Uncertainty Quantification for the DESC Stellarator System Using Optimized Perturbation Method

Yulong Liang
Georgia Institute of Technology
Atlanta, Georgia, USA
yliang329@gatech.edu

Yicong Fu
Georgia Institute of Technology
Atlanta, Georgia, USA
aaronfu@gatech.edu

1 ABSTRACT

In this project, we aim to improve the robustness of stellarator design and optimization by quantifying forward uncertainties. To achieve this, we will conduct numerical experiments using the DESC code, which is capable of solving three-dimensional magnetohydrodynamics equilibrium equations. The discretized governing equations will be solved using the spectral method, and perturbation method will be applied to incrementally solve for the plasma state of a complex geometry configuration by sequentially perturbing from a standard simple geometry. The accuracy of the solution can be improved by defining different orders of perturbation.

To quantify the forward uncertainties, several approaches will be proposed, including Gaussian process, polynomial chaos approximation, and reduced basis approximation. The effectiveness and accuracy of these approaches will be compared and analyzed to identify the optimal method. Based on our study, we present the final results which demonstrate and compare how different approaches produce perturbed distributions for pressure profile and rotational transform profile, and their corresponding output distribution for force eroor balance value $|\mathbf{F}|$. These distributions were analyzed to evaluate the performance of each method.

2 INTRODUCTION

Nuclear reaction is considered the ultimate means of generating power for mankind. In essence, nuclear reaction includes the process of fusion of ultra-high temperature plasma that contains ionized high-energy particles which collapse into each other under specific conditions and initiate the chain reaction that subsequently releases the significant amount of energy. This work focuses on magnetic confinement fusion, which is one of the two major types of fusion reaction. Figure 1 illustrates the Wendelstein 7-X stellarator, which is a typical plasma device for magnetic confinement. High speed and temperature plasma travels along the yellow tunnel, such motion is confined by external magnets in blue.

One of the major challenges of designing a stellarator is atom drifting, which is the behavior of plasma material deviating from the expected orbit due to insufficient confinement. Such drifting can cause the waste of fusion material and lead to premature termination. In theory, such drifting phenomenon is often minimized by numerical optimization utilizing computer simulation code, such as DESC [DK20], which is a steady-state code for solving the three dimensional magnetohydrodynamics equilibrium. However, the real operational condition may deviate from ideal design state due to uncertainties introduced by imperfect geometry, manufacturing defect, material imperfection, etc. In this work, we propose to use statistical methods to quantify such uncertainties in order to provide a better understanding of the robustness of stellarator design.

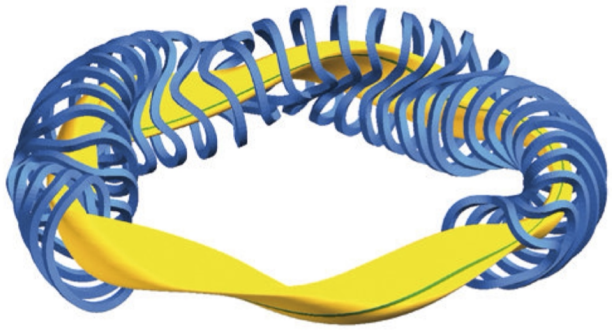


Figure 1: Illustration of Wendelstein 7-X Stellarator

3 LITERATURE REVIEW

3.1 Quick and Accurate Equilibria

One important feature for stellarator design and operation is its 3D equilibrium codes, which requires high-accuracy equilibria for stability studies. In DESC coding system, we aim to find equilibria by minimizing the MHD force balance error in real space directly. The accuracy for generated results can be determined by final plasma energy and satisfaction of MHD force balance equation. Here we show the computational step for the plasma equilibrium state with the desired geometry. The ideal MHD equilibrium model is determined by three main equations [PC22]. The Ampere's Law

