phi(Z; \mathbf{P})$ has N_Z points, since we discretized Z into N_Z points. Then we have $N_Z N_{\text{in}}$ input training points $\mathbf{z}^{\text{MCS}} \times \mathbf{p}^{\text{in}}$. Except the N_{in} points at Z_{\max} , we select the other $(N_Z - 1)N_{\text{in}}$ points to form the first part of input training sample set of $\hat{\phi}(Z; \mathbf{P})$. We denote those $(N_Z - 1)N_{\text{in}}$ input training points as \mathbf{x}^{inp1} , where superscript *inp* of \mathbf{x} represents input, and the superscript *1* means that \mathbf{x}^{inp1} is only the first part of the entire input training sample set. The other part \mathbf{x}^{inp2} is given below.

Since for any $\mathbf{p} \in \mathbf{p}^{\text{MCS}}$, it is known that $\phi(Z_{\max}; \mathbf{p}) \equiv 0$ (Fig. 1), theoretically we also need to add all the N_P points $Z_{\max} \times \mathbf{p}^{\text{MCS}}$ as input training samples so that we make good use of all known information. However, it is not practical to do so. For example, if $N_i = 10, i = 1, 2, \dots, 5$, we need to add $N_P = 10^5$ points as input training samples. So many training samples will make $\hat{\phi}(Z; \mathbf{P})$ complex, expensive and not accurate, losing its expected properties. To

balance the need to add them and the drawback of adding all of them, we add part of them.

Specifically, we evenly generate N'_p samples \mathbf{p}' of \mathbf{P} using procedures similar to what we used to generate \mathbf{p}^{in} . Then \mathbf{x}^{inp2} is given by

$$\mathbf{x}^{\text{inp2}} = \{(Z_{\text{max}}, \mathbf{p}) | \mathbf{p} \in \mathbf{p}'\} \quad (19)$$

The input training sample set $\mathbf{x}^{\text{inp}} = \mathbf{x}^{\text{inp1}} \cup \mathbf{x}^{\text{inp2}}$. Denote the corresponding electric potential ϕ at \mathbf{x}^{inp} as ϕ^{out} . The input-output training sample pairs $(\mathbf{x}^{\text{inp}}, \phi^{\text{out}})$ are used to build the initial $\hat{\phi}(Z; \mathbf{P})$. More training samples will be added to update $\hat{\phi}(Z; \mathbf{P})$ later.

4.3. Learning Function

Generally, the initial Kriging model is not accurate enough to get $\bar{\phi}(Z)$ or $\underline{\phi}(Z)$ accurately through Eq. (5) and Eq. (6). To improve the accuracy of $\hat{\phi}(Z; \mathbf{P})$ and hence of $\bar{\phi}(Z)$ and $\underline{\phi}(Z)$, we need to add training samples of $\phi(Z; \mathbf{P})$ to refine $\hat{\phi}(Z; \mathbf{P})$. A learning function is used to determine which sample of \mathbf{P} , and hence of $\phi(Z; \mathbf{P})$, should be added.

In our previous work [3], we used the learning function given by

$$\mathbf{p}^{(\text{next})} = \arg \max_{\mathbf{p} \in \mathbf{p}^{\text{MCS}}} \sum_{z \in \mathbf{z}^{\text{MCS}}} \left| \frac{\sigma(z; \mathbf{p})}{\hat{\phi}(z; \mathbf{p})} \right| \quad (20)$$

where $\mathbf{p}^{(\text{next})}$ is the next to-be-added sample of \mathbf{P} , $\hat{\phi}(z; \mathbf{p})$ is the predicted value of $\phi(z; \mathbf{p})$ by the Kriging model $\hat{\phi}(Z; \mathbf{P})$, and $\sigma(z; \mathbf{p})$ is the standard deviation of the prediction. Both $\hat{\phi}(z; \mathbf{p})$ and $\sigma(z; \mathbf{p})$ are calculated by the Kriging toolbox. $\frac{\sigma(z^{(j)}; \mathbf{p})}{\hat{\phi}(z^{(j)}; \mathbf{p})}$ is the deviation coefficient of the prediction at $(z; \mathbf{p})$, and thus the learning function in Eq. (20) determines the training sample

$\mathbf{p}^{(\text{next})}$ at which the summation of the absolute deviation coefficients of the predictions along Z coordinate is maximal. The summation $\sum_{z \in Z^{\text{MCS}}} \left| \frac{\sigma(z; \mathbf{p})}{\hat{\phi}(z; \mathbf{p})} \right|$ measures the overall prediction error at \mathbf{p} . Adding a sample of $\phi(Z; \mathbf{P})$ at \mathbf{p} to update $\hat{\phi}(Z; \mathbf{P})$ will let $\sum_{z \in Z^{\text{MCS}}} \left| \frac{\sigma(z; \mathbf{p})}{\hat{\phi}(z; \mathbf{p})} \right|$ become zero, and therefore adding a sample of $\phi(Z; \mathbf{P})$ at $\mathbf{p}^{(\text{next})}$ to update $\hat{\phi}(Z; \mathbf{P})$ will decrease the overall prediction error of $\phi(Z; \mathbf{P})$ by the largest extent. This is the basic mechanism of the learning function in Eq. (20).

However, we do not necessarily need $\hat{\phi}(Z; \mathbf{P})$ to be overall accurate. Since the objective is to estimate $\bar{\phi}(Z)$ and $\underline{\phi}(Z)$ accurately and efficiently, we only need $\hat{\phi}(Z; \mathbf{P})$ to be partially or locally accurate enough so that it can help estimate $\bar{\phi}(Z)$ and $\underline{\phi}(Z)$ accurately. With this idea, we can further improve the efficiency of updating $\hat{\phi}(Z; \mathbf{P})$ by adding training samples more skillfully.

A widely used learning function in an AKM which aims at calculating extreme values is the expected improvement function [28]. The expected improvement function $\bar{\xi}(z, \mathbf{p})$ of $\bar{\phi}(z)$ is given by

$$\begin{aligned} \bar{\xi}(z, \mathbf{p}) = & \left(\hat{\phi}(z; \mathbf{p}) - \bar{\phi}(z) \right) \Phi \left(\frac{\hat{\phi}(z; \mathbf{p}) - \bar{\phi}(z)}{\sigma(z; \mathbf{p})} \right) \\ & + \sigma(z; \mathbf{p}) \varphi \left(\frac{\hat{\phi}(z; \mathbf{p}) - \bar{\phi}(z)}{\sigma(z; \mathbf{p})} \right) \end{aligned} \quad (21)$$

where $\Phi(\cdot)$ and $\varphi(\cdot)$ are the cumulative distribution function and probability density function of the standard Gaussian variable, respectively. A simple explanation to the expected improvement function $\bar{\xi}(z, \mathbf{p})$ is that if we added a training point at (z, \mathbf{p}) , we could expect to improve current $\bar{\phi}(z)$ to $\bar{\phi}(z) + \bar{\xi}(z, \mathbf{p})$, with an improvement rate of $\bar{\xi}(z, \mathbf{p})/\bar{\phi}(z)$. If the

objective is to estimate $\bar{\phi}(z)$, which is a maximal value, instead of $\bar{\phi}(Z)$, which is an entire function, we can determine the next training sample $\mathbf{p}^{(\text{next})}$ of \mathbf{P} using the learning function given by

$$\mathbf{p}^{(\text{next})} = \arg \max_{\mathbf{p} \in \mathbf{p}^{\text{MCS}}} |\bar{\xi}(z, \mathbf{p}) / \bar{\phi}(z)| \quad (22)$$

However, since the objective is to determine the entire function $\bar{\phi}(Z)$ and one ODE solution has N_Z training points, we must have a learning function which aims at improving the calculation accuracy of the entire function $\bar{\phi}(Z)$. Therefore, we propose a learning function given by

$$\mathbf{p}^{(\text{next})} = \arg \max_{\mathbf{p} \in \mathbf{p}^{\text{MCS}}} \sum_{z \in \mathbf{z}^{\text{MCS}}} |\bar{\xi}(z, \mathbf{p}) / \bar{\phi}(z)| \quad (23)$$

where we sum up the absolute values of the improvement rate. This learning function means that if we added a training sample $\phi(Z; \mathbf{p}^{(\text{next})})$, which has N_Z points, to update $\hat{\phi}(Z; \mathbf{P})$, we could expect to get the best improvement of $\bar{\phi}(Z)$.

