

Simulation-based surrogate methodology of electric field for electrospray emitter geometry design and uncertainty quantification

IEPC-2022-271

*Presented at the 37th International Electric Propulsion Conference
Boston, MA
June 19-23, 2022*

Joshua D. Eckels, Collin B. Whittaker, Benjamin A. Jorns, and Alex A. Gorodetsky
University of Michigan, Ann Arbor, MI, 48109

Benjamin St. Peter and Rainer A. Dressler
Spectral Sciences Inc., Burlington, MA 01803

Electrospray thruster technology requires the use of arrayed emitter and extractor grid structures to scale thrust to mission operation levels. With recent advances in manufacturing capabilities of porous-media emitter substrates, a design space opens where more control is given over the geometry of individual emitters in an arrayed structure. For many electrospray design applications, a quantity of interest for evaluating performance and efficiency of emitters is the peak electric field strength near the tip of the emitter. The mapping between the geometry design space of emitters and the resulting electric field for a given bias voltage is not analytically available; this study proposes a simulation and surrogate methodology for the prediction of peak electric field strength based on the geometry parameters of an individual emitter. An electrostatic partial differential equation solver was used to generate an electric field solution at all design considerations. The maximum electric field magnitude was then extracted and used as training data for a surrogate model to be used in an engineering toolkit for mapping geometry parameters to maximum electric field. The surrogate was found to significantly outperform existing models in the toolkit on average over the design space considered, with a median error of 6% vs. 160% for the existing models. Finally, the computational advantage of the surrogate enabled a global sensitivity analysis of the peak electric field. Variance in output electric field predictions was found to be most attributable to uncertainties in the radius of curvature at the tip of the emitter and the slant angle of the emitter cone.

I. Introduction

Electrospray thrusters can potentially offer immense performance and lifetime benefits for in-space propulsion, but are limited in operational scale by low thrust levels [1]. To scale electrospray thruster technology to small and medium spacecraft level operations, it is necessary to arrange emitters in an arrayed fashion to increase total thrust levels. The design of multiplexed structures naturally leads to the question of how the design of individual emitter elements affects the overall performance of the arrayed structure. Specifically, the electric field distribution near the surface of the emitter is not analytically available as a function of the emitter geometry. Knowledge or prediction of the electric field near the emitter is often necessary for accurate predictions of performance metrics such as thrust, efficiency, etc. or for the prediction of multi-site emissions [2]. This study seeks to generate an efficient surrogate model to obtain these predictions.

The Electrospray Propulsion Engineering Toolkit (ESPET) is a numerical tool for performance estimation of electrospray thrusters [3, 4]. ESPET provides a database of propellant properties and a suite of engineering models

to solve flow properties and provide high-level performance metrics based on a given thruster design and a set of associated model and parameter uncertainties. Several studies have demonstrated the use of ESPET for experimental design of emitters and uncertainty quantification (UQ) [2, 5–7]. Presently, the underlying numerical model in ESPET for electric field prediction is based on the hyperboloid approximation for the conical shape of the emitter, as presented by Martinez-Sanchez [8]. The Martinez-Sánchez approximation provides an analytical solution for the electrostatic field between an infinitely-long hyperboloid geometry and an infinite flat plate; however, several aspects of emitter design, including cone shape and extractor grid aperture radii, are not accounted for in the approximation. To study the effects of emitter geometry design on arrayed thruster performance, a more realistic geometry model is required.

This study proposes a simulation-based surrogate methodology for electric field prediction and applies it in a sensitivity analysis of emitter geometry parameters. First, several sets of parameters are sampled from a design space expanded from [5]. Then, an electrostatics numerical simulation is developed and used to generate electric field data for each of the samples. A feedforward neural network (FNN) is trained as a surrogate for the simulation to map emitter geometry to peak electric field strength. The performance of the surrogate is studied on a test set and on a suite of parameter sweep case studies. Finally, the surrogate is used in a sensitivity analysis of the electric field.

II. Methodology

An overview of the surrogate development and analysis methodology is provided in this section. This overview is summarized in Figure 1.

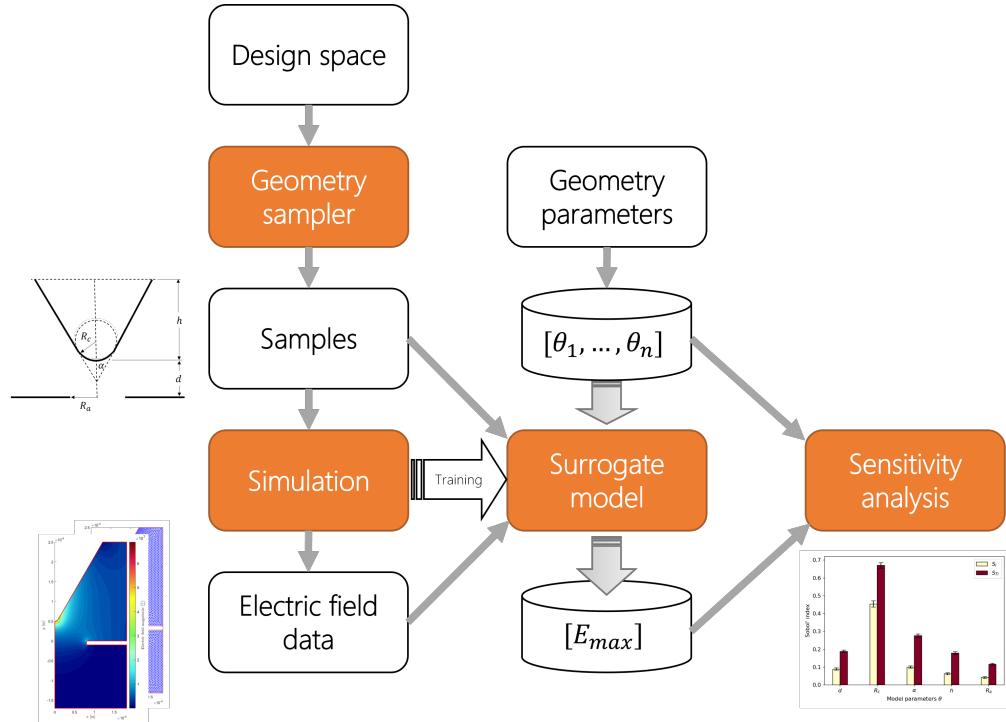


Fig. 1 Overview of the simulation-based surrogate methodology for electric field prediction and sensitivity analysis

First, a geometry model is developed with a design space for its parameters. The design space is explored with a rejection sampling technique to generate samples of the parameters that meet a set of practical design constraints [5]. The samples are used to construct a finite-element analysis simulation of the emitter in an electrostatic field, producing a set of electric field data for each geometry sample. An FNN is trained on the electric field data to form the surrogate model. Finally, the trained surrogate is used in a sensitivity analysis to quantify the geometry parameters that are most influential to the variance of the output electric field predictions.

A. Emitter geometry model

The proposed emitter geometry model is a conical shape with a spherical cap at the tip that is centered above a circular aperture in a rectangular grid extractor plate [5], as shown in Figure 2. There are 5 emitter geometry design parameters under consideration: the height of the emitter h , the cone half-angle α , the radius of curvature at the tip R_c , the tip-to-extractor distance d , and the radius of the aperture R_a in the extractor grid. The range considered for each geometry parameter in this study is summarized in Table 1.

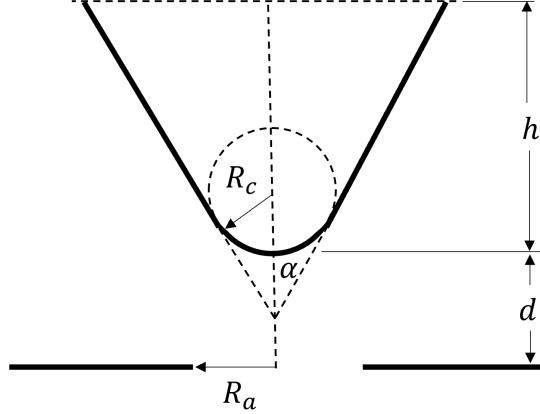


Fig. 2 Emitter geometry model [5]

Table 1 Emitter geometry parameters

Parameter	Symbol	Range
Emitter height	h	[50, 1000] μm
Cone half-angle	α	[10, 70] deg
Radius of curvature	R_c	[1, 100] μm
Tip-to-extractor distance	d	[-1000, 3000] μm
Radius of aperture	R_a	[10, 3000] μm

The bounds were chosen as practical manufacturing limits, such as might be used for thrusters similar to the AFET-2 electrospray thruster [9]. The negative tip-to-extractor distance lower bound indicates an emitter that extends through the extractor grid aperture. To ensure a well-formed emitter, two additional practical constraints are imposed on the geometry parameters. The first constraint follows from [5], where the lower bound on the height of the emitter is the height of a spherical cap:

$$h > h_l, \quad \text{where } h_l = R_c(1 - \sin(\alpha)). \quad (1)$$

Second, for emitters that extend through the extractor grid (i.e. $d < 0$), the emitter must not contact the extractor grid. An illustration of this case is shown in Figure 3 (reversed in orientation for ease in notation).