$$\nabla \times \mathbf{B} = \mu_0 \mathbf{J} \tag{1}$$

The Gauss's Law

$$\nabla \cdot \mathbf{B} = 0 \tag{2}$$

and momentum density conservation equation

$$\mathbf{J} \times \mathbf{B} = \nabla p \tag{3}$$

where $\mathbf{B}$ is the magnetic field, $\mathbf{J}$ is the current density, p is the scalar pressure, and μ_0 is the permeability of free space. These equations combined together to form the force error balance equation for plasma

$$\mathbf{F} = \mathbf{J} \times \mathbf{B} - \nabla p = 0 \tag{4}$$

and the stationary state for plasma potential energy will be

$$W = \int_V \left(\frac{B^2}{2\mu_0} + \frac{p}{\gamma - 1} dV \right) \tag{5}$$

where V is the plasma volume and γ is the adiabatic index. This is the main equilibrium system for plasma in stellarator model and we will describe how we solve it using current perturbation method in the next subsection.

3.2 Perturbations

From the equilibria we calculated above, we can define a set of parameters for the general fixed-boundary equilibrium problem $\mathbf{c} = \{R_b, Z_b, p.\iota, \Psi\}$ where R_b, Z_b are the R, Z coordinates of the boundary surface, p is the pressure profile, ι the rotational transform which is defined by $\frac{d\Psi_P/d\psi}{d\Psi_T/d\psi}$, and Ψ the total toroidal flux through the torus. We can set $\mathbf{c}$ as a input parameter set and form the nonlinear algebraic equation for the condition of MHD equilibrium

$$\mathbf{f}(\mathbf{x}, \mathbf{c}) = 0 \tag{6}$$

where function $\mathbf{f}$ is the discretized form of the general MHD force balance equation. Here the output parameter set $\mathbf{x}$ consist of three independent variables $\mathbf{x} = [R_{lmn}, Z_{lmn}, \lambda_{lmn}]$ where R_{lmn}, Z_{lmn} are spectral coefficients of the flux surface coordinate and λ_{lmn} is the poloidal stream function. This is our general UQ model for the project.

With an input parameter set $\mathbf{c}$ and output parameter set $\mathbf{x}$ that satisfy $\mathbf{f}(\mathbf{x}, \mathbf{c}) = 0$, we wish to know how the equilibrium and generated output results would change if $\mathbf{c}$ change with a small perturbation $\mathbf{c} + \Delta\mathbf{c}$.

$$\mathbf{f}(\mathbf{x} + \Delta\mathbf{x}, \mathbf{c} + \Delta\mathbf{c}) = 0 \tag{7}$$

Here the research paper from Princeton Plasma Control Group used Taylor Series Approximation to solve the perturbed equation [CD22]. First we expand the equation through Taylor Series with different order for Δx and Δc

$$\begin{aligned}\mathbf{f}(\mathbf{x} + \Delta\mathbf{x}, \mathbf{c} + \Delta\mathbf{c}) = \mathbf{f}(\mathbf{x}, \mathbf{c}) &+ \frac{\partial \mathbf{f}}{\partial \mathbf{x}}\Delta\mathbf{x} + \frac{\partial \mathbf{f}}{\partial \mathbf{c}}\Delta\mathbf{c} + \frac{1}{2}\frac{\partial^2 \mathbf{f}}{\partial \mathbf{x}^2}\Delta\mathbf{x}^2 \\ &+ \frac{1}{2}\frac{\partial^2 \mathbf{f}}{\partial \mathbf{c}^2}\Delta\mathbf{c}^2 + \frac{\partial^2 \mathbf{f}}{\partial \mathbf{x}\partial \mathbf{c}}\Delta\mathbf{x}\Delta\mathbf{c} + ...\end{aligned} \tag{8}$$

Then we further expand Δx and Δc through a perturbation series with a parameter ϵ. Here since we assume Δc is a known parameter, we can expand it with only the first order term.