Similarly, the expected improvement function $\underline{\xi}(z, \mathbf{p})$ of $\underline{\phi}(z)$ is given by

$$\begin{aligned} \underline{\xi}(z, \mathbf{p}) = & \left(\underline{\phi}(z) - \hat{\phi}(z; \mathbf{p}) \right) \Phi \left(\frac{\underline{\phi}(z) - \hat{\phi}(z; \mathbf{p})}{\sigma(z; \mathbf{p})} \right) \\ & + \sigma(z; \mathbf{p}) \varphi \left(\frac{\underline{\phi}(z) - \hat{\phi}(z; \mathbf{p})}{\sigma(z; \mathbf{p})} \right) \end{aligned} \quad (24)$$

To estimate $\underline{\phi}(Z)$, we also propose a learning function given by

$$\mathbf{p}^{(\text{next})} = \arg \max_{\mathbf{p} \in \mathbf{p}^{\text{MCS}}} \sum_{z \in \mathbf{z}^{\text{MCS}}} |\underline{\xi}(z, \mathbf{p}) / \underline{\phi}(z)| \quad (25)$$

In order to estimate both $\bar{\phi}(Z)$ and $\underline{\phi}(Z)$ simultaneously, we combine Eq. (23) and Eq. (25) to propose a learning function given by

$$\mathbf{p}^{(\text{next})} = \arg \max \left\{ \max_{\mathbf{p} \in \mathbf{p}^{\text{MCS}}} \sum_{z \in \mathbf{z}^{\text{MCS}}} \left| \frac{\bar{\xi}(z, \mathbf{p})}{\bar{\phi}(z)} \right|, \max_{\mathbf{p} \in \mathbf{p}^{\text{MCS}}} \sum_{z \in \mathbf{z}^{\text{MCS}}} \left| \frac{\underline{\xi}(z, \mathbf{p})}{\underline{\phi}(z)} \right| \right\} \quad (26)$$

Once $\mathbf{p}^{(\text{next})}$ has been determined, we solve the ODE to numerically get a function

$\phi(Z; \mathbf{p}^{(\text{next})})$, from which we get $(N_z - 1)$ points (the remaining one at Z_{max} , where $\phi \equiv 0$, is excluded) and add them into $(\mathbf{x}^{\text{inp}}, \boldsymbol{\phi}^{\text{out}})$ to enrich the training samples.

4.4. Error Metric and Stopping Criterion

Since Eq. (16) and Eq. (17) cannot obtain absolutely accurate $\bar{\phi}(Z)$ and $\underline{\phi}(Z)$ due to the prediction error of $\hat{\phi}(Z; \mathbf{P})$, we need an error metric to measure the error of currently

estimated $\bar{\phi}(Z)$ and $\underline{\phi}(Z)$. Since $\left| \frac{\bar{\xi}(z, \mathbf{p})}{\bar{\phi}(z)} \right|$ measures the absolute expected improvement rate of

$\bar{\phi}(z)$, if $\left| \frac{\bar{\xi}(z, \mathbf{p})}{\bar{\phi}(z)} \right|$ is small for any $z \in \mathbf{z}^{\text{MCS}}$ and $\mathbf{p} \in \mathbf{p}^{\text{MCS}}$, $\bar{\phi}(Z)$ is expected to sufficiently

accurate. Therefore, we propose to use $\max_{z \in \mathbf{z}^{\text{MCS}}, \mathbf{p} \in \mathbf{p}^{\text{MCS}}} \left| \frac{\bar{\xi}(z, \mathbf{p})}{\bar{\phi}(z)} \right|$ to quantify the error of $\bar{\phi}(Z)$.

Similarly, $\max_{z \in \mathbf{z}^{\text{MCS}}, \mathbf{p} \in \mathbf{p}^{\text{MCS}}} \left| \frac{\underline{\xi}(z, \mathbf{p})}{\underline{\phi}(z)} \right|$ is used to quantify the error of $\underline{\phi}(Z)$. Combining both, we have

the error metric Γ , which measures the error of both $\bar{\phi}(Z)$ and $\underline{\phi}(Z)$, given by

$$\Gamma = \max \left\{ \max_{z \in \mathbf{z}^{\text{MCS}}, \mathbf{p} \in \mathbf{p}^{\text{MCS}}} \left| \frac{\bar{\xi}(z, \mathbf{p})}{\bar{\phi}(z)} \right|, \max_{z \in \mathbf{z}^{\text{MCS}}, \mathbf{p} \in \mathbf{p}^{\text{MCS}}} \left| \frac{\underline{\xi}(z, \mathbf{p})}{\underline{\phi}(z)} \right| \right\} \quad (27)$$

Once Γ is small enough, the estimated $\bar{\phi}(Z)$ and $\underline{\phi}(Z)$ are expected to be sufficiently accurate. Therefore, the stopping criterion shown in Fig. 7 is defined as

$$\Gamma < \gamma \quad (28)$$

where γ is a threshold that controls the efficiency and accuracy of the proposed AKM. Generally speaking, a smaller γ will lead to higher accuracy but lower efficiency.

4.5. Implementation

As shown in Fig. 1, $\phi(Z; \mathbf{P})$ approaches zero when Z takes large value. As a result, $\bar{\phi}(z)$ and $\underline{\phi}(z)$ in Eq. (26) and Eq. (27) are likely to take very small values close to zero. It leads to the singularity of the calculation of Eq. (26) and Eq. (27), doing harm to the robustness of the proposed algorithm. To solve this issue, we translate all training samples of $\phi(Z; \mathbf{P})$ simply by adding a negative constant ϵ . This way, the translated $\phi(Z; \mathbf{P})$ will never approach zero and the singularity issue is evitable. Trained by the translated samples of $\phi(Z; \mathbf{P})$, the Kriging model $\hat{\phi}(Z; \mathbf{P})$ will also lead to the translation of $\bar{\phi}(Z)$ and $\underline{\phi}(Z)$. We can translate $\bar{\phi}(Z)$ and $\underline{\phi}(Z)$ back simply by subtracting ϵ from them. Note that there is no rigorous theory to quantify how ϵ affect the properties of the proposed AKM. We suggest determining ϵ using

$$\epsilon = \text{mean}\{\phi(0; \mathbf{p}) | \mathbf{p} \in \mathbf{p}^{\text{in}}\} \quad (29)$$

where $\text{mean}(\cdot)$ represents mean value.

Based on all the procedures given above, we generate the pseudo codes of the proposed AKM given in Algorithm 1.

4.6. Validation discussion

Theoretically, it is vital to validate the Kriging model to make sure that it has been trained accurately. An explicit validation, however, is not involved in the proposed AKM. There are two main reasons. First, the adaptive training focuses on the accuracy of QoI instead of the accuracy of the Kriging model. Once there is an indication that the accuracy of QoI in current training iteration is sufficient, i.e., the stopping criterion in Eq. (27) is satisfied, the algorithm stops adding more training samples, no matter the Kriging model is globally accurate or not. As a result, when the algorithm has converged, it is very likely that the Kriging model is accurate only on some subdomains but not accurate on other domains. Therefore, it is not suitable to do explicit cross validation. Second, the error metric I' can measure the accuracy of QoI, and therefore we in fact do validation implicitly. As long as the accuracy of QoI is sufficient, it does not matter if the Kriging model is or not accurate on the entire domain.

5. RESULTS

In this section, we illustrate the proposed AKM. MCS is used to solve the same problems with brute force. Results by MCS are treated as standard to verify the proposed AKM. We build the Kriging model and calculate the Kriging predictions using the DACE toolbox [39]. The anisotropic Gaussian kernel is used.

5.1. Sheath Profile

We consider the 1-D photoelectron sheath problem discussed in Section 2. The sun elevation angle is given as 9 degrees. The maximal and minimal values of $\mathbf{P} = (v_d, T_e, T_p, n_{i,\infty}, n_p)$ are given in Table 1. We use both MCS and the proposed AKM to estimate $\overline{\phi}(Z)$ and $\underline{\phi}(Z)$. The values of all involved parameters are given in Table 2.