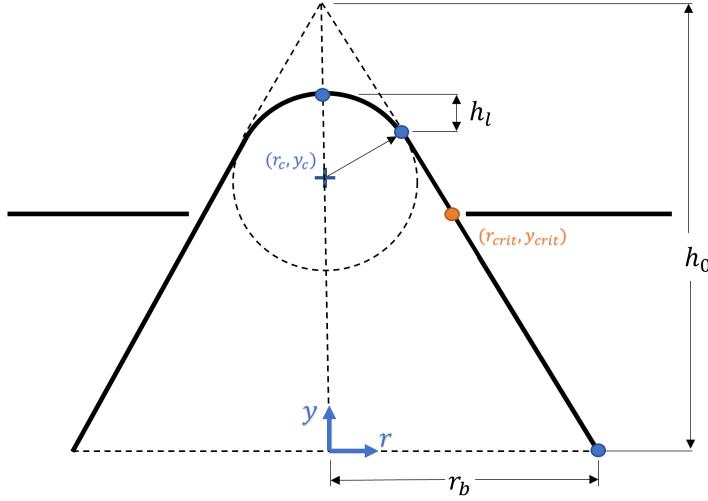


Fig. 3 Coordinate system for an emitter that extends through the extractor grid

A coordinate system (r, y) is centered at the base of the emitter cone and the critical point of contact is located at $(r_{\text{crit}}, y_{\text{crit}})$. In the chosen coordinate system, the non-contact constraint is specified as

$$r_{\text{crit}} < R_a \quad (2)$$

where R_a is the radius of the aperture. If the cross-sectional radius of the emitter is parameterized as $r(y)$, then the critical radius r_{crit} is found by

$$r_{\text{crit}} = r(y = y_{\text{crit}}) = r(y = h - |d|) \quad (3)$$

where h is the height of the emitter and d is the tip-to-extractor distance. The cross-sectional radius can be parameterized by the piecewise function:

$$r(y) = \begin{cases} r_b - \frac{r_b}{h_0}y & 0 \leq y < h - h_l \\ \sqrt{R_c^2 - (y - y_c)^2} + r_c & h - h_l \leq y \leq h \end{cases} \quad (4)$$

where the second piece comes from a circle of radius R_c centered at (r_c, y_c) , with $r_c = 0$ and $y_c = h - R_c$ in the chosen coordinate system. The base radius of the cone is given by

$$r_b = R_c \cos \alpha + (h - h_l) \tan \alpha \quad (5)$$

and the cone height is given by

$$h_0 = \frac{r_b}{\tan \alpha}. \quad (6)$$

Samples of the geometry are generated that fill the five dimensional input space using the Latin hypercube sampling approach [10]. The samples that do not satisfy the two specified practical constraints are then rejected. In this way, a space of satisfactory emitter geometry designs is generated that spans the input space under the given constraints.

B. Simulation technique

A 2d axisymmetric electrostatic simulation of the emitter geometry in Figure 2 is performed to obtain the distribution of the electric field near the surface of the emitter. Matlab's partial differential equation (PDE) toolbox is used to solve the electrostatic Laplace equation:

$$\nabla^2 \Phi = 0 \quad (7)$$

where Φ is the spatial distribution of electric potential, and 0 charge density is assumed in between the emitter and the extractor, since no ions are yet emitted at the immediate onset of the potential difference. A bias voltage of V_0 is applied

to the emitter surface relative to the extractor grid, and a far-field potential of 0V is applied at the bottom of the solver domain. The electric field is obtained by the solver using the electrostatic relation:

$$\vec{E} = -\nabla\Phi. \quad (8)$$

An example geometry mesh and electric field magnitude solution obtained from the solver is shown in Figure 4 (with the emitter, extractor grid, and boundary conditions (BCs) labeled).

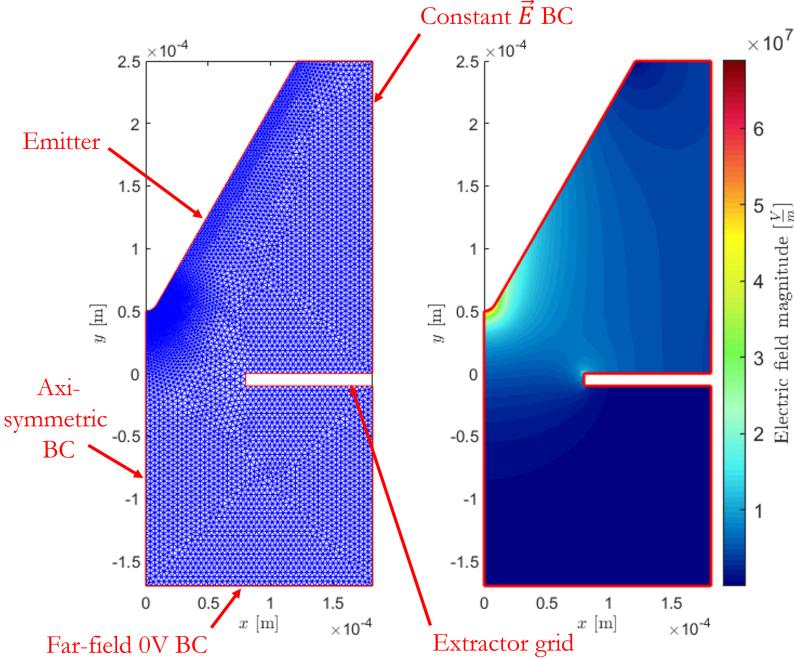


Fig. 4 An example geometry mesh and electric field solution

An axisymmetric BC is imposed through the center of the emitter cone and a constant electric field directed vertically down from the top of the domain to the extractor grid surface is applied at the far right of the domain, assuming sufficient distance away from the emitter cone region.

One challenge of automating this simulation procedure over the geometry design space in Table 1 is ensuring mesh convergence over orders of magnitude of change in length scale. Since the region of interest in this study is the field solution at the emitter surface, it was only necessary to design the mesh to be sufficiently refined around the tip of the cone, where potential gradients are the largest. The mesh in the rest of the field was allowed to grow with the dimension of the domain to reduce computational cost.

An emitter geometry based on the AFET-2 electrospray thruster [9] was used to validate the simulation procedure. The geometry parameters of this thruster are summarized in Table 2.

Table 2 AFET-2 geometry parameters [9]

Parameter	Symbol	Value
Emitter height	h	100 μm
Cone half-angle	α	15 deg
Radius of curvature	R_c	10 μm
Tip-to-extractor distance	d	5 μm
Radius of aperture	R_a	254 μm

To validate the simulation procedure on this design, a hyperboloid approximation of the design is constructed and the analytical Martinez-Sanchez electric field solution on the surface of the hyperboloid geometry is obtained [8]. It is noted that the hyperboloid solution only takes into account the tip-to-extractor distance and the tip radius of curvature of the original design (in Table 2); the hyperboloid is assumed to lie above an infinite flat plate at $y = 0$, so the extractor grid aperture is not included in this simulation. The hyperboloid geometry is simulated with mesh sizes ranging from 1×10^{-7} m to 1×10^{-5} m. The convergence of the electric field magnitude at the tip of the hyperboloid to the analytical solution is shown in Figure 5.

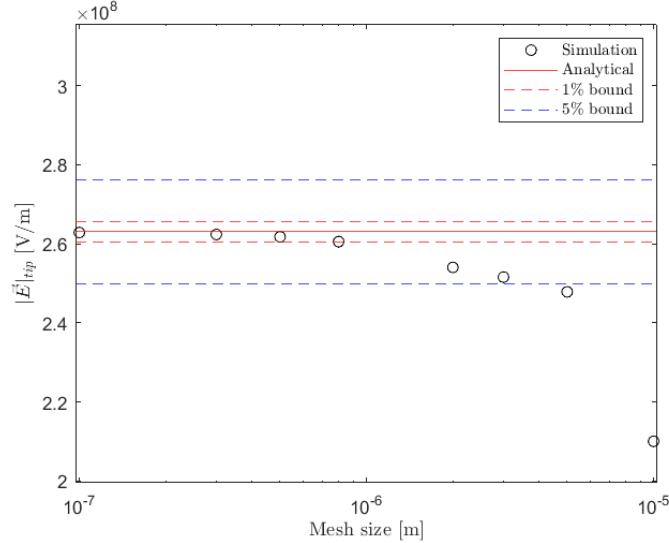


Fig. 5 Convergence of simulation to analytical solution within 5% bounds (blue) and 1% bounds (red)

The field solution of a 0.5% converged 5×10^{-7} m mesh is shown in Figure 6. Note that the mesh size was allowed to grow outside the region near the surface of the emitter.

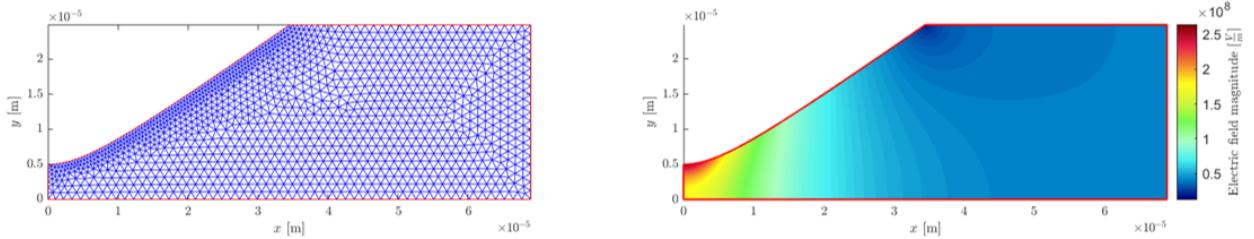


Fig. 6 0.5% converged mesh and field solution of the AFET-2 hyperboloid approximation simulation

Since the simulation procedure described here has been shown to converge to the analytical solution for the hyperboloid geometry, it is used in the rest of the study, under sufficient mesh refinement, as a high-fidelity model for electric field prediction on conical emitters. During validation, the simulations were observed to run on the order of $O(1)$ s, which is feasible for small-use cases. However, for many UQ tasks, thousands of model evaluations are required, so this study proposes a lower-fidelity, lower-cost surrogate to replace the simulation in the next section.

C. Surrogate model

To avoid the computational expense of the numerical simulation, (e.g. when many evaluations of the solver are required), this study seeks to train a feedforward neural network (FNN) as a surrogate model to approximate the map $y(\theta)$ between emitter geometry and peak electric field strength:

$$y(\theta) : [h, \alpha, R_c, d, R_a] \rightarrow \max |\vec{E}| \quad (9)$$

where $\theta = [h, \alpha, R_c, d, R_a]$ are the geometry parameters. An FNN is chosen as a naïve, simple function approximator in this study as a first-pass surrogate for the electrostatic solver. Before training the FNN, each of the length-dependent parameters were scaled by the height of the emitter, while the cone half-angle remained in dimensionless rad units. The electric field was additionally scaled by the bias voltage V_0 . After scaling, it was observed that the parameters R_c, d, R_a , and E_{\max} had highly skewed distributions, which can bias the training of the FNN and result in poor generalization. To account for this, if each parameter with a skewed distribution is treated as a random variable X with a probability density function (PDF) $f_X(x)$ and cumulative distribution function (CDF) $F_X(x)$, a one-to-one mapping $g(X)$ is sought that redistributes X as a uniform random variable U on the interval $[0, 1]$:

$$g(X) : \mathbb{R} \rightarrow [0, 1], \quad \text{s.t.} \quad g(X) \stackrel{d}{=} U. \quad (10)$$

From the definition of a CDF:

$$F_X(x) = P(X \leq x) \quad (11)$$

$$= P(g^{-1}(U) \leq x) \quad (12)$$

$$= P(U \leq g(x)) \quad (13)$$

$$= g(x) \quad (14)$$

from which the desired one-to-one mapping $g(x)$ is simply given by the CDF of X , i.e. $g(x) = F_X(x)$. In the current study, each of the parameters R_c, d, R_a , and E_{\max} , were observed to approximately follow an exponential distribution after being scaled, which is parameterized by the constant λ :

$$f_X(x) = \lambda e^{-\lambda x}, \quad \text{and} \quad (15)$$

$$F_X(x) = 1 - e^{-\lambda x}. \quad (16)$$

For each of these parameters, an exponential PDF was fit to the empirical distribution of the training data to obtain the fit constant λ , and then the CDF (Eq. (16)) was applied to map the parameters to an approximately uniform distribution. A summary of this scaling procedure is shown in Table 3, and the results of the empirical fits are included in Appendix A. Note that an additional offset of $d_0 = 1000 \mu\text{m}$ was applied to the tip-to-extractor distance before scaling to place it in the positive reals domain of the exponential PDF, (i.e. $d + d_0 \in [0, \infty)$).