$$\Delta\mathbf{x} = \epsilon\mathbf{x}_1 + \epsilon^2\mathbf{x}_2 + \epsilon^3\mathbf{x}_3... \tag{9}$$

$$\Delta\mathbf{c} = \epsilon\mathbf{c}_1 \tag{10}$$

Substitute these two perturbation series to Equation (8) and we can get

$$\begin{aligned}0 = &\frac{\partial \mathbf{f}}{\partial \mathbf{x}}(\epsilon\mathbf{x}_1 + \epsilon^2\mathbf{x}_2 + \epsilon^3\mathbf{x}_3) + \frac{\partial \mathbf{f}}{\partial \mathbf{c}}\epsilon\mathbf{c}_1 \\ &+ \frac{1}{2}\frac{\partial^2 \mathbf{f}}{\partial \mathbf{x}^2}(\epsilon\mathbf{x}_1 + \epsilon^2\mathbf{x}_2 + \epsilon^3\mathbf{x}_3)^2 + \frac{1}{2}\frac{\partial^2 \mathbf{f}}{\partial \mathbf{c}^2}(\epsilon\mathbf{c}_1)^2 \\ &+ \frac{\partial^2 \mathbf{f}}{\partial \mathbf{x}\partial \mathbf{c}}(\epsilon\mathbf{x}_1 + \epsilon^2\mathbf{x}_2 + \epsilon^3\mathbf{x}_3)\epsilon\mathbf{c}_1 \\ &+ \frac{1}{6}\frac{\partial^3 \mathbf{f}}{\partial \mathbf{x}^3}(\epsilon\mathbf{x}_1 + \epsilon^2\mathbf{x}_2 + \epsilon^3\mathbf{x}_3)^3 + \frac{1}{6}\frac{\partial^3 \mathbf{f}}{\partial \mathbf{c}^3}(\epsilon\mathbf{c}_1)^3 \\ &+ \frac{\partial^3 \mathbf{f}}{\partial \mathbf{x}^2\partial \mathbf{c}}(\epsilon\mathbf{x}_1 + \epsilon^2\mathbf{x}_2 + \epsilon^3\mathbf{x}_3)^2(\epsilon\mathbf{c}_1) \\ &+ \frac{\partial^3 \mathbf{f}}{\partial \mathbf{x}\partial \mathbf{c}^2}(\epsilon\mathbf{x}_1 + \epsilon^2\mathbf{x}_2 + \epsilon^3\mathbf{x}_3)(\epsilon\mathbf{c}_1)^2 + h.o.t.\end{aligned} \tag{11}$$

In order to have a more clear view for the relation of $\mathbf{x}$ between different orders, we collect powers of ϵ and set each order to zero. In this way we can get the first order equation

$$0 = \frac{\partial \mathbf{f}}{\partial \mathbf{x}}\epsilon\mathbf{x}_1 + \frac{\partial \mathbf{f}}{\partial \mathbf{c}}\epsilon\mathbf{c}_1 \tag{12}$$

$$\mathbf{x}_1 = -\left(\frac{\partial \mathbf{f}}{\partial \mathbf{x}}\right)^{-1}\left(\frac{\partial \mathbf{f}}{\partial \mathbf{c}}c_1\right) \tag{13}$$

second order equation

$$0 = \frac{\partial \mathbf{f}}{\partial \mathbf{x}}\epsilon^2\mathbf{x}_2 + \frac{1}{2}\frac{\partial^2 \mathbf{f}}{\partial \mathbf{x}^2}(\epsilon\mathbf{x}_1)^2 + \frac{1}{2}\frac{\partial^2 \mathbf{f}}{\partial \mathbf{c}^2}(\epsilon\mathbf{c}_1)^2 + \frac{\partial^2 \mathbf{f}}{\partial \mathbf{x}\partial \mathbf{c}}(\epsilon^2\mathbf{x}_1\mathbf{c}_1) \tag{14}$$