The domain Ω of \mathbf{P} is discretized into $N_p = 5^5$ points, which are assembled into \mathbf{p}^{MCS} . The $N_{\text{in}} = 5$ samples in hypercube space $[0,1]^5$, generated by Hammersley sampling method, are given in Table 3. Then the 5 samples are mapped into Ω , as given in Table 4. Rounding the 5 samples in Ω to the nearest ones in \mathbf{p}^{MCS} , we get the initial samples \mathbf{p}^{in} of \mathbf{P} , as given in Table 5. Solving the ODE five times, each with a sample in \mathbf{p}^{in} , we get five samples of $\phi(Z; \mathbf{P})$ as shown in Fig. 8.

Each sample of $\phi(Z; \mathbf{P})$ contains $N_Z = 50$ numerical points. Excluding the five points at Z_{max} , we have $N_Z N_{\text{in}} - 5 = 245$ training points in $(\mathbf{x}^{\text{inp1}}, \boldsymbol{\phi}^{\text{out1}})$. With Hammersley sampling method, we generate $N'_p = 100$ samples of \mathbf{P} and hence 100 training points in $(\mathbf{x}^{\text{inp2}}, \boldsymbol{\phi}^{\text{out2}})$. Note that all points in $(\mathbf{x}^{\text{inp2}}, \boldsymbol{\phi}^{\text{out2}})$ have $Z = Z_{\text{max}}$ and $\phi = 0$. Combining $(\mathbf{x}^{\text{inp1}}, \boldsymbol{\phi}^{\text{out1}})$ and $(\mathbf{x}^{\text{inp2}}, \boldsymbol{\phi}^{\text{out2}})$, we have 345 training points in $(\mathbf{x}^{\text{inp}}, \boldsymbol{\phi}^{\text{out}})$. To do the translation mentioned in Subsection 4.5, we update $\boldsymbol{\phi}^{\text{out}}$ simply by $\boldsymbol{\phi}^{\text{out}} = \boldsymbol{\phi}^{\text{out}} + \epsilon$, where $\epsilon = -6.97$ V is obtained with Eq. (29). With the updated $(\mathbf{x}^{\text{inp}}, \boldsymbol{\phi}^{\text{out}})$, we build an initial Kriging model and then estimate $\overline{\phi}(Z)$ and $\underline{\phi}(Z)$ through Eq. (16) and Eq. (17). Finally, we translate $\overline{\phi}(Z)$ and $\underline{\phi}(Z)$ back by $\overline{\phi}(Z) = \overline{\phi}(Z) - \epsilon$ and $\underline{\phi}(Z) = \underline{\phi}(Z) - \epsilon$. Fig. 9 shows the $\overline{\phi}(Z)$ and $\underline{\phi}(Z)$ estimated by both MCS and the proposed AKM (with the initial Kriging model). It shows that the initial Kriging model is not able to predict $\overline{\phi}(Z)$ or $\underline{\phi}(Z)$ with sufficient accuracy.

To improve the accuracy, the proposed method indicates adding a sample at $\mathbf{p}^{(\text{next})} = (514800, 13.2, 2.2, 9.57, 57.6)$. With the $\mathbf{p}^{(\text{next})}$, we solve the ODE and get a new sample of $\phi(Z; \mathbf{P})$. This sample contains $N_Z = 50$ numerical points. We translate all the numerical points and add them, excluding the one at Z_{max} , to update $(\mathbf{x}^{\text{inp}}, \boldsymbol{\phi}^{\text{out}})$. The reason why we abandon the point at Z_{max} is that there are already enough points at Z_{max} in $(\mathbf{x}^{\text{inp2}}, \boldsymbol{\phi}^{\text{out2}})$. With the updated $(\mathbf{x}^{\text{inp}}, \boldsymbol{\phi}^{\text{out}})$, we build a new $\hat{\phi}(Z; \mathbf{P})$. With the new $\hat{\phi}(Z; \mathbf{P})$ another $\mathbf{p}^{(\text{next})}$ is indicated. With similar procedures, more and more samples of $\phi(Z; \mathbf{P})$ are added to refine $\hat{\phi}(Z; \mathbf{P})$ until the stopping criterion given in Eq. (28) is satisfied.

The final estimation of $\bar{\phi}(Z)$ and $\underline{\phi}(Z)$ is shown in Fig. 10. It shows that the proposed AKM can estimate $\bar{\phi}(Z)$ and $\underline{\phi}(Z)$ very accurately. 16 more samples of $\phi(Z; \mathbf{P})$ have been added to refine $\hat{\phi}(Z; \mathbf{P})$, and therefore in total $N_{\text{ODE}} = N_{\text{in}} + 16 = 21$ ODE solutions are needed. Compared to $N_p = 3,125$ ODE solutions needed in MCS, the proposed method is very efficient.

5.2. Dust Levitation

In this example, we still consider the same 1-D photoelectron sheath problem in Subsection 5.1, but the objective is to estimate $\bar{E}(Z)$ and $\underline{E}(Z)$ and then calculate the dust levitation height. The values of all involved parameters are given in Table 6.

The procedures used to estimate $\bar{E}(Z)$ and $\underline{E}(Z)$ are almost the same as that used to estimate $\bar{\phi}(Z)$ and $\underline{\phi}(Z)$. The only difference is that the samples of $E(Z; \mathbf{P})$ instead of $\phi(Z; \mathbf{P})$ are used. The final estimation of $\bar{E}(Z)$ and $\underline{E}(Z)$ is shown in Fig. 11. It shows that the proposed

AKM method is very accurate. As for the efficiency, the proposed method needs only $N_{\text{ODE}} = N_{\text{in}} + 18 = 23$ ODE solutions. Compared to $N_p = 3,125$ ODE solutions needed in MCS, the proposed method is very efficient.

When the upper and lower bounds of the electric field have been determined, we can use them to determine the levitation heights of the dust grains. Assuming there are two types of dust grains, A and B, in the electric field. The relevant parameters of the grains are given in Table 7, where $e = 1.062 \times 10^{-19}\text{C}$ is the electric charge of an electron. The dust levitation heights are shown in Fig. 12 and given in Table 8. Due to the uncertainty of \mathbf{P} , the levitation heights of both A and B are also uncertain. The levitation height of A may be any value in the interval $[0 \text{ m}, 9.33 \text{ m}]$, which is estimated by the proposed method. The interval determined by MCS is $[0 \text{ m}, 9.26 \text{ m}]$. It shows that the proposed method can estimate the levitation height of Grain A with sufficient accuracy. Similar conclusion applies to the levitation height of Grain B.

Given any dust grain with known w value, we can easily determine its levitation height interval using the method shown in Fig. 12. This will help to evaluate the risk or damage caused by the levitated dust grains for lunar exploration missions.

6. CONCLUSIONS

We presented an adaptive Kriging based method to perform UQ analysis of the 1-D photoelectron sheath and dust levitation on the lunar surface. A recently derived 1-D photoelectron sheath model was used as the high-fidelity physics-based model and the black-box function. Adaptive Kriging method, with a task-oriented learning function and stopping

criterion, was utilized to improve the efficiency in calculating the upper and lower bounds of electric potential as well as dust levitation height, given the intervals of model parameters. Experiment analysis shows that the proposed AKM method is both accurate and efficient. Current and ongoing efforts are focused on building adaptive Kriging model for 2-D and 3-D kinetic particle simulations of lunar plasma/dust environment and perform UQ analysis.

ACKNOWLEDGMENT

We would like to acknowledge the support from NASA-Missouri Space Grant Consortium through NASA-EPSCoR Missouri and NSF-CMMI #1923799.