Table 3 Parameter scaling and redistribution

Parameter	Symbol	Units	Scaling
Emitter height	h	m	$\tilde{h} = 1$
Cone half-angle	α	rad	$\tilde{\alpha} = \alpha$
Radius of curvature	R_c	m	$\tilde{R}_c = g\left(\frac{R_c}{h}\right)$
Tip-to-extractor distance	d	m	$\tilde{d} = g\left(\frac{d+d_0}{h}\right)$
Radius of aperture	R_a	m	$\tilde{R}_a = g\left(\frac{R_a}{h}\right)$
Electric field strength	E_{\max}	V m^{-1}	$\tilde{E}_{\max} = g\left(\frac{E_{\max}}{V_0/h}\right)$

After scaling, the parameters were normalized linearly to the range $[-1, 1]$ according to:

$$\tilde{\theta}_n = \frac{\tilde{\theta} - \tilde{\theta}_{\min}}{\tilde{\theta}_{\max} - \tilde{\theta}_{\min}} (\tilde{\theta}_{n,\max} - \tilde{\theta}_{n,\min}) + \tilde{\theta}_{n,\min} \quad (17)$$

where $[\tilde{\theta}_{\min}, \tilde{\theta}_{\max}]$ corresponds to the range of the scaled parameters $\tilde{\theta} = [\tilde{\alpha}, \tilde{R}_c, \tilde{d}, \tilde{R}_a]$ and $[\tilde{\theta}_{n,\min}, \tilde{\theta}_{n,\max}] = [-1, 1]$ corresponds to the range of the normalized parameters $\tilde{\theta}_n$. After normalization, the FNN is only exposed to inputs and outputs that fall in the range $[-1, 1]$. It is noted that the design space of 5 geometric parameters was reduced to 4 scaled geometric parameters, since $\tilde{h} = 1 = \text{constant}$.

This study employs an FNN to approximate the map in Eq. (9), rewritten in terms of the scaled and normalized parameters as

$$y(\tilde{\theta}_n) : [\tilde{a}_n, \tilde{R}_{c,n}, \tilde{d}_n, \tilde{R}_{a,n}] \rightarrow \tilde{E}_{\max,n}. \quad (18)$$

The FNN has four network inputs, two fully-connected hidden layers, and one regression output, as summarized in Table 4. This FNN form was chosen as the simplest architecture that had the lowest training error.

Table 4 Electric field FNN surrogate summary

Parameter	Symbol	Value
Number of inputs	N_θ	4
Number of outputs	N_y	1
Number of hidden layers	h	2
Hidden layer sizes	N_k	[9, 5]
Number of trainable parameters	p_T	105

For network parameters $p \in \mathbb{R}^{p_T}$ and an input $\theta \in \mathbb{R}^{N_\theta}$, the network returns the regression output $\hat{y} \in \mathbb{R}^{N_y}$, which is an approximation of the function $y(\tilde{\theta}_n)$ in Eq. (18). For a set of N training samples $\{\theta^{(i)}, y^{(i)}\}_{i=1}^N$, where $y^{(i)}$ is the known output value of the input $\theta^{(i)}$, the cost function of the network predictor $\hat{y}(\theta)$ over the training set is expressed as the mean squared error (MSE):

$$C(p; \theta, y) = \frac{1}{N} \sum_{i=1}^N (y^{(i)} - \hat{y}(\theta^{(i)}; p))^2 \quad (19)$$

where the cost $C(p; \theta, y)$ is a function of the network parameters p for a given set of training data $\{\theta, y\}$. The task of training the network is one of optimization:

$$p^* = \underset{p}{\operatorname{argmin}} C(p; \theta, y) \quad (20)$$

where the optimal network parameters p^* are those which minimize the cost function. The FNN was trained on a dataset size of $N = 50,000$ Latin hypercube samples from the ranges in Table 1, under the constraints Eq. (1) and (2), with a training and validation split of 80/20%. More details on the FNN are provided in Appendix B.

D. Sensitivity analysis

In this study, we apply the surrogate to the task of sensitivity analysis, where we seek to quantify the uncertainty in the model output that arises from uncertainty in the inputs. Specifically, we employ the global, variance-based Sobol' method for sensitivity analysis [11], where we compute a set of indices that measure how much of the output variance can be attributed to variance in each of the input parameters, as well as to interactions between the inputs. Following [12], for a scalar function $Y = f(\theta_1, \theta_2, \dots, \theta_d)$ of d independent input parameters, the variance of the output can be decomposed as

$$V(Y) = \sum_{i=1}^d V_i + \sum_{i=1}^d \sum_{j>i}^d V_{ij} + \dots + V_{i_1\dots i_d} \quad (21)$$

where V_i is the partial variance due to the generic parameter θ_i , V_{ij} is the partial variance from the interaction of parameters θ_i and θ_j , and so on. Eq. (21) applies for a square-integrable function Y over a d -dimensional hypercube with independent input parameters. Dividing Eq. (21) by the output variance $V(Y)$, we get:

$$1 = \sum_{i=1}^d S_i + \sum_{i=1}^d \sum_{j>i}^d S_{ij} + \dots + S_{i_1\dots i_d} \quad (22)$$

where the n -th order sensitivity indices are defined as:

$$S_i = \frac{V_i}{V(Y)}, \quad n = 1 \quad (23)$$

$$S_{ij} = \frac{V_{ij}}{V(Y)}, \quad n = 2 \quad (24)$$

⋮

$$S_{ij\dots d} = \frac{V_{ij\dots d}}{V(Y)}, \quad n = d. \quad (25)$$

From Eq. (22), we see that the 1st-order sensitivity indices S_i measure what fraction of the output variance is attributable to each parameter individually, the 2nd-order indices S_{ij} measure the fraction attributable to interactions between two parameters, and so on for higher-order indices. The 1st-order partial variances are obtained from [13]:

$$V_i = V_{\theta_i} (\mathbb{E}_{\theta_{\sim i}} [Y|\theta_i]) \quad (26)$$

where the inner expectation is taken of Y for a fixed parameter θ_i over all possible values of the other parameters $\theta_{\sim i}$. The outer variance is then taken over all possible values of θ_i . Alternatively, we fix the parameter θ_i and average the response Y over variations in all the other parameters. Then we sweep θ_i over its full domain and measure the variance observed in this averaged response of Y . This observed variance corresponds to variations in the parameter θ_i alone, which is exactly the 1st-order partial variance V_i . The 2nd-order partial variances are given by

$$V_{ij} = V_{\theta_i, \theta_j} \left(\mathbb{E}_{\theta_{\sim ij}} [Y|\theta_i, \theta_j] \right) - V_i - V_j \quad (27)$$

where the first term measures the partial variance over θ_i and θ_j simultaneously, and the intersecting 1st-order variances V_i and V_j are removed to obtain the parameter interaction variance V_{ij} alone. In this study, we are also interested in the total-order sensitivity index:

$$S_{Ti} = 1 - \frac{V_{\theta_{\sim i}} (\mathbb{E}_{\theta_i} [Y|\theta_{\sim i}])}{V(Y)} \quad (28)$$

where $V_{\theta_{\sim i}} (\mathbb{E}_{\theta_i} [Y|\theta_{\sim i}])$ is the 1st-order partial variance of $\theta_{\sim i}$, so that $V(Y) - V_{\theta_{\sim i}} (\mathbb{E}_{\theta_i} [Y|\theta_{\sim i}])$ is the total contribution of all terms that involve θ_i , and S_{Ti} as given in Eq. (28) is the corresponding total-order sensitivity index.

The 1st-order indices S_i , 2nd-order indices S_{ij} , and total-order indices S_{Ti} , as given in Eq. (23), (24), and (28), respectively, are the focus in this study. Because the indices require the computation of high-dimensional integrals, we instead use the estimators developed by Saltelli et al [12, 14]. We briefly summarize the estimators in Appendix C.

III. Results

This section presents the performance of the FNN surrogate model on a randomly sampled test set of new geometry parameters over the input space and compares the results to the Martinez-Sanchez solution. Then, a set of parameter sweeps is presented to show the strengths and limitations of the surrogate over the input space. Finally, the surrogate is used to perform a sensitivity analysis of the output electric field.

A. Surrogate performance on test set

Here, we present the surrogate's performance on the test set. First, $N = 2000$ new geometry parameters were randomly sampled from the input space and the peak electric field magnitude was obtained for each from the electrostatic solver to form the test set $\{\theta^{(j)}, y^{(j)}\}_{j=1}^N$. The parameters were scaled and normalized, and then passed to the trained FNN to obtain forward predictions of peak electric field. The performance of the FNN was evaluated by the relative percent error on each sample j :

$$\% \text{ error} = \frac{|\hat{y}(\theta^{(j)}) - y^{(j)}|}{y^{(j)}} \times 100\% \quad (29)$$

where $y^{(j)}$ is the simulation value of the peak electric field and $\hat{y}(\theta^{(j)})$ is the surrogate prediction of peak electric field. The distribution of percent errors over the test set is shown in Figure 7, and the statistics of the distribution are included in Table 5.