$$\mathbf{x}_2 = -\left(\frac{\partial \mathbf{f}}{\partial \mathbf{x}}\right)^{-1}\left(\frac{1}{2}\frac{\partial^2 \mathbf{f}}{\partial \mathbf{x}^2}\mathbf{x}_1^2 + \frac{1}{2}\frac{\partial^2 \mathbf{f}}{\partial \mathbf{c}^2}\mathbf{c}_1^2 + \frac{\partial^2 \mathbf{f}}{\partial \mathbf{x}\partial \mathbf{c}}(\mathbf{x}_1\mathbf{c}_1)\right) \tag{15}$$

and third order equation

$$\begin{aligned}0 = &\frac{\partial \mathbf{f}}{\partial \mathbf{x}}\epsilon^3\mathbf{x}_3 + \frac{\partial^2 \mathbf{f}}{\partial \mathbf{x}^2}(\epsilon^3\mathbf{x}_1\mathbf{x}_2) + \frac{\partial^2 \mathbf{f}}{\partial \mathbf{x}\partial \mathbf{c}}(\epsilon^3\mathbf{x}_2\mathbf{c}_1) + \frac{1}{6}\frac{\partial^3 \mathbf{f}}{\partial \mathbf{x}^3}(\epsilon\mathbf{x}_1)^3 \\ &+ \frac{1}{6}\frac{\partial^3 \mathbf{f}}{\partial \mathbf{c}^3}(\epsilon\mathbf{c}_1)^3 + \frac{\partial^3 \mathbf{f}}{\partial \mathbf{x}^2\partial \mathbf{c}}(\epsilon^3\mathbf{x}_1^2\mathbf{c}_1) + \frac{\partial^3 \mathbf{f}}{\partial \mathbf{x}\partial \mathbf{c}^2}(\epsilon^3\mathbf{x}_1\mathbf{c}_1^2)\end{aligned} \tag{16}$$

$$\begin{aligned}\mathbf{x}_3 = -\left(\frac{\partial \mathbf{f}}{\partial \mathbf{x}}\right)^{-1}\Bigg(&\frac{\partial^2 \mathbf{f}}{\partial \mathbf{x}^2}(\mathbf{x}_1\mathbf{x}_2) + \frac{\partial^2 \mathbf{f}}{\partial \mathbf{x}\partial \mathbf{c}}(\mathbf{x}_2\mathbf{c}_1) + \frac{1}{6}\frac{\partial^3 \mathbf{f}}{\partial \mathbf{x}^3}\mathbf{x}_1^3 \\ &+ \frac{1}{6}\frac{\partial^3 \mathbf{f}}{\partial \mathbf{c}^3}\mathbf{c}_1^3 + \frac{\partial^3 \mathbf{f}}{\partial \mathbf{x}^2\partial \mathbf{c}}(\mathbf{x}_1^2\mathbf{c}_1) + \frac{\partial^3 \mathbf{f}}{\partial \mathbf{x}\partial \mathbf{c}^2}(\mathbf{x}_1\mathbf{c}_1^2)\Bigg)\end{aligned} \tag{17}$$

As you can see we can compute the resulting $\Delta \mathbf{x}$ with increasing order and higher accuracy, which means that this perturbation method can perform in an effective way if we know $\Delta \mathbf{c}$.

In fact, in the actual coding work, higher-order derivative terms are still large, dense and extremely expensive to compute. Full derivatives are never needed, and what we care about are directional derivatives and Jacobian vector products. Thus at each order equation, the most expensive operation is solving a linear system

$$\mathbf{J}\mathbf{x}_i = \mathbf{b} \tag{18}$$

As a result, instead of finding the exact solution $\mathbf{x}_i^*$, we want a solution for

$$\min_{x_i} \|\mathbf{J}\mathbf{x}_i - \mathbf{b}\|^2 \qquad s.t. \quad \|\mathbf{x}_i\| \leq r \tag{19}$$