REFERENCES

- [1] Der Kiureghian, A., and Ditlevsen, O., 2009, "Aleatory or epistemic? Does it matter?," *Structural safety*, 31(2), pp. 105-112.
- [2] Williams, C. K., and Rasmussen, C. E., 2006, *Gaussian processes for machine learning*, MIT Press Cambridge, MA.
- [3] Zhao, J., Wei, X., Hu, Z., He, X., Han, D., Hu, Z., and Du, X., "Photoelectron Sheath near the Lunar Surface: Fully Kinetic Modeling and Uncertainty Quantification Analysis," *Proc. AIAA Scitech 2020 Forum*, p. 1548.
- [4] Zhao, J., Wei, X., Du, X., He, X., and Han, D., 2020, "Photoelectron Sheath on the Lunar Surface: Analytic Solutions and Fully-Kinetic Particle-in-Cell Simulations for Maxwellian and Kappa Distributions," submitted.
- [5] Nitter, T., Havnes, O., and Melandsø, F., 1998, "Levitation and dynamics of charged dust in the photoelectron sheath above surfaces in space," *Journal of Geophysical Research: Space Physics*, 103(A4), pp. 6605-6620.
- [6] Fu, J. H., 1971, "Surface potential of a photoemitting plate," *Journal of Geophysical Research*, 76(10), pp. 2506-2509.
- [7] Poppe, A., and Horányi, M., 2010, "Simulations of the photoelectron sheath and dust levitation on the lunar surface," *Journal of Geophysical Research: Space Physics*, 115(A8).
- [8] Wang, J., He, X., and Cao, Y., 2008, "Modeling electrostatic levitation of dust particles on lunar surface," *IEEE transactions on plasma science*, 36(5), pp. 2459-2466.
- [9] Hirt, C., and Featherstone, W., 2012, "A 1.5 km-resolution gravity field model of the Moon," *Earth and Planetary Science Letters*, 329, pp. 22-30.

- [10] Snelling, A. A., and Rush, D. E., 1993, "Moon dust and the age of the solar system," *Creation Ex Nihilo Technical Journal*, 7(1), pp. 2-42.
- [11] Mooney, C. Z., 1997, *Monte carlo simulation*, Sage Publications, Thousand Oaks.
- [12] Rasmussen, C. E., 2004, "Gaussian processes in machine learning," *Advanced lectures on machine learning*, Springer, pp. 63-71.
- [13] Jeong, S., Murayama, M., and Yamamoto, K., 2005, "Efficient optimization design method using kriging model," *Journal of aircraft*, 42(2), pp. 413-420.
- [14] Wei, X., and Du, X., 2020, "Robustness Metric for Robust Design Optimization Under Time-and Space-Dependent Uncertainty Through Metamodeling," *Journal of Mechanical Design*, 142(3).
- [15] Shi, Y., Lu, Z., Xu, L., and Chen, S., 2019, "An adaptive multiple-Kriging-surrogate method for time-dependent reliability analysis," *Applied Mathematical Modelling*, 70, pp. 545-571.
- [16] Moustapha, M., and Sudret, B., 2019, "Surrogate-assisted reliability-based design optimization: a survey and a unified modular framework," *Structural and Multidisciplinary Optimization*, pp. 1-20.
- [17] Yun, W., Zhenzhou, L., Zhou, Y., and Jiang, X., 2018, AK-SYSi: an improved adaptive Kriging model for system reliability analysis with multiple failure modes by a refined U learning function.
- [18] Sun, Z., Wang, J., Li, R., and Tong, C., 2017, "LIF: A new Kriging based learning function and its application to structural reliability analysis," *Reliability Engineering & System Safety*, 157, pp. 152-165.
- [19] Ma, J., Ren, Z., Zhao, G., Zhang, Y., and Koh, C.-S., 2017, A New Reliability Analysis Method Combining Adaptive Kriging With Weight Index Monte Carlo Simulation.
- [20] Zhu, Z., and Du, X., "A System Reliability Method With Dependent Kriging Predictions," *Proc. ASME 2016 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, p. V02BT03A043.
- [21] Zhu, Z., and Du, X., 2016, "Reliability analysis with Monte Carlo simulation and dependent Kriging predictions," *Journal of Mechanical Design*, 138(12), p. 121403.
- [22] Hu, Z., and Mahadevan, S., 2016, "A single-loop kriging surrogate modeling for time-dependent reliability analysis," *Journal of Mechanical Design*, 138(6), p. 061406.
- [23] Zhang, L., Lu, Z., and Wang, P., 2015, "Efficient structural reliability analysis method based on advanced Kriging model," *Applied Mathematical Modelling*, 39(2), pp. 781-793.
- [24] Lv, Z., Lu, Z., and Wang, P., 2015, "A new learning function for Kriging and its applications to solve reliability problems in engineering," *Computers & Mathematics with Applications*, 70(5), pp. 1182-1197.
- [25] Cheng, J., Liu, Z., Wu, Z., Li, X., and Tan, J., 2015, "Robust optimization of structural dynamic characteristics based on adaptive Kriging model and CNSGA," *Structural and Multidisciplinary Optimization*, 51(2), pp. 423-437.
- [26] Echard, B., Gayton, N., and Lemaire, M., 2011, "AK-MCS: an active learning reliability method combining Kriging and Monte Carlo simulation," *Structural Safety*, 33(2), pp. 145-154.
- [27] Dubourg, V., Sudret, B., and Bourinet, J.-M., 2011, "Reliability-based design optimization using kriging surrogates and subset simulation," *Structural and Multidisciplinary Optimization*, 44(5), pp. 673-690.
- [28] Jones, D. R., Schonlau, M., and Welch, W. J., 1998, "Efficient global optimization of expensive black-box functions," *Journal of Global Optimization*, 13(4), pp. 455-492.

- [29] Box, G. E., and Tiao, G. C., 2011, Bayesian inference in statistical analysis, John Wiley & Sons.
- [30] Browne, M. W., 2000, "Cross-validation methods," *Journal of mathematical psychology*, 44(1), pp. 108-132.
- [31] Huang, X., Chen, J., and Zhu, H., 2016, "Assessing small failure probabilities by AK–SS: An active learning method combining Kriging and Subset Simulation," *Structural Safety*, 59, pp. 86-95.
- [32] Wen, Z., Pei, H., Liu, H., and Yue, Z., 2016, "A Sequential Kriging reliability analysis method with characteristics of adaptive sampling regions and parallelizability," *Reliability Engineering & System Safety*, 153, pp. 170-179.
- [33] Yu, Z., Sun, Z., Wang, J., and Chai, X., 2018, A new Kriging-based DoE strategy and its application to structural reliability analysis.
- [34] Cheng, K., Lu, Z., Ling, C., Zhou, S. J. S., and Optimization, M., 2020, "Surrogate-assisted global sensitivity analysis: an overview," pp. 1-27.
- [35] Zhou, H., Zhou, Q., Liu, C., and Zhou, T., 2018, "A kriging metamodel-assisted robust optimization method based on a reverse model," *Engineering Optimization*, 50(2), pp. 253-272.
- [36] Wu, H., Zhu, Z., and Du, X., 2020, "System Reliability Analysis With Autocorrelated Kriging Predictions," *Journal of Mechanical Design*, 142(10).
- [37] Chen, W., Tsui, K.-L., Allen, J. K., and Mistree, F., 1995, "Integration of the response surface methodology with the compromise decision support problem in developing a general robust design procedure," *ASME Design Engineering Technical Conference*, pp. 485-492.
- [38] Hosder, S., Walters, R., and Balch, M., 2007, "Efficient sampling for non-intrusive polynomial chaos applications with multiple uncertain input variables," *Proc. 48th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*, p. 1939.
- [39] Lophaven, S. N., Nielsen, H. B., and Søndergaard, J., 2002, Aspects of the matlab toolbox DACE, Citeseer, Technical University of Denmark.