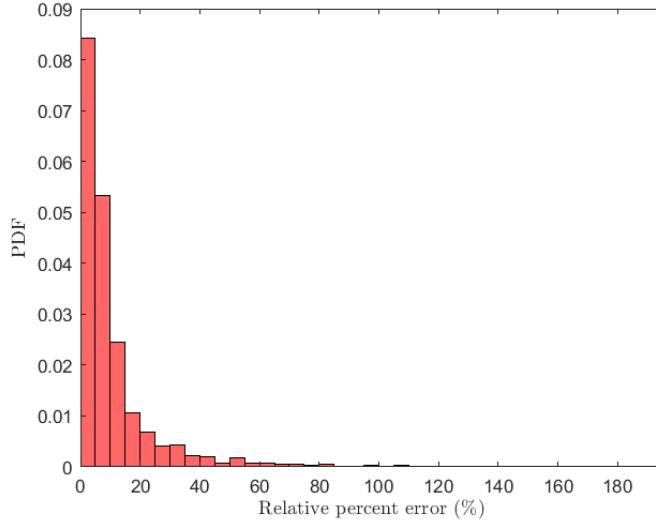


Fig. 7 Performance of the FNN on the test set

Table 5 Statistics of test set relative percent error distribution ($N = 2000$ samples)

Statistic	Mean	Std dev	Min	25%-tile	50%-tile	75%-tile	Max
Rel. % error	11%	16%	0.01%	2.7%	6.2%	12%	195%

While the FNN shows extreme behavior on a few samples from the test set, the majority of samples show good performance at less than 12% error. The large skew in the distribution is evident not only from the histogram in Figure 7, but also from the large discrepancy between the median error of 6.2% and the mean error of 11%. The surrogate's performance over the input space is shown in the 2d scatter plots in Figure 8.

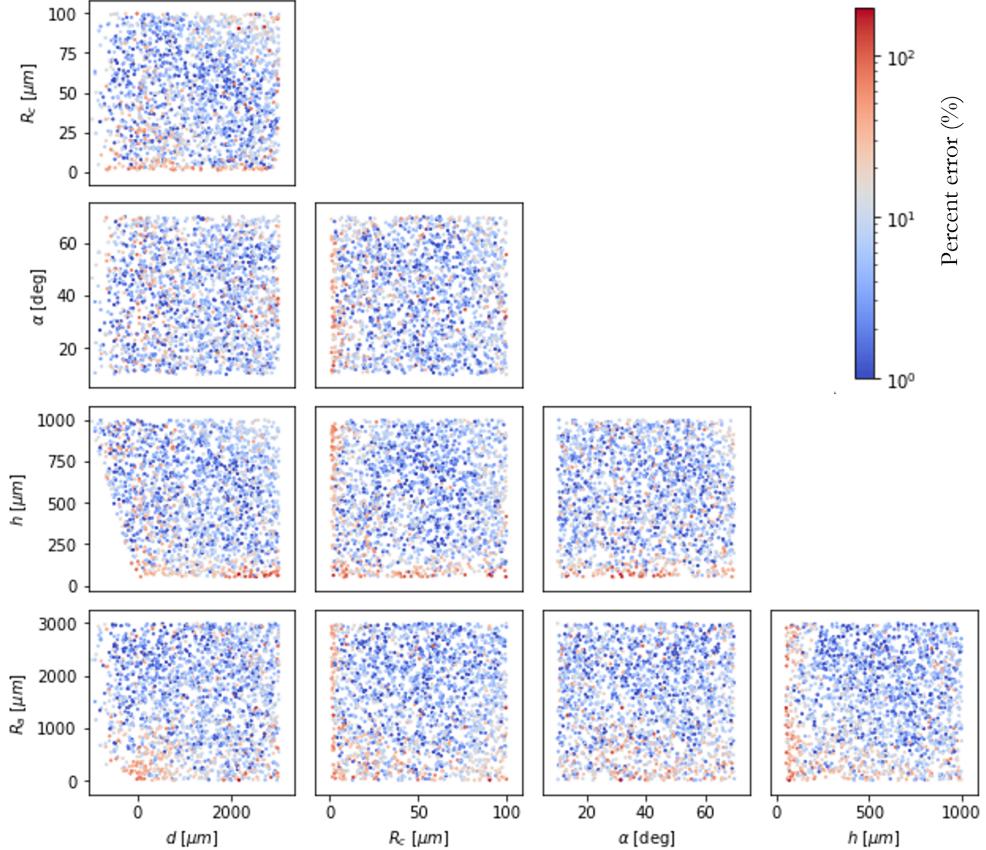


Fig. 8 2d scatter plots of the test set samples over the input space. The relative error of the surrogate on each sample is indicated by color.

We first observe that the majority of high percent error cases reside in the extremes of the input space, namely in regions of small h , R_c , or R_a , and especially in the regime of small emitter height and large tip-to-extractor distance. The rest of the input space is fairly well covered by the surrogate, with percent errors less than 10% in the predominantly blue regions. It is noted that the lack of test samples in the lower-left corners of the (h, d) and (R_a, d) marginals is a result of the practical constraints in Section II.A. To gain insight into the behavior of the FNN in the extremes of the input space, the numeric solver solution is shown in Figure 9 for the geometry with the worst performance of 195% error.

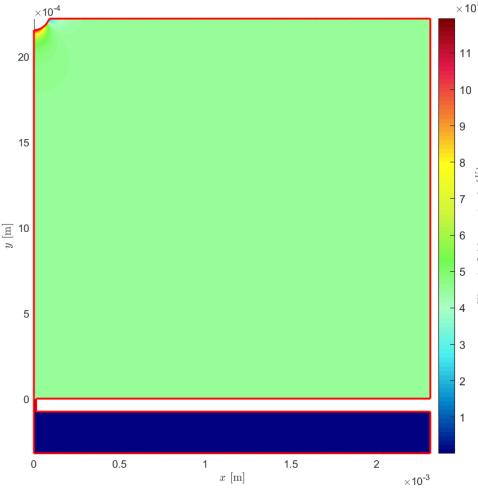


Fig. 9 Electrostatic solution for the 195% error geometry

The geometry parameters for this case are $[h, \alpha, R_c, d, R_a] = [67.5 \mu\text{m}, 31.4 \deg, 90.4 \mu\text{m}, 2154 \mu\text{m}, 13.1 \mu\text{m}]$. A possible explanation for the poor surrogate performance on this case is the absence of training data in this extreme region of the input space. In addition, while a very small emitter that is very far away from an extractor grid with a very small aperture radius is within the constraints of the input space, it is most likely not a desirable design for an emitter.

To further quantify the regions of the input space that exhibited poor surrogate performance, the statistics of the worst 1% of the test set in terms of relative error are shown in Table 6.

Table 6 Statistics of the 1% worst-performing test cases ($N = 20$ samples)

Parameter	Emitter height	Cone half-angle	Radius of curvature	Tip-to-extractor distance	Radius of aperture	Relative error
Unit	[μm]	[deg]	[μm]	[μm]	[μm]	[%]
Mean	202	31.7	42.0	2010	1220	107
Std dev	284	12.4	38.5	1030	1010	29.3
Min	52.5	11.6	1.09	-177	13.1	80.4
25%-tile	58.9	23.2	5.98	1770	160	85.9
50%-tile	69.4	33.3	41.0	2390	1130	99.4
75%-tile	111	39.4	83.5	2720	2280	111
Max	879	55.5	99.9	2980	2750	195

While the results in Table 6 can provide some indication of trends in the input space where the FNN may not perform well, it should be used with caution; trends in one parameter alone do not necessarily hold in the full 5d input space. For example, the 20 samples in Table 6 span nearly the full domain of the radius of curvature ($R_c \in [1, 100] \mu\text{m}$) and the radius of aperture ($R_a \in [10, 3000] \mu\text{m}$); any correlation between these parameters would be hard to determine from Table 6 alone.

However, we can make a few important observations from Table 6. First, the top 1% worst-performing samples from the test set span relative errors from 80.4% to 195%, accounting for the majority of error across the entire test set. Second, these samples are clustered in a region of the input space with small emitter height $h < 100 \mu\text{m}$ and large tip-to-extractor distance $d > 1000 \mu\text{m}$. This region was identified in the scatter plots in Figure 8, and is also the scenario seen in Figure 9. One possible explanation for the poor performance in this regime is that for short emitters that are far away from the extractor grid, the electric field at the tip of the emitter is strongly influenced by proximity to the top of the simulation domain rather than by proximity to the extractor grid; the surrogate may not have learned this distinction in the model response due to lack of adequate training data. While the surrogate fails to characterize this behavior, its

performance may be improved by training on more data in this area of the input space. Alternatively, better sampling efficiency of the edges of the input space could be obtained using other quasi-random sampling methods [15].

B. Comparison of surrogate to Martinez-Sanchez model

To gain an idea of the utility of the surrogate, we now attempt to compare its performance relative to the Martinez-Sanchez (MS) model. The MS model provides an exact, analytical solution for the peak electric field on an infinitely-long, hyperboloid geometry over an infinite flat plate. This geometry is used as an approximation for a finite-length, conical emitter over an extractor grid aperture. The utility of the simulation methodology in this study is that we can obtain accurate electric field predictions outside of the limiting assumptions made by the MS model. Namely, we can account for the presence of an aperture in the extractor grid and predict for finite-length, conical-shaped emitters that may potentially extend through the extractor grid aperture. When a high-fidelity prediction is desired, the electrostatic simulation provides the most accurate results; however, when computational burden is a concern, we hope to approach the accuracy of the numerical simulation with the surrogate but maintain the cheap computational cost of the analytical MS model. In this sense, we seek to quantify the advantage of using the surrogate *in place* of the MS model, such as in direct replacement for electric field prediction in the ESPET code [4].

Since the MS model is undefined for a negative tip-to-extractor distance, all samples from the test set where $d \leq 0$ are removed. The statistics of the distribution of percent errors over the modified test set ($N = 1742$) is provided in Table 7 for both the surrogate and the MS solution. Note that the error in both models is taken with respect to the simulation results as in Eq. (29).