This is called trust region optimization and there are two possible solutions: either the true solution $\mathbf{x}_i^*$ lies within the trust region, or there is a scalar $\alpha > 0$ such that

$$(\mathbf{J} + \alpha I)\mathbf{x}_i = b \qquad \|\mathbf{x}_i\| = r \tag{20}$$

4 METHODS

4.1 Computational model

DESC solves the equilibrium problem defined by the following four sets of inputs: (1) one-dimensional pressure profile $p(\rho)$, (2) one-dimensional rotational transform profile $\iota(\rho)$, (3) Last Closed Flux Surface (LCFS) boundary shape, and (4) total toroidal magnetic flux enclosed by the LCFS [IG22]. Given the specified input, DESC solves the MHD equilibrium problem and is able to compute quantities of interest such as force balance error or magnetic field strength. In this study, we focus on the the forward propagation of uncertainty sourced from pressure and rotational transform on the force balance error.

4.2 Pressure profile

Pressure profile for the plasma is key to achieving equilibrium. Figure 2 illustrates two typical plasma pressure fields for the Wendelstein 7-X stellarator. For stellarator design, it's convenient to model the plasma pressure field using a one-dimensional polynomial series due to smoothness and axial-symmetry in equation (21). In addition, 0 gauge pressure at the boundary needs to be satisfied, as Figure 3 shows.

4.3 Force balance error

It is worthy to point out that DESC is an equilibrium code rather than dynamics code. The state of equilibrium is attempted to achieve by solving for the plasma state such that the difference between mechanical force and electromagnetic force is minimized. However, for stellarator with complex geometry and twisting, such

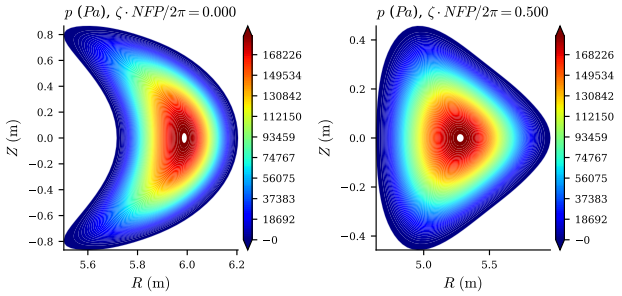


Figure 2: Sectional plasma pressure field for the Wendelstein 7-X stellarator

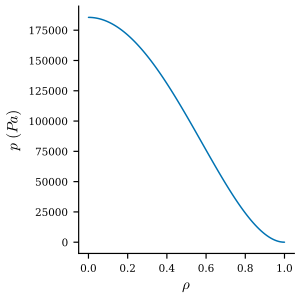


Figure 3: A one-dimensional pressure profile for the Wendelstein 7-X stellarator

force balance error is usually not zero. For example, Figure 4 shows the contour plots of force balance error at different cross sections of the Wendelstein 7-X stellarator given the pressure profile in Figure 3. It can be seen that although most of the internal areas has very low force balance error, there are regions around the boundary where the force balance error is high.

4.4 Current Limitations

The DESC simulation code that we propose to use to conduct numerical experiments has some limitations specifically for uncertainty quantification. The code is designed to perform single point analysis given a deterministic input configuration of the plasma field, hence it doesn't have native support of taking inputs of random variables and probability functions. A non-intrusive workaround is to treat the analysis code as a black box, and perform the uncertainty quantification by sampling and constructing surrogate model. The downside of such approach is the potentially high computational cost to construct high fidelity surrogates.