Figure Captions List

- Fig. 1 Three types of sheath potential profiles in the analytic 1-D photoelectron sheath model [2]
- Fig. 2 A typical Type C sample of $E(Z; \mathbf{P})$
- Fig. 3 Method to solve for the equilibrium height of dust levitation
- Fig. 4 Original Kriging model: Prediction error is large near $x = 4$ and $x = 8$
- Fig. 5 Updated Kriging model with one more training sample added at $x = 8$:
Overall prediction accuracy is improved significantly
- Fig. 6 Brief flowchart of AKM
- Fig. 7 Brief flowchart of the proposed method
- Fig. 8 Initial samples of $\phi(Z; \mathbf{P})$
- Fig. 9 Results by initial Kriging model: Predicted electric potential bounds are not accurate
- Fig. 10 Final result: Predicted electric potential bounds are accurate
- Fig. 11 Final result: Predicted electric field bounds are accurate
- Fig. 12 Dust levitation heights: The electric field bounds determines the dust levitation heights

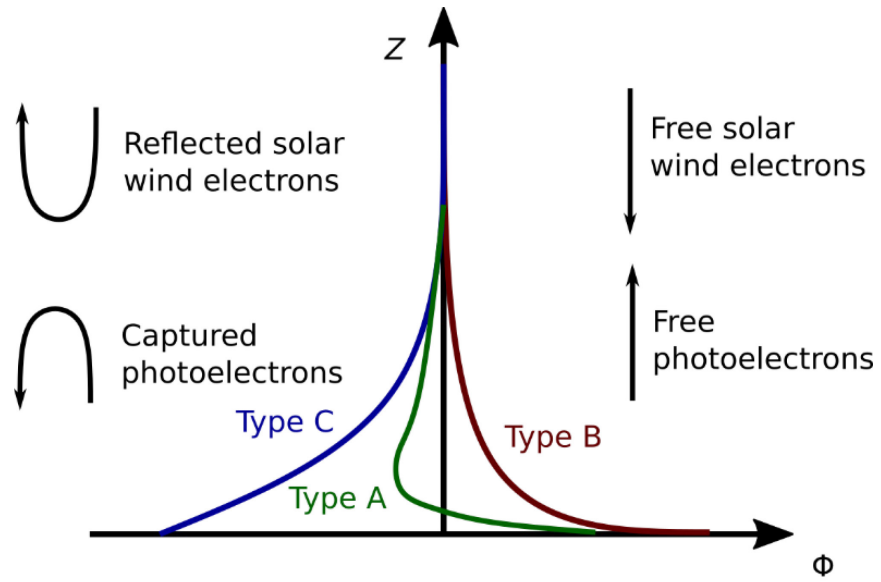


Fig. 1: Three types of sheath potential profiles in the analytic 1-D photoelectron sheath model [2]

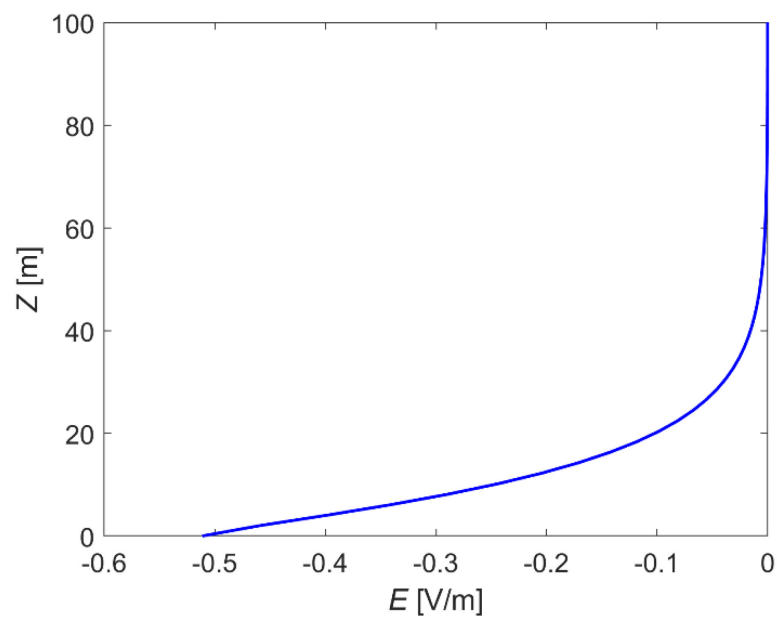


Fig. 2: A typical Type C sample of $E(Z; \mathbf{P})$

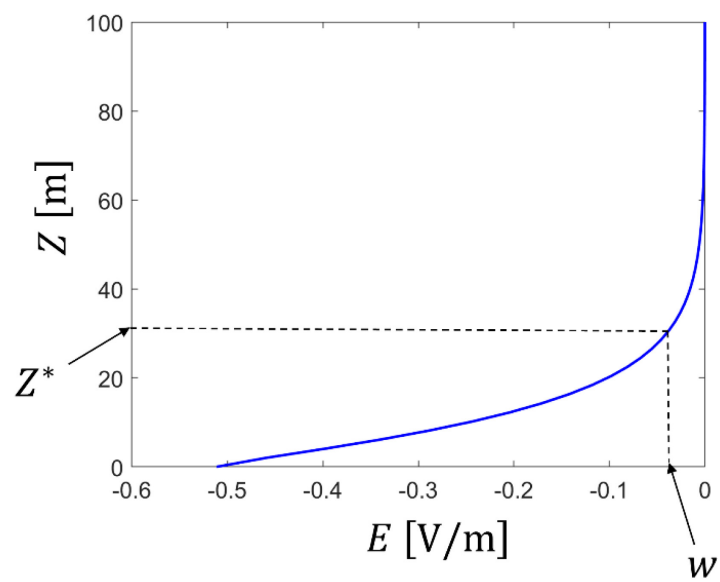


Fig. 3: Method to solve for the equilibrium height of dust levitation

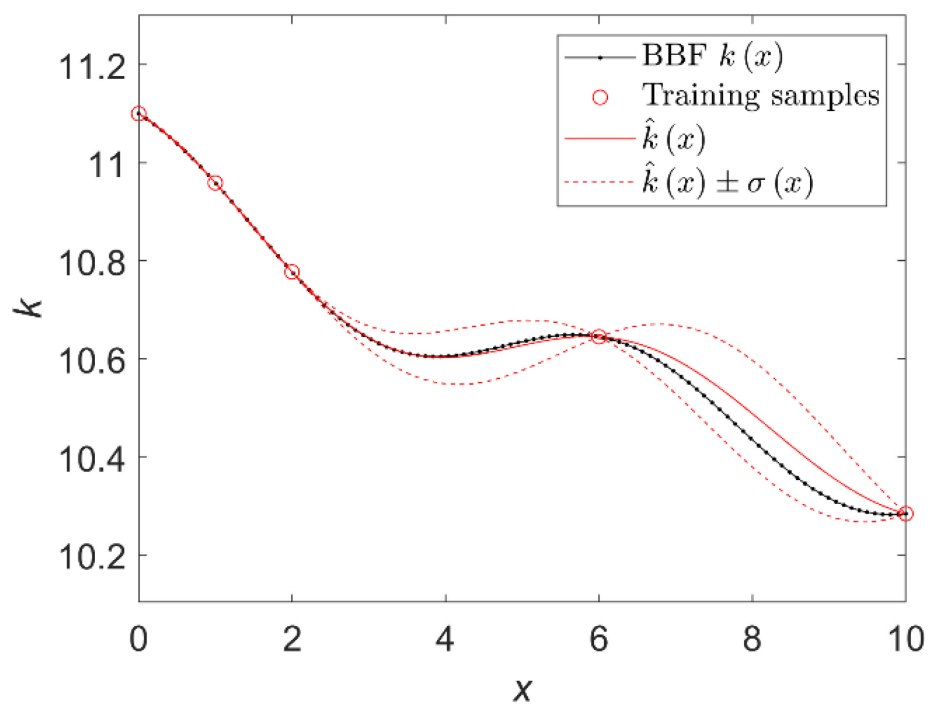


Fig. 4: Original Kriging model: Prediction error is large near $x = 4$ and $x = 8$