Table 7 Statistics of surrogate and MS distributions of percent error on test set ($N = 1742$ samples)

Statistic	Mean	Std dev	Min	25%-tile	50%-tile	75%-tile	Max
Surrogate	11%	16%	0.01%	2.8%	6.2%	12%	195%
Martinez-Sanchez	254%	658%	1.2%	87%	162%	284%	20,600%

The results in Table 7 clearly outline the advantage of using the surrogate *in place* of the MS model for electric field prediction of conical emitters. While a percent error as high as 195% is certainly not desirable, the overall distribution of errors for the surrogate is much more dense at lower relative errors on the scale of 10% vs. on the order of 100% for the MS solution, which constitutes a significant overall improvement on average over the input space.

C. Parameter sweeps

In this section, the performance of the FNN is evaluated on a set of parameter sweeps, such as might be used in the design of an electrospray emitter. A base emitter design is chosen that was representative of our experiments and then a series of 1d sweeps over each of the geometry parameters is performed, as summarized in Table 8.

Table 8 Emitter geometry parameter sweep summary

Parameter	Symbol	Unit	Base design	Sweep
Emitter height	h	μm	300	[50, 1000]
Cone half-angle	α	deg	30	[10, 70]
Radius of curvature	R_c	μm	30	[1, 100]
Tip-to-extractor distance	d	μm	500	[-280, 3000]
Radius of aperture	R_a	μm	300	[10, 3000]

The results of each parameter sweep are shown in Figure 10. Each plot shows the MS, surrogate, and simulation electric field predictions against the swept parameter. All cases were run at the same bias voltage of $V_0 = 1000$ V. Each parameter sweep was performed at a uniform discretization of $N = 20$ locations across the ranges in Table 8.

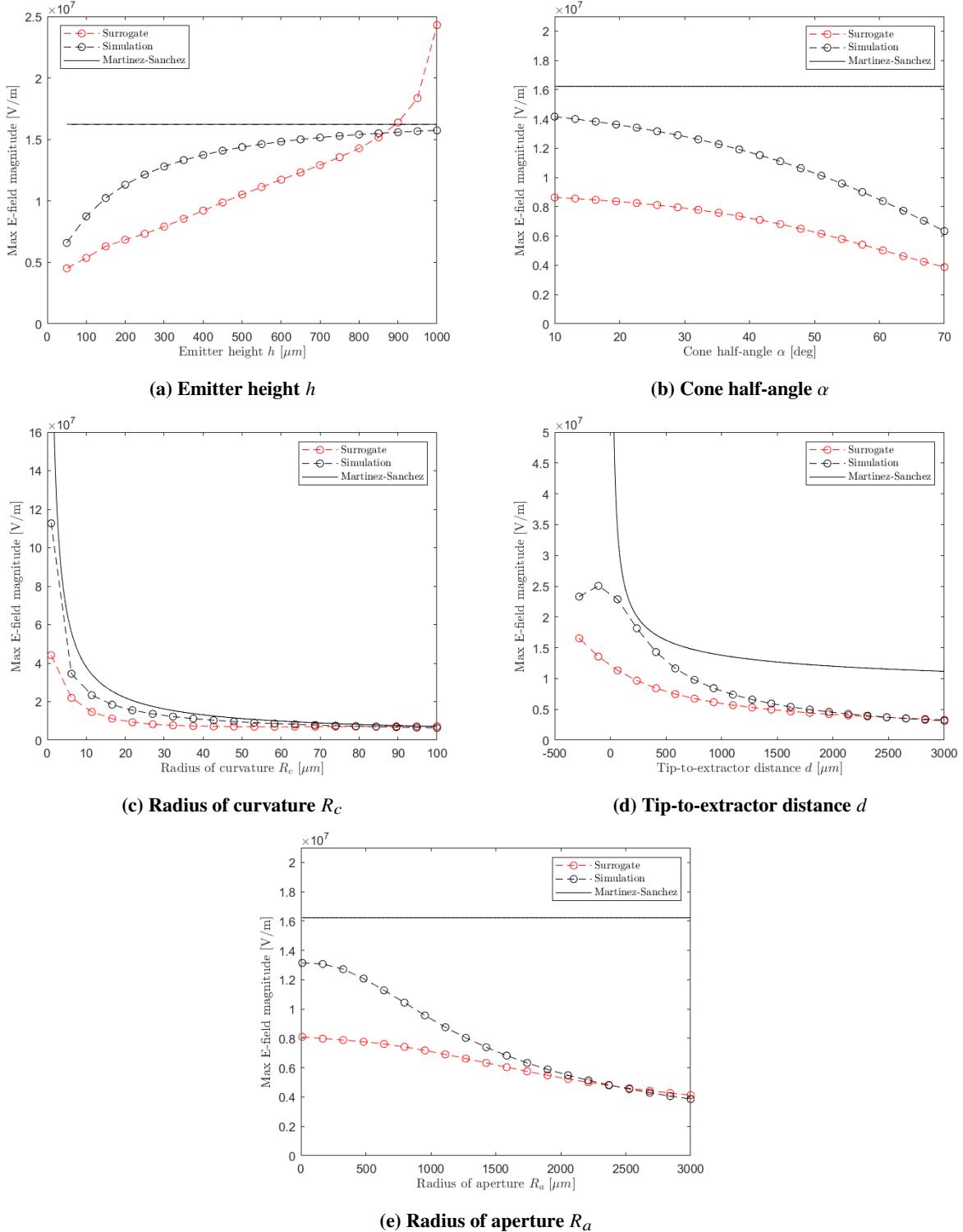


Fig. 10 Results of swept parameter studies in Table 8

Several important observations can be made from the results in Figure 10. First, in all cases, the simulation solution is treated as the most accurate model of electric field prediction for a given set of geometry parameters. It is also noted that the MS solution is constant for cases where R_c and d are constant. Overall, the surrogate is able to demonstrate the

same, expected trends as the simulation: electric field strength increases with emitter height, and decreases for increasing radius of curvature, cone half-angle, tip-to-extractor distance, and radius of aperture. For large tip-to-extractor distance, radius of aperture, and radius of curvature, the surrogate converges to the simulation results. In nearly all cases, the surrogate under-predicts the simulation while the MS solution over-predicts (due to the infinite flat plate assumption).

While the surrogate has demonstrated reasonable performance in matching the overall trends of the simulation results, it fails to capture the asymptotic behavior of Figure 10a and the local maxima behavior of Figure 10d. In addition, for small α , R_c , R_a and large h , the simulation results appear to approach the MS solution, while the surrogate results trail off.

It is important to note that the cases studied here are just arbitrary 1d slices of the large 5d input space in Table 1; the FNN has learned the best regression fit it could to the limited training samples it was provided. The main contribution of the simulation is a high-fidelity model for electric field, without the limiting assumptions of the MS model. The main contribution of the *surrogate* is computational expense on the same order as the analytical MS model, with significantly better performance *on average* across the input space. To improve the performance of the surrogate, more data can be collected in the poor-performing regions of the input space. Alternatively, a more complicated model could be chosen that is more expressive than a simple FNN, or that is better tailored for approximating PDEs, such as a physics-informed neural network [16].

D. Sensitivity analysis

This section presents the application of the surrogate to a global, variance-based sensitivity analysis of the electric field using Sobol' indices, as outlined in Section II.D. First, to meet the independence assumption of the Saltelli method for computing Sobol' indices [12], we restrict the input space of the geometry parameters to remove the practical constraints Eq. (1) and (2). Namely, we enforce the tip-to-extractor distance to be positive to remove constraint (2), and we set the lower bound on the emitter height to $h = \max(h_l)$ to remove constraint (1), (where the maximum lower height bound is $\max(h_l) = R_{c,\max}(1 - \sin \alpha_{\min}) \approx 83 \mu\text{m}$ from Eq. (1)). While this restriction only removes a small portion of the input space from the sensitivity analysis, other estimators could be pursued in future work that account for dependencies between model parameters [17].

After the practical constraints are removed, a Monte Carlo sample size of $N = 100,000$ is used over the restricted $d = 5$ dimensional input space with the Latin hypercube sampler, for a total number of model evaluations of 700,000 required for the 1st-order and total-order index estimators, and a total number of 1,200,000 required for the 2nd-order index estimator. The surrogate is essential to completing this task, since the computational cost is on the order $O(10 \text{ days})$ for the simulation vs. $O(10 \text{ s})$ for the surrogate.

From Eq. (22), it was shown that the total variance of the output $V(Y)$ can be decomposed over a summation of the first, second, and higher-order Sobol' indices. The decomposition of the output variance for the electric field model is summarized in Figure 11 for all of the 1st-order indices and the highest-contributing 2nd-order indices, (expressed as percents), where the notation (θ_i, θ_j) indicates the Sobol' index S_{ij} for the two indicated geometry parameters.

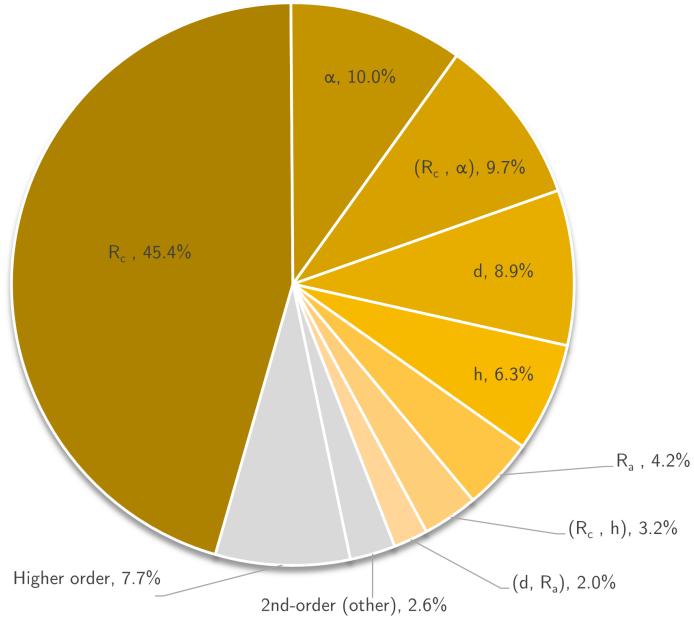


Fig. 11 Sobol' variance decomposition of the maximum electric field due to uncertain design parameters over the input space in Table 1

From this decomposition, it is clear that the electric field model response is most sensitive on average over the input space to the radius of curvature R_c at the emitter tip. In fact, this result is also indicated in Figure 10c, where the electric field strength at the emitter tip grows exponentially as $R_c \rightarrow 0$. Intuitively, changes to the “sharpness” or “bluntness” of the emitter tip should cause significant variation in peak electric field strength. Additionally, changes to the “steepness” of the emitter cone via the cone half-angle α parameter should cause similar variation; indeed, the combination of R_c , α , and (R_c, α) interaction comprises over 65% of the observed variance in the peak electric field. Additionally, higher order effects from interactions between 3 or more parameters account for 7.7% of observed model variance; this implies a high degree of coupling between variables. Future work can utilize this analysis to focus the design of emitter geometry on the most sensitive model parameters.