Furthermore, in terms of the solution technique, DESC utilizes perturbation method to incrementally solve the equilibrium problem with user-defined perturbation order. Considering the high-order perturbation terms can

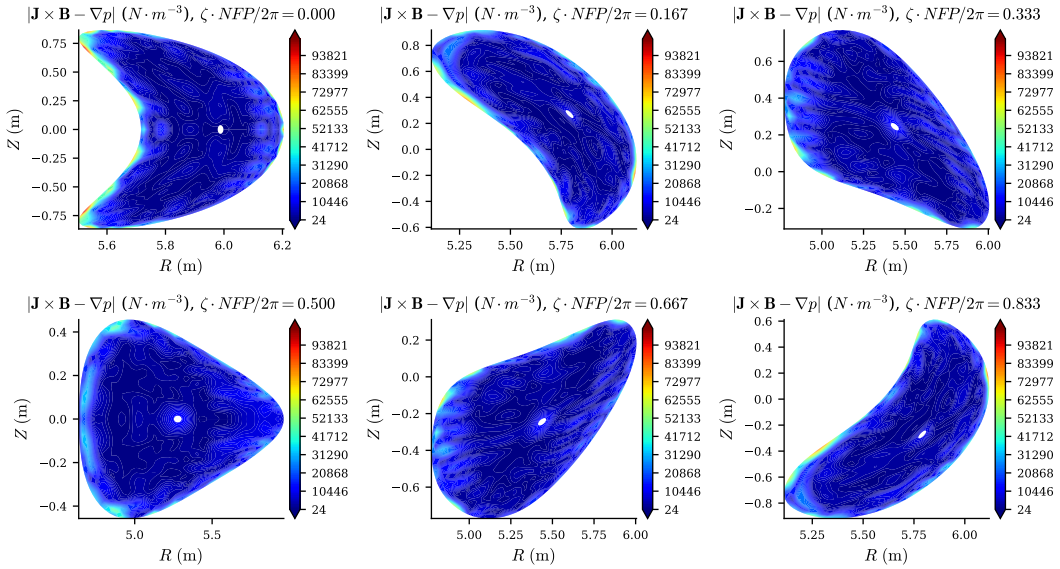


Figure 4: Force balance error at different sections for the Wendelstein 7-X Stellarator

lead to better accuracy with the cost of extra computational time. Such trade-off of precision and efficiency can also effect validity and feasibility of the proposed uncertainty quantification study.

4.5 Problem Statement

For our project, we decide to make improvement and comparison based on current perturbation method described above. The first part of our project is to assign distribution for input parameter set so that the system is able to simulate the whole process automatically and spontaneously with expected distribution generated from self-computing model. For 2D jointly space field generated by R_b and Z_b, we will assign Gaussian random field to them; for 1D profile function variables p and ι, we will assign Gaussian process to them, and for 1D distributed variable Ψ, we will assign 1D Gaussian distribution to it.

To perform these distributions, instead of directly replacing the original data array with random generated distributed data points, we should add the result with a positive parameter σ for an expected trending line [ea21]. Here we take pressure profile p as an example. Suppose the original p has a function with general form

$$P(p) = \sum_{i=1}^{n} p_i \rho^i \qquad (P_1, P_2, ..., P_n) \tag{21}$$

where ρ^i is an inner parameter for $P(p)$ and will generate original data array $(P_1, P_2, ..., P_n)$. Based on the original function, we produce a Gaussian process function

$$G(\rho) \approx \sum_{i=1}^{n} g_i \rho^i \qquad (G_1, G_2, ..., G_n) \tag{22}$$

Then the final pressure profile data array we will use to update the original one should be computed as

$$\hat{P}(\rho) = P(\rho)+G(\rho) \approx \sum_{i=1}^{n} p_i(1+\sigma g_i)\rho^i \qquad (\hat{P}_1, \hat{P}_2, ..., \hat{P}_n) \tag{23}$$

with new data array $(\hat{P}_1, \hat{P}_2, ..., \hat{P}_n)$.

Another task for our project is to simulate different UQ methods for the model and make a comparison between these UQ methods to solve the system. Since the current Taylor Series Approximation method need to have a bargain between accuracy and time complexity, we want to perform more approximation methods discussed in course such as Gaussian process approximation, polynomial chaos approximation, MCMC method, etc. Then we make error analysis for all modeling results and determine the best method for this kind of model. We can further explore the system and find out what features and properties lead to this method.