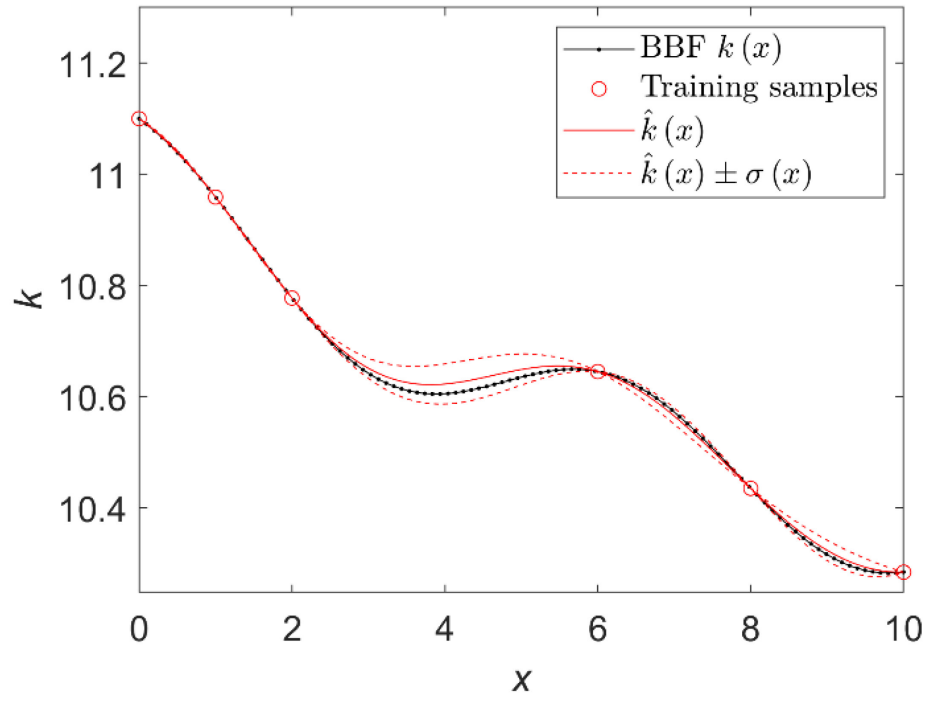


Fig. 5: Updated Kriging model with one more training sample added at $x = 8$: Overall prediction accuracy is improved significantly