To further quantify the model sensitivity, the total-order indices for each model parameter are shown in Figure 12 along with the 1st-order indices.

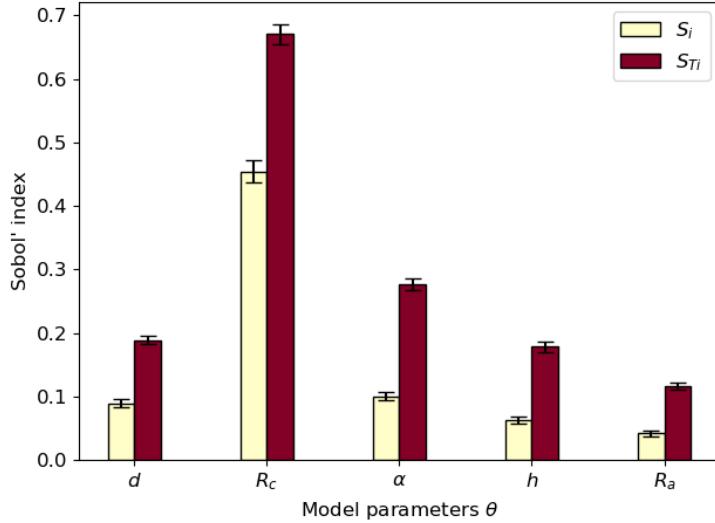


Fig. 12 First S_i and total order S_{Ti} Sobol' indices for the electric field model parameters

The error bars in Figure 12 indicate the 95% Monte Carlo confidence intervals for each index estimate. Figure 12 further indicates the dominance of the R_c and α geometry parameters in contribution to the model output variance, with the total-order sensitivity index of R_c , (which accounts for all first, second, and higher order interactions involving R_c), of 0.67. It is noted that the total-order indices will typically sum to greater than 1, since many interactions are double-counted over all of the parameters.

IV. Conclusion

This paper has presented a sampling, simulation, surrogate, and uncertainty quantification workflow for the design of electrospray emitter geometry. A cone-shaped geometry model was proposed to parameterize the geometry of the emitter and to enable performance studies that benefit the design of arrayed structures. A simulation technique was presented and validated against analytical results for obtaining emitter electric field data. A feedforward neural network was presented as a surrogate model and trained on simulated data to learn the map between emitter geometry and peak electric field strength. The surrogate was shown to perform well on average over the design space when used in place of the analytical Martinez-Sanchez solution for electric field prediction. However, while the surrogate was able to predict general trends, it was shown to under-predict the simulation in almost all cases studied, and it failed to fully capture the output response in several regions of the design space. Future work is needed to identify and train a more robust surrogate model for electric field prediction. In addition, the current surrogate is trained only on peak electric field while the majority of the simulation data is discarded; future directions in this work might target a more complex surrogate to make full-field, physics-informed predictions, with potential benefits to other aspects of emitter design. Techniques from models such as Bayesian networks could also be applied to provide the surrogate with some level of ability to quantify uncertainty in its own predictions, especially in regions outside of the input space.

Finally, this paper was enabled by the computational advantage of the surrogate to perform a global sensitivity analysis of peak electric field. Overall, the electric field model was found to be most sensitive to the radius of curvature at the tip of the emitter, and to a lesser extent the emitter cone half-angle. Future work would expand the sensitivity analysis to regions of the design space where practical constraints are imposed; results from the sensitivity analysis can be used to guide future design of emitters for electrospray applications.

Appendix

A. Empirical PDF fits of training data

The empirical PDF fits to the exponential PDF (Eq. (15)) for each of the scaled R_c , d , R_a , and E_{\max} parameters are shown overlaid on the 1d marginals of the training data in Figure 13.

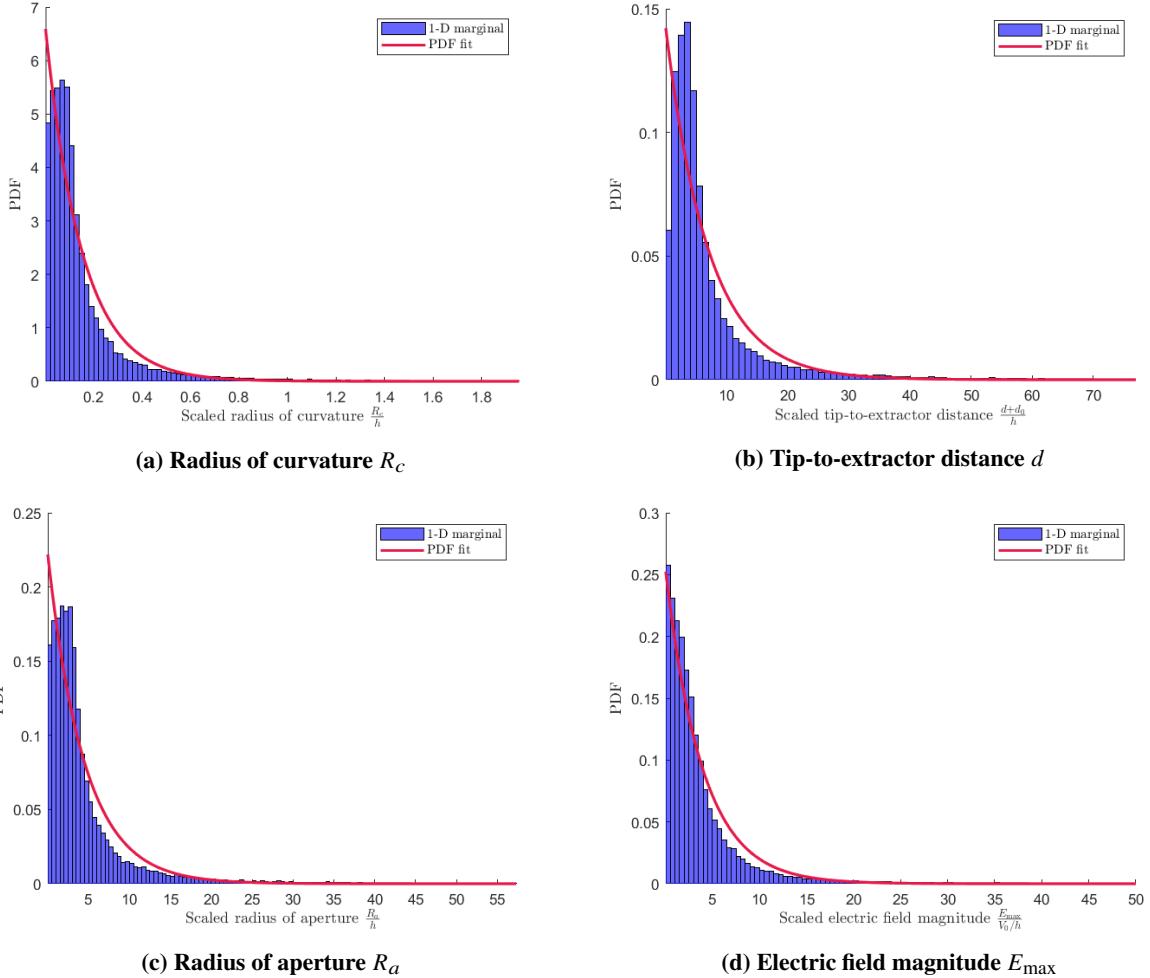


Fig. 13 Empirical exponential PDF fits on the training data for the R_c , d , R_a , and E_{\max} parameters

Table 9 provides the fit coefficient λ for each parameter.

Table 9 Empirical fit coefficients

Parameter	Symbol	Scaling	Fit coefficient λ in Eq. (15)
Radius of curvature	R_c	$\frac{R_c}{h}$	6.64
Tip-to-extractor distance	d	$\frac{d+d_0}{h}$	0.142
Radius of aperture	R_a	$\frac{R_a}{h}$	0.223
Electric field magnitude	E_{\max}	$\frac{E_{\max}}{V_0/h}$	0.255

Figure 14 shows the resulting distributions of each parameter after applying the CDF transform $g(\cdot)$ with the fit coefficients in Table 9.