5 RESULTS

In this section, we present the uncertainty quantification results of the force balance error given uncertain pressure and rotational transform profiles. We use two methods to define the prior distribution of the inputs:

(1) direct perturbation of the coefficients of the pressure profile polynomial series, and (2) Gaussian process. Finally, we construct a neural network as the surrogate model and perform Monte Carlo simulation to simulate the distribution of the output using a larger number of sample points.

5.1 Direct coefficient perturbation

First, we generate the sample set of the input by directly perturbing the coefficients $\Delta p_i \sim U(-25, 25)$ of the predefined power series that prescribes the pressure profile subject to a uniform distribution:

$$\Delta p(\rho) = \Delta p_0 + \Delta p_2 \rho^2 + \Delta p_4 \rho^4 + \cdots + \Delta p_{10} \rho^{10} \quad (24)$$

Figure 5a shows 1000 perturbed pressure profiles that are to be used to evaluate the corresponding force balance error, Figure 5b shows Δp for each perturbation, Figure 5c shows the mean and standard deviation of the pressure profile samples as functions of radial coordinate ρ. Figure 6 shows the histogram and kernel density estimation that illustrate the distribution of the maximum force balance error using different sample size. It can be seen that the distributions as well as first two statistical moments are consistent as sample size grows.

5.2 Gaussian Process

We utilized a Gaussian process to generate a perturbed distribution for the pressure profile and rotational transform. Specifically, we employed a Radial Basis Function (RBF) kernel

$$k(\rho_i, \rho_j) = exp\left(\frac{d(\rho_i, \rho_j)^2}{2l^2}\right) \quad (25)$$

with a parameter l that controls the length scale. For our analysis, we set $l = 0.5$ and present the resulting graphs for pressure profile perturbations. Figure 7 To normalize all perturbed pressure profiles, we fixed the point at $(1, 1)$. The corresponding result graph is displayed below. Figure 8 We do the same performance for rotational transform and normalized it so that two points $(0, 0)$ and $(1, 1)$ are fixed. Figure 9 Figure 10

Based on the perturbed input parameters, we generated possible output distributions for force balance error and coefficients within each input file. The resulting graph below shows the force balance error in the pressure profile with different sample size from 100 to 1000. We also demonstrate how the mean and standard deviation change with sample size. Figure 13

Additionally, we present the distributions of each coefficient in the pressure profile and rotational transform with various degrees. For instance, in the case of the DSHAPE example, the input pressure profile is represented by

$$p(\rho) = 1600 - 3200\rho^2 + 1600\rho^4 \quad (26)$$

with three initial parameters of $[1600, -3200, 1600]$. We display the resulting output distributions for all three parameters below. Figure 14a Figure 14b Figure 14c

We also applied the same technique to the rotational transform. As a reminder, the input rotational transform profile is expressed as

$$\iota(\rho) = -1 + 0.67\rho^2 \quad (27)$$

with two initial parameters of $[-1, 0.67]$. We present the resulting output distributions for both parameters below. Figure 15a Figure 15b

5.3 Monte Carlo Simulation using Neural Network Surrogate Model

Finally, we utilize surrogate models to simulate the output distributions. The benefit of surrogate model is to drastically speed up the evaluation of the output given a new perturbed input. In this study, a neural network with two hidden layers is used. The network is fitted using the data sets generated in the previous sections. Training is performed with Adam optimizer that employs a weight decay strategy. Figure 11 shows the validation of the model fit by visualizing the actual value v.s. predicted value by the neural network. It can be seen that good agreement with few outliers is achieved. Figure 12 shows the distribution histogram by evaluating 50000 samples using the trained neural network. It can be seen that the mean and standard deviation match the results of the real simulation shown in Figure 6. However with the neural network a very large sample size ($N = 50000$) can be used, as a result, a refined histogram is produced, which captures the second peak at a relative lower $\max|F|$ that is not captured in Figure 6.