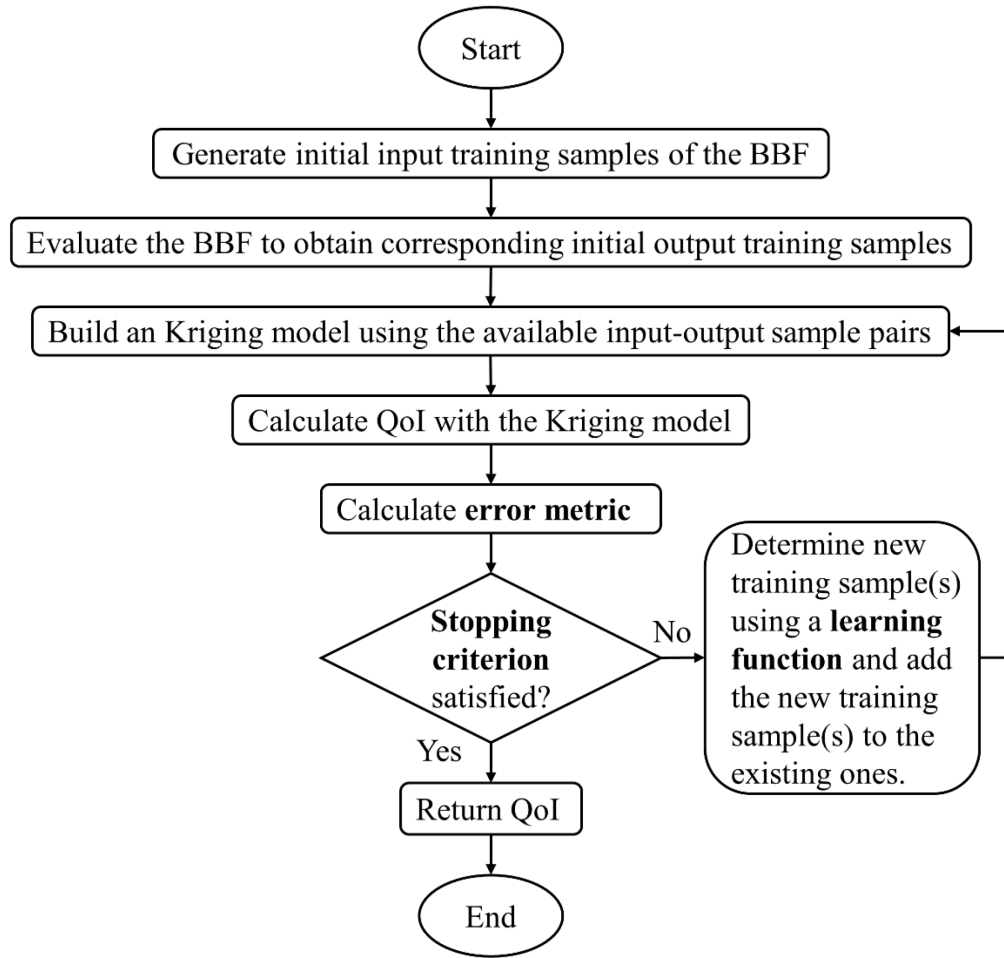


Fig. 6: Brief flowchart of AKM

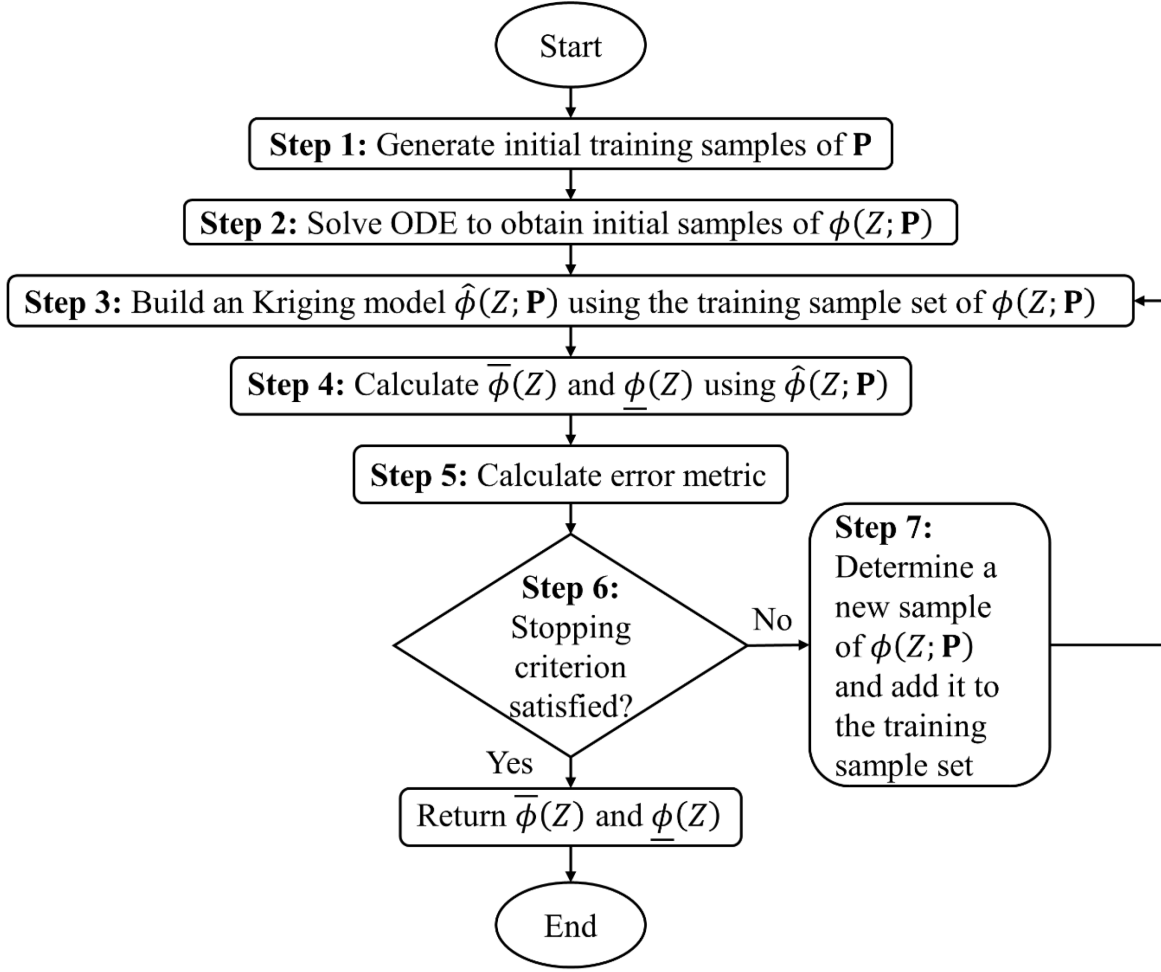


Fig. 7: Brief flowchart of the proposed method

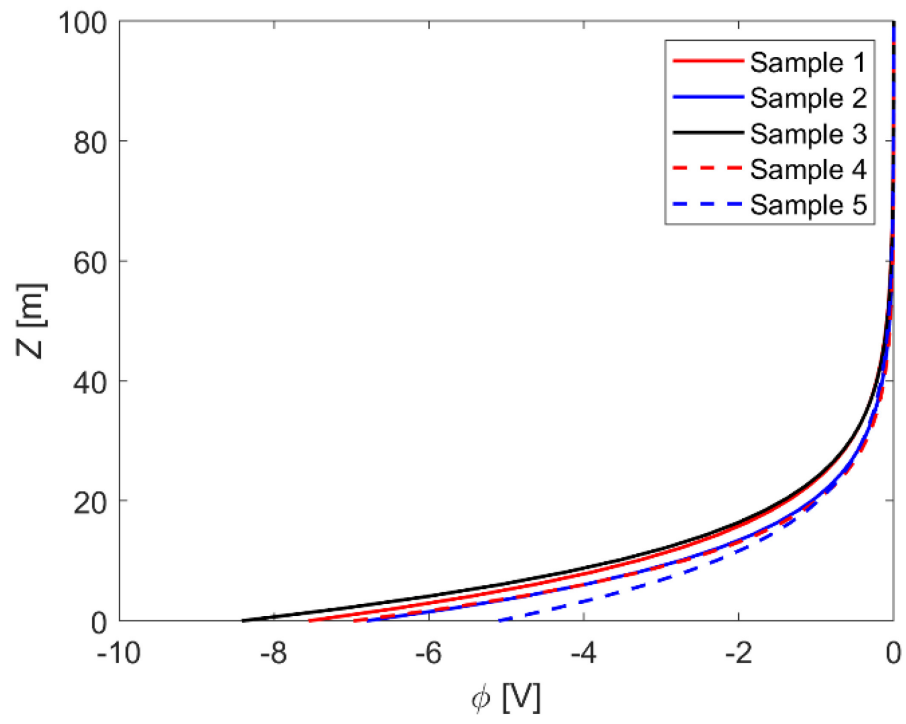


Fig. 8: Initial samples of $\phi(Z; \mathbf{P})$

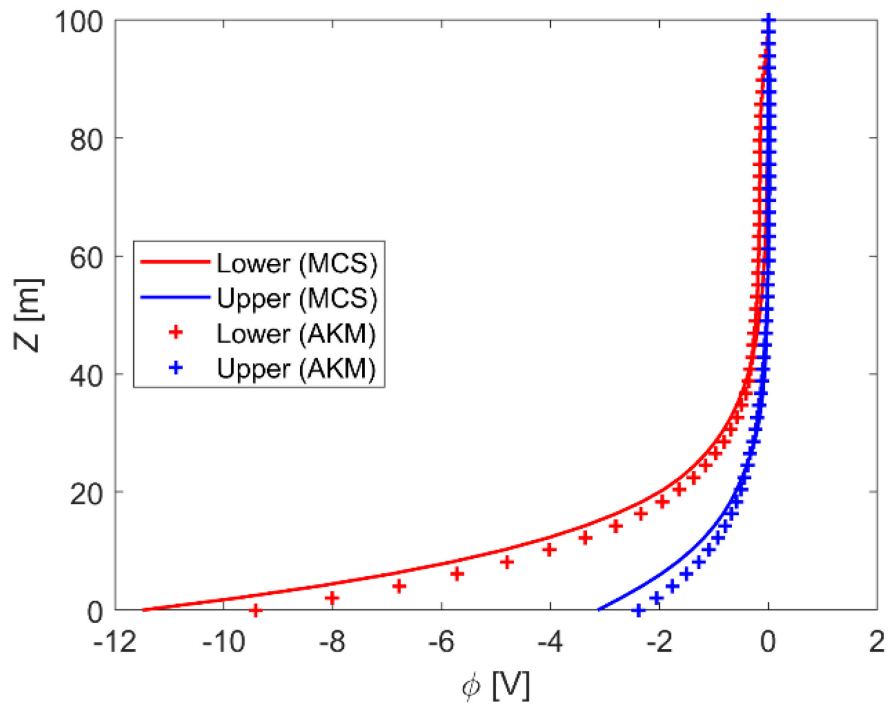


Fig. 9: Results by initial Kriging model: Predicted electric potential bounds are not accurate