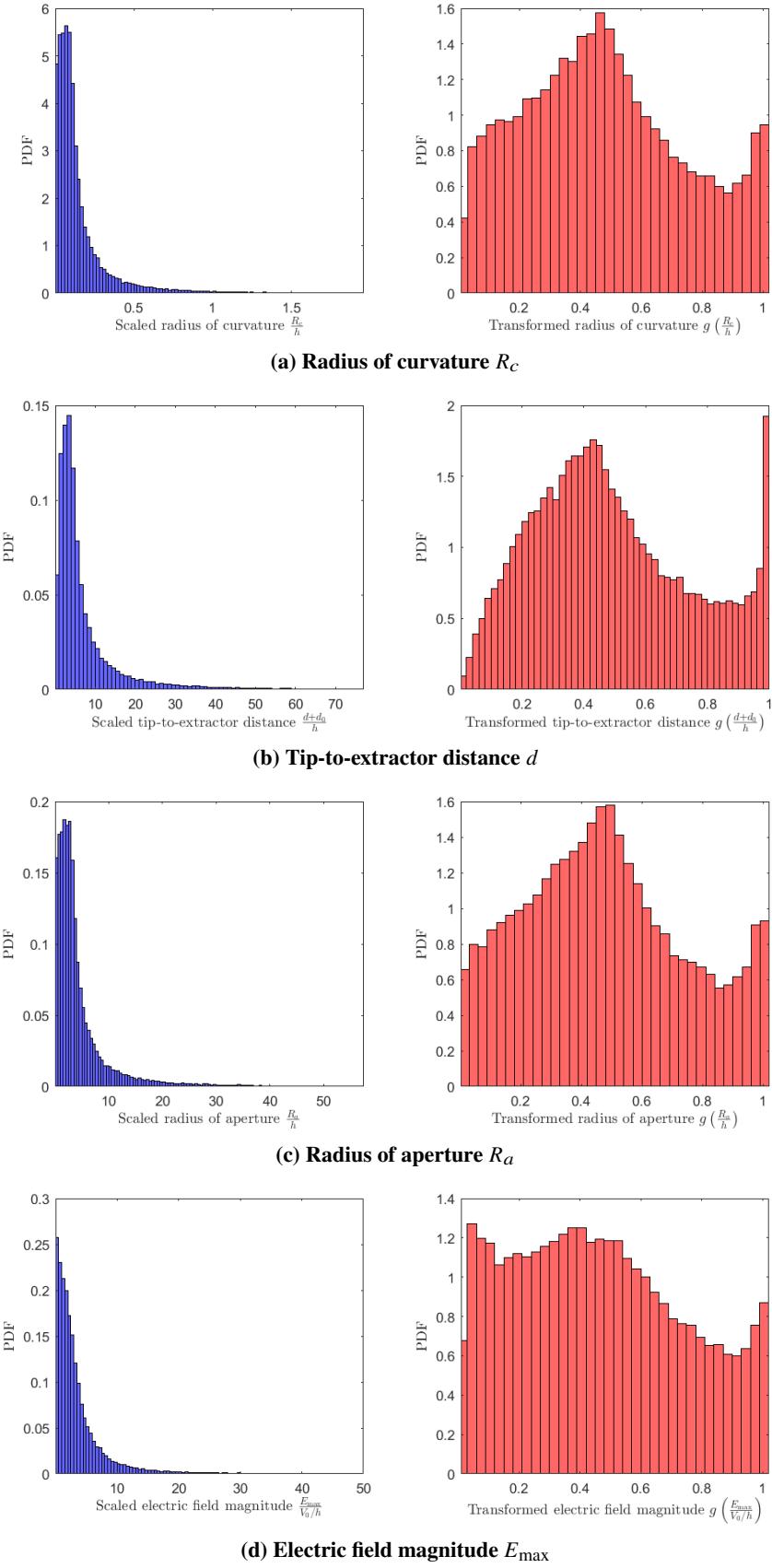


Fig. 14 1d marginals of the R_c , d , R_a , and E_{\max} parameters before and after the CDF transform

Since none of the empirical distributions perfectly follow an exponential distribution, the CDF transform does not result in a perfectly uniform output distribution; however, the transformation worked good enough for the purposes of this study in reducing the skewness of the input parameter distributions. The convenience of using the exponential distribution was the availability of an analytical CDF transform. Alternatively, one could have constructed an empirical CDF to better fit the data and to produce a more uniform random variable from the transform.

B. Feedforward neural network details

The feedforward neural network (FNN) surrogate model has four network inputs, two fully-connected hidden layers, and one regression output, as shown in Figure 15. Here, we include a brief summary of how network outputs are computed for an FNN.

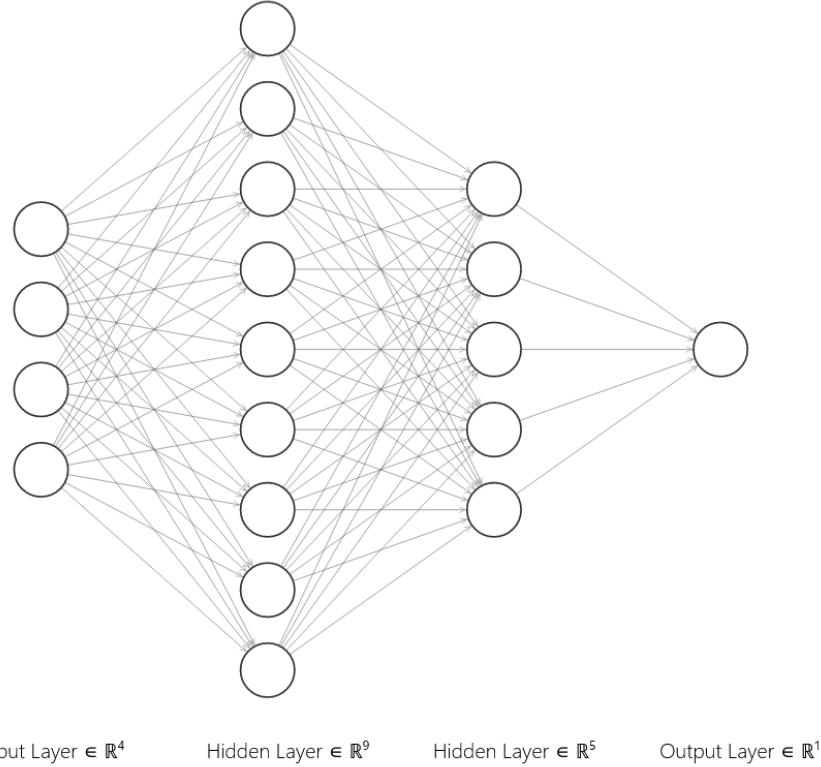


Fig. 15 FNN surrogate model for electric field prediction

The activation of the first node of the first hidden layer is computed as the weighted sum of the 4 feature inputs θ_i plus a scalar bias term b_1 :

$$a_1 = w_1\theta_1 + w_2\theta_2 + w_3\theta_3 + w_4\theta_4 + b_1 \quad (30)$$

where we have dropped the scaling and normalization notation for convenience, (i.e. $[\theta_1, \theta_2, \theta_3, \theta_4] = [\tilde{\alpha}_n, \tilde{R}_{c,n}, \tilde{d}_n, \tilde{R}_{a,n}]$). Likewise, the activation of node j in the first hidden layer is

$$a_j = \sum_{i=1}^N w_{i,j}\theta_i + b_j \quad (31)$$

where $i = 1 \dots N$ are the feature inputs and $w_{i,j}$ is the weight from feature input i to node j in the first hidden layer. The

activation of all N_1 nodes in the first hidden layer can then be written as the matrix product

$$\begin{bmatrix} a_1 \\ \vdots \\ a_j \\ \vdots \\ a_{N_1} \end{bmatrix} = \begin{bmatrix} w_{1,1} & w_{2,1} & w_{3,1} & w_{4,1} \\ \vdots & \vdots & \vdots & \vdots \\ w_{1,j} & w_{2,j} & w_{3,j} & w_{4,j} \\ \vdots & \vdots & \vdots & \vdots \\ w_{1,N_1} & \dots & \dots & w_{4,N_1} \end{bmatrix} \begin{bmatrix} \theta_1 \\ \theta_2 \\ \theta_3 \\ \theta_4 \end{bmatrix} + \begin{bmatrix} b_1 \\ \vdots \\ b_j \\ \vdots \\ b_{N_1} \end{bmatrix} \quad (32)$$

More generally, the activation of all N_k nodes in the k^{th} layer is given by

$$\begin{bmatrix} a_{1,k} \\ \vdots \\ a_{j,k} \\ \vdots \\ a_{N_k,k} \end{bmatrix} = \begin{bmatrix} w_{1,1} & w_{2,1} & \dots & w_{N_{k-1},1} \\ \vdots & \vdots & & \vdots \\ w_{1,j} & w_{2,j} & \dots & w_{N_{k-1},j} \\ \vdots & \vdots & & \vdots \\ w_{1,N_k} & \dots & \dots & w_{N_{k-1},N_k} \end{bmatrix} \begin{bmatrix} a_{1,k-1} \\ a_{2,k-1} \\ \vdots \\ a_{N_{k-1},k-1} \end{bmatrix} + \begin{bmatrix} b_{1,k} \\ \vdots \\ b_{j,k} \\ \vdots \\ b_{N_k,k} \end{bmatrix} \quad (33)$$

where $a_{j,k-1}$ is the activation of the j^{th} node in the previous layer. This matrix product is typically written in shorthand as

$$\mathbf{a}_k = \mathbf{W}_k \mathbf{a}_{k-1} + \mathbf{b}_k \quad (34)$$

where \mathbf{W}_k is the weight matrix of the connections to the k^{th} layer, \mathbf{b}_k is the bias vector of the k^{th} layer, and \mathbf{a}_k is the node activations of the k^{th} layer. After evaluating Eq. (34), a nonlinear function σ is applied element-wise:

$$\mathbf{a}_k = \sigma(\mathbf{W}_k \mathbf{a}_{k-1} + \mathbf{b}_k) \quad (35)$$

to produce the final node activations of the k^{th} layer. Each additional network layer takes the previous activations \mathbf{a}_k as an input and applies a new weight matrix and bias vector. From a layer with N_{k-1} nodes to a layer with N_k nodes, the total number of additional trainable parameters p_k (weights and biases) is given by:

$$p_k = N_k N_{k-1} + N_k \quad (36)$$

For an FNN with h hidden layers, the total number of trainable parameters is

$$p_T = \sum_{k=1}^{h+1} p_k \quad (37)$$

where $N_0 = N_\theta$ is the number of network feature inputs and $N_{h+1} = N_y$ is the number of network regression outputs. For an FNN with h hidden layers, N_θ input features, and N_y outputs, the output $\hat{y} \in \mathbb{R}^{N_y}$ after a full forward pass of a given network input $\theta \in \mathbb{R}^{N_\theta}$ is given by

$$\hat{y}(\theta) = \mathbf{W}_{h+1} \mathbf{a}_h + \mathbf{b}_{h+1}, \quad \text{where} \quad (38)$$

$$\mathbf{a}_h = K_{k=1}^h [\sigma(\mathbf{W}_k \mathbf{a}_{k-1} + \mathbf{b}_k)] \quad (39)$$

where the $K_{k=1}^h [\cdot]$ notation implies repeated function composition of the quantity in the brackets. It can be seen that there is a pure linear mapping \mathbf{W}_{h+1} from the last hidden layer to the regression output (i.e. no nonlinear function $\sigma(\cdot)$ is applied). It is also noted that the \mathbf{a}_0 activations are the network inputs (i.e. $\mathbf{a}_0 = \theta$). To coordinate with the normalization procedure in this study, the hyperbolic tangent function is used as the nonlinear function in each layer, which maps inputs x to the range $[-1, 1]$:

$$\sigma(\cdot) = \tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \quad (40)$$

The Adam optimizer is a popular gradient descent method for minimizing the MSE for neural networks [18]. The Adam optimizer is in the class of stochastic gradient descent (SGD) algorithms, which use one small subset (batch) of

the full training set at a time to compute gradients and update parameters. SGD typically converges faster than if the full dataset was used for each gradient step. In this study, the Adam optimizer was used with L_2 regularization of the network weights and a batch size of 2048. The optimizer is provided by default in the Matlab environment.

In order to obtain a dimensional electric field prediction from the fully-trained FNN, one can use the following procedure:

- 1) Choose a set of geometry parameters $\theta = [h, \alpha, R_c, d, R_a]$ from the bounds in Table 1 that meets the constraints of Eq. (1) and (2).
- 2) Scale the parameters according to Table 3. Use the CDF transform (Eq. (16)) for the function $g(\cdot)$ with the fit coefficients in Table 9.
- 3) Normalize the scaled parameters $\tilde{\theta} = [\tilde{\alpha}, \tilde{R}_c, \tilde{d}, \tilde{R}_a]$ with Eq. (17), where the range $[\tilde{\theta}_{\min}, \tilde{\theta}_{\max}]$ is obtained from the training data set.
- 4) Perform a forward pass of the normalized parameters $\tilde{\theta}_n$ through the network via Eq. (38) using the learned network parameters p , to obtain $\tilde{E}_{\max,n}$.
- 5) Undo the normalization from $\tilde{E}_{\max,n}$ to \tilde{E}_{\max} via Eq. (17), where the range $[\tilde{E}_{\max,min}, \tilde{E}_{\max,max}]$ is once again obtained from the training data set.
- 6) Apply the inverse CDF function: $g^{-1}(x) = -\frac{1}{\lambda} \ln(1-x)$ using the fit coefficient in Table 9 for E_{\max} .
- 7) Finally, rescale by a desired bias voltage V_0 and the emitter height: $E_{\max} = g^{-1}(\tilde{E}_{\max}) \cdot \frac{V_0}{h}$

C. Estimators for Sobol' indices

Here, we briefly summarize estimators for the 1st-order S_i , 2nd-order S_{ij} , and total-order S_{Ti} Sobol' sensitivity indices [12, 14]. First, two sets of N samples of the parameters are obtained, forming the sampling matrices A and B :

$$A = \begin{bmatrix} \theta_{1,1} & \theta_{1,2} & \dots & \theta_{1,d} \\ \theta_{2,1} & \theta_{2,2} & \dots & \theta_{2,d} \\ \vdots & \vdots & \ddots & \vdots \\ \theta_{N,1} & \theta_{N,2} & \dots & \theta_{N,d} \end{bmatrix} \quad (41)$$

$$B = \begin{bmatrix} \theta'_{1,1} & \theta'_{1,2} & \dots & \theta'_{1,d} \\ \theta'_{2,1} & \theta'_{2,2} & \dots & \theta'_{2,d} \\ \vdots & \vdots & \ddots & \vdots \\ \theta'_{N,1} & \theta'_{N,2} & \dots & \theta'_{N,d} \end{bmatrix} \quad (42)$$

where a single row $(A)_r$ or $(B)_r$ is a single realization of the d model parameters. Next, we form two sets of d cross-sampled matrices:

$$A_B^{(k)} = \begin{bmatrix} \theta_{1,1} & \theta_{1,2} & \dots & \theta'_{1,k} & \dots & \theta_{1,d} \\ \theta_{2,1} & \theta_{2,2} & \dots & \theta'_{2,k} & \dots & \theta_{2,d} \\ \vdots & \vdots & \dots & \vdots & \ddots & \vdots \\ \theta_{N,1} & \theta_{N,2} & \dots & \theta'_{N,k} & \dots & \theta_{N,d} \end{bmatrix} \quad (43)$$

$$B_A^{(k)} = \begin{bmatrix} \theta'_{1,1} & \theta'_{1,2} & \dots & \theta_{1,k} & \dots & \theta'_{1,d} \\ \theta'_{2,1} & \theta'_{2,2} & \dots & \theta_{2,k} & \dots & \theta'_{2,d} \\ \vdots & \vdots & \dots & \vdots & \ddots & \vdots \\ \theta'_{N,1} & \theta'_{N,2} & \dots & \theta_{N,k} & \dots & \theta'_{N,d} \end{bmatrix} \quad (44)$$

for $k = 1 \dots d$, where we swap the k 'th column of A and B at each iteration of k . Then, the first and total order sensitivity index estimators for parameter θ_i are given by

$$\hat{S}_i = \frac{\frac{1}{N} \sum_{r=1}^N f(B)_r \left(f(A_B^{(i)})_r - f(A)_r \right)}{V(Y)}, \quad (45)$$

$$\hat{S}_{Ti} = \frac{\frac{1}{2N} \sum_{r=1}^N \left(f(A)_r - f(A_B^{(i)})_r \right)^2}{V(Y)} \quad (46)$$

where $A_B^{(i)}$ is the i 'th cross-sampled matrix in the set $A_B^{(k)}$, and we evaluate the scalar function $f(\cdot)$ with the model parameters in row r of the indicated matrices. The 2nd-order sensitivity index estimator between parameters θ_i and θ_j is given by

$$\hat{S}_{ij} = \frac{\frac{1}{N} \sum_{r=1}^N \left(f(B_A^{(i)})_r \cdot f(A_B^{(j)})_r - f(A)_r \cdot f(B)_r \right)}{V(Y)} - \hat{S}_i - \hat{S}_j \quad (47)$$

where \hat{S}_i and \hat{S}_j are estimated from Eq. (45). We additionally estimate the total output variance $V(Y)$ by the sample variance over all model evaluations in $f(A)$ and $f(B)$. For a set of d parameters and N samples, the total number of model evaluations required for the \hat{S}_i and \hat{S}_{Ti} estimators is $N(d + 2)$, while the total number is $N(2d + 2)$ for the \hat{S}_{ij} estimator [12, 19].

Acknowledgements

This work was supported by an Early Stage Innovations grant from NASA's Space Technology Research Grants Program. The authors would also like to thank Dr. John Yim of the NASA Glenn Research Center for his technical guidance.

References

- [1] Dankanich, J., and Lozano, P., "Dual Mode Green Propulsion for Revolutionary Performance Gains with Minimal Recurring Investments," *Planetary Science Vision 2050 Workshop*, Vol. 1989, 2017, p. 8155.
- [2] Whittaker, C. B., Gorodetsky, A., and Jorns, B., "Quantifying Uncertainty in the Scaling Laws of Porous Electrospray Emitters," *AIAA Propulsion and Energy 2020 Forum*, 2020, p. 3615.
- [3] St Peter, B., Dressler, R. A., Chiu, Y.-h., and Fedkiw, T., "Electrospray Propulsion Engineering Toolkit (ESPET)," *Aerospace*, Vol. 7, No. 7, 2020, p. 91.
- [4] Dressler, R. A., and St Peter, B., "Electrospray Propulsion Engineering Toolkit, ESPET 1.0," , 2018.
- [5] Gorodetsky, A., Whittaker, C. B., Szulman, A., and Jorns, B., "Robust Design of Electrospray Emitters," *AIAA Propulsion and Energy 2021 Forum*, 2021, p. 3422.
- [6] Jorns, B. A., Gorodetsky, A., Lasky, I., Kimber, A., Dahl, P., St. Peter, B., and Dressler, R., "Uncertainty Quantification of Electrospray Thruster Array Lifetime," *Proceedings of the 36th International Electric Propulsion Conference, Vienna, Austria*, 2019, pp. 15–20.
- [7] Whittaker, C. B., Gorodetsky, A., and Jorns, B. A., "Model Inference from Electrospray Thruster Array Tests," *AIAA SciTech 2022 Forum*, 2022, p. 0041.
- [8] Martinez-Sanchez, M., "Lecture 23-25: Colloidal Engines," , 2004.
- [9] Natisin, M., Zamora, H., McGehee, W., Arnold, N., Holley, Z., Holmes, M., and Eckhardt, D., "Fabrication and characterization of a fully conventionally machined, high-performance porous-media electrospray thruster," *Journal of Micromechanics and Microengineering*, Vol. 30, No. 11, 2020, p. 115021.
- [10] McKay, M. D., Beckman, R. J., and Conover, W. J., "A Comparison of Three Methods for Selecting Values of Input Variables in the Analysis of Output from a Computer Code," *Technometrics*, Vol. 21, No. 2, 1979, pp. 239–245.
- [11] Sobol, I. M., "Global sensitivity indices for nonlinear mathematical models and their Monte Carlo estimates," *Mathematics and computers in simulation*, Vol. 55, No. 1-3, 2001, pp. 271–280.

- [12] Saltelli, A., Annoni, P., Azzini, I., Campolongo, F., Ratto, M., and Tarantola, S., “Variance based sensitivity analysis of model output. Design and estimator for the total sensitivity index,” *Computer physics communications*, Vol. 181, No. 2, 2010, pp. 259–270.
- [13] Sobol, I. M., “Sensitivity analysis for non-linear mathematical models,” *Mathematical modelling and computational experiment*, Vol. 1, 1993, pp. 407–414.
- [14] Saltelli, A., “Making best use of model evaluations to compute sensitivity indices,” *Computer physics communications*, Vol. 145, No. 2, 2002, pp. 280–297.
- [15] Sobol’, I. M., “On the distribution of points in a cube and the approximate evaluation of integrals,” *Zhurnal Vychislitel’noi Matematiki i Matematicheskoi Fiziki*, Vol. 7, No. 4, 1967, pp. 784–802.
- [16] Raissi, M., Perdikaris, P., and Karniadakis, G. E., “Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations,” *Journal of Computational physics*, Vol. 378, 2019, pp. 686–707.
- [17] Da Veiga, S., Wahl, F., and Gamboa, F., “Local polynomial estimation for sensitivity analysis on models with correlated inputs,” *Technometrics*, Vol. 51, No. 4, 2009, pp. 452–463.
- [18] Kingma, D. P., and Ba, J., “Adam: A method for stochastic optimization,” *arXiv preprint arXiv:1412.6980*, 2014.
- [19] Herman, J., and Usher, W., “SALib: An open-source Python library for Sensitivity Analysis,” *The Journal of Open Source Software*, Vol. 2, No. 9, 2017. <https://doi.org/10.21105/joss.00097>.