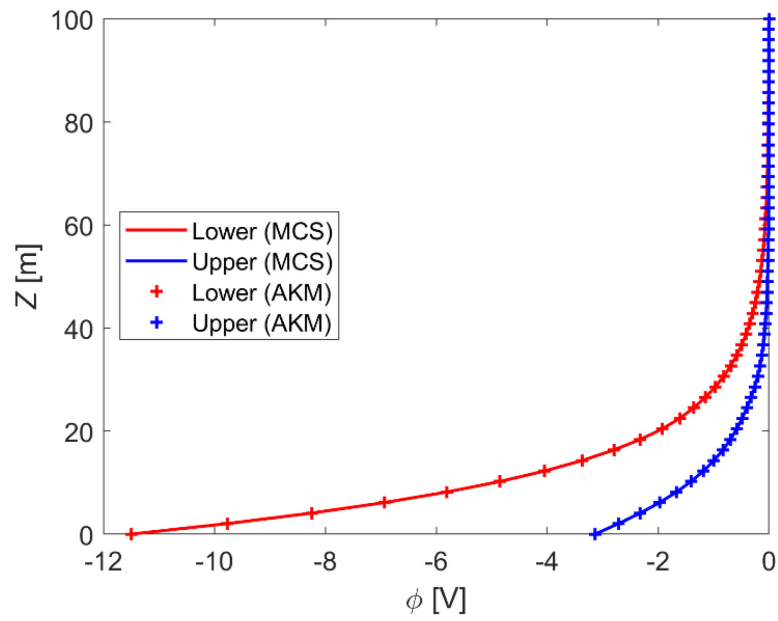


Fig. 10: Final result: Predicted electric potential bounds are accurate

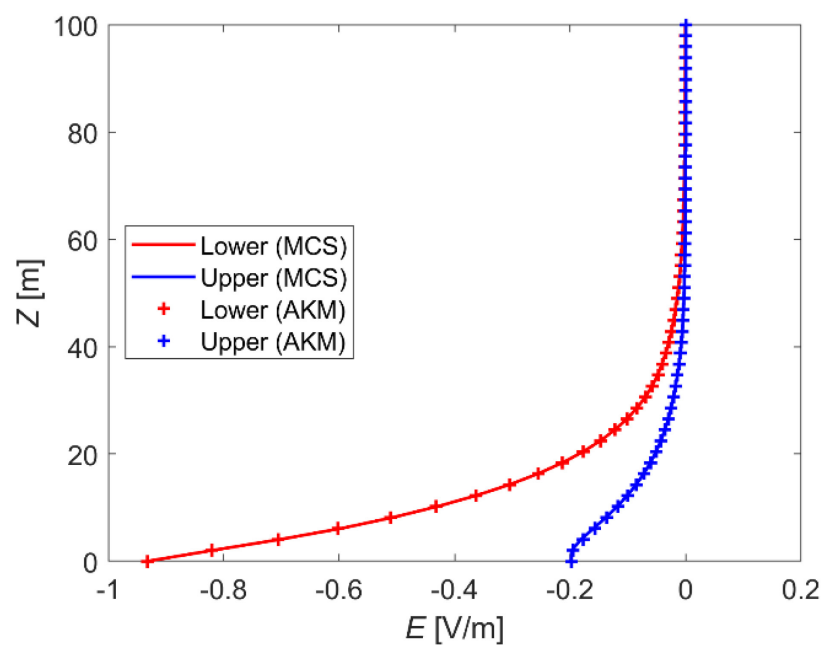


Fig. 11: Final result: Predicted electric field bounds are accurate

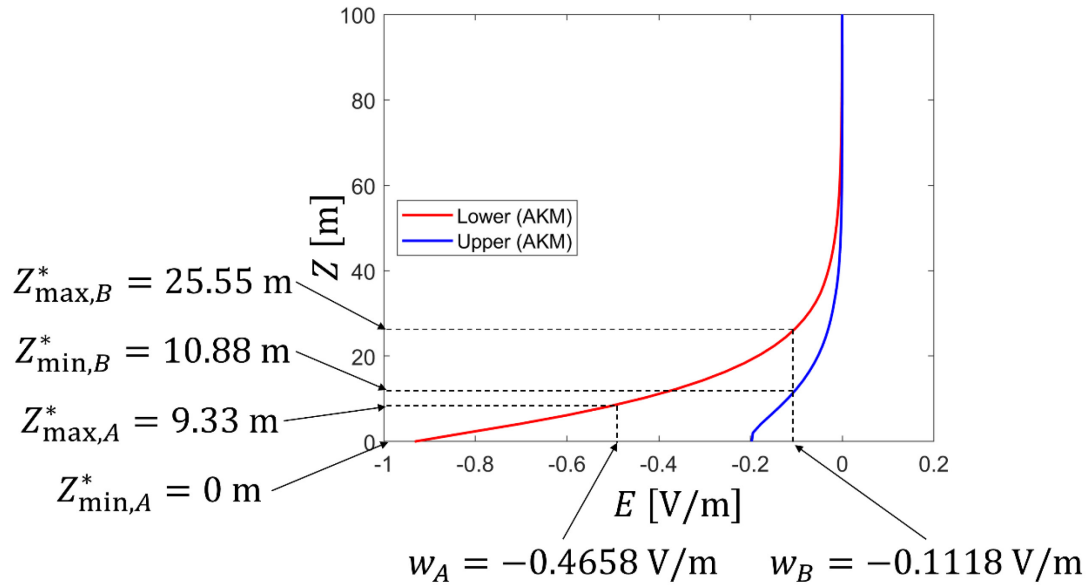


Fig. 12: Dust levitation heights: The electric field bounds determines the dust levitation heights

Table Caption List

Algorithm 1	Pseudo codes of the proposed method
Table 1	Variables of uncertainty
Table 2	Parameter values
Table 3	Samples generated by Hammersley sampling method
Table 4	Samples mapped into Ω
Table 5	Initial samples of P
Table 6	Parameter values
Table 7	Parameters of Grains A and B
Table 8	Dust levitation heights: The proposed AKM obtained accurate levitation heights

Algorithm 1: Pseudo codes of the proposed method

Row	Pseudo codes
1	Evenly discretize Ω into N_P points \mathbf{p}^{MCS} .
2	Evenly discretize interval $[Z_{\min}, Z_{\max}]$ into N_Z points \mathbf{z}^{MCS} .
3	Generate N_{in} samples \mathbf{p}^{in} of \mathbf{P} with procedures given in Subsection 4.2.
4	Solve ODE N_{in} times to get N_{in} samples $\phi(Z; \mathbf{p})$, $\mathbf{p} \in \mathbf{p}^{\text{in}}$ of $\phi(Z; \mathbf{P})$; Calculate ϵ with Eq. (29); $N_{\text{ODE}} = N_{\text{in}}$.
5	Determine $(\mathbf{x}^{\text{inp}}, \boldsymbol{\phi}^{\text{out}})$ with procedures given in Subsection 4.2; $\boldsymbol{\phi}^{\text{out}} = \boldsymbol{\phi}^{\text{out}} + \epsilon$.
6	WHILE TRUE DO
7	Build Kriging model $\phi(Z; \mathbf{P})$ using $(\mathbf{x}^{\text{inp}}, \boldsymbol{\phi}^{\text{out}})$.
8	Calculate $\bar{\phi}(Z)$ and $\underline{\phi}(Z)$ with Eq. (16) and Eq. (17); $\bar{\phi}(Z) = \bar{\phi}(Z) - \epsilon$; $\underline{\phi}(Z) = \underline{\phi}(Z) - \epsilon$.
9	Calculate Γ with Eq. (27).
10	IF ($\Gamma \geq \gamma$) DO
11	Solve Eq. (20) for $\mathbf{p}^{(\text{next})}$; $N_{\text{ODE}} = N_{\text{ODE}} + 1$. Solve ODE to get a new sample $\phi(Z; \mathbf{p}^{(\text{next})})$; $\phi(Z; \mathbf{p}^{(\text{next})}) =$
12	$\phi(Z; \mathbf{p}^{(\text{next})}) + \epsilon$; All points of $\phi(Z; \mathbf{p}^{(\text{next})})$ excluding the one at Z_{\max} are added into $(\mathbf{x}^{\text{inp}}, \boldsymbol{\phi}^{\text{out}})$.
13	ELSE
14	BREAK WHILE
15	END IF
16	END WHILE
17	RETURN $\bar{\phi}(Z)$, $\underline{\phi}(Z)$, and N_{ODE} .

Table 1: Variables of uncertainty

Variables	$v_d(\text{m/s})$	$T_e(\text{eV})$	$T_p(\text{eV})$	$n_{i,\infty}(\text{cm}^{-3})$	$n_p(\text{cm}^{-3})$
Minimum	421,200	10.8	1.8	7.83	57.6
Maximum	414,800	13.2	2.2	9.57	70.4

Table 2: Parameter values

Parameters	$N_1 \sim N_5$	N_P	N_{in}	N'_P	N_Z	γ
Values	5	5^5	5	100	50	0.01

Table 3: Samples generated by Hammersley sampling method

Sample number	Dimension 1	Dimension 2	Dimension 3	Dimension 4	Dimension 5
1	0	0.5000	0.3333	0.2000	0.1429
2	0.2	0.2500	0.6667	0.4000	0.2857
3	0.4	0.7500	0.1111	0.6000	0.4286
4	0.6	0.1250	0.4444	0.8000	0.5714
5	0.8	0.6250	0.7778	0.0400	0.7143

Table 4: Samples mapped into Ω

Sample number	$v_d(\text{m/s})$	$T_e(\text{eV})$	$T_p(\text{eV})$	$n_{i,\infty}(\text{ cm}^{-3})$	$n_p(\text{ cm}^{-3})$
1	421,200	12.0000	1.9333	8.1780	59.4286
2	439,920	11.4000	2.0667	8.5260	61.2571
3	458,640	12.6000	1.8444	8.8740	63.0857
4	477,360	11.1000	1.9778	9.2220	64.9143
5	496,080	12.3000	2.1111	7.8996	66.7429

Table 5: Initial samples of **P**

Sample number	$v_d(\text{m/s})$	$T_e(\text{eV})$	$T_p(\text{eV})$	$n_{i,\infty}(\text{ cm}^{-3})$	$n_p(\text{ cm}^{-3})$
1	421,200	12.0000	1.9000	8.2650	60.8000
2	444,600	11.4000	2.1000	8.7000	60.8000
3	468,000	12.6000	1.8000	8.7000	64.0000
4	468,000	11.4000	2.0000	9.1350	64.0000
5	491,400	12.0000	2.1000	7.8300	67.2000

Table 6: Parameter values

Parameters	$N_1 \sim N_5$	N_P	N_{in}	N'_P	N_Z	γ
Values	5	5^5	5	100	50	0.01

Table 7: Parameters of Grains A and B

Grains	r (μm)	m (g)	q/e	w (V/m)
A	0.5	1.5268×10^{-12}	50,000	-0.4658
B	0.3	3.2979×10^{-13}	45,000	-0.1118

Table 8: Dust levitation heights: The proposed AKM obtained accurate levitation heights

Grains		AKM	MCS	Relative error (%)
A	$Z_{\min}^*(\text{m})$	0.00	0.00	0.0
	$Z_{\max}^*(\text{m})$	9.33	9.26	0.8
B	$Z_{\min}^*(\text{m})$	10.88	11.00	-1.1
	$Z_{\max}^*(\text{m})$	25.55	25.55	0.0