6 CONCLUSIONS AND FURTHER STUDIES

In this project, our goal is to develop an optimized perturbation method for the DESC Stellarator system, which can detect changes in the equilibrium of constraint functions for perturbed input parameters $\mathbf{f}(x + \Delta x, c + \Delta c) = 0$. We started by examining the current perturbation method, which extends the Taylor Series and detects force error balance at different orders. Our aim is to improve the current method by incorporating direct coefficient perturbation, Gaussian Process and Monte Carlo distributions for input parameters such as pressure profile (p) and rotational transform (ι). We also present the resulting output distributions for force balance error and coefficients at different orders based on the perturbed input distributions. Our project has several key takeaways, such as the effectiveness of the perturbation method and continuation technique for efficient Monte Carlo simulation, and the significant impact of prior distribution assumptions on output

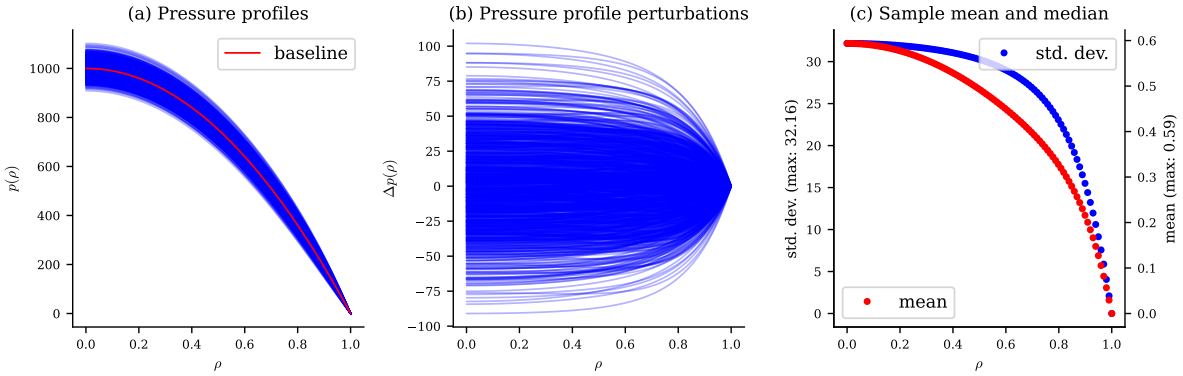


Figure 5: Pressure profile samples by direct coefficient perturbation

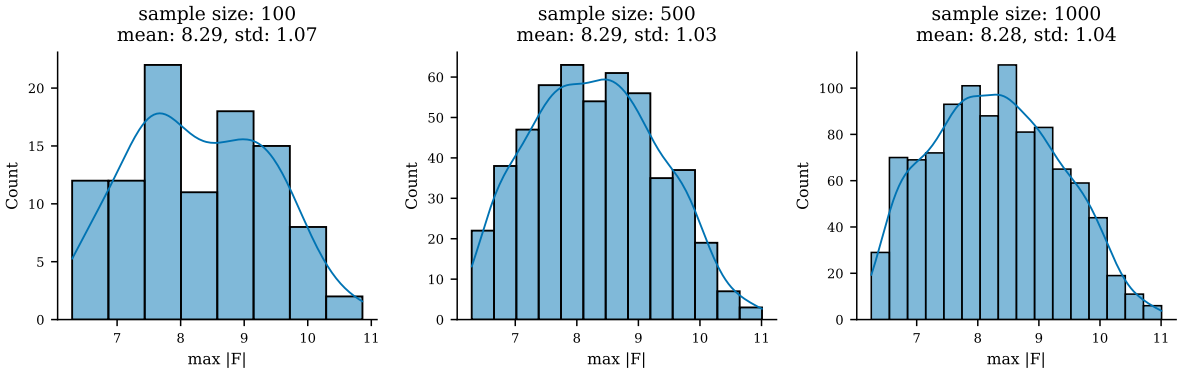


Figure 6: Distribution of $|F|$ for different sample size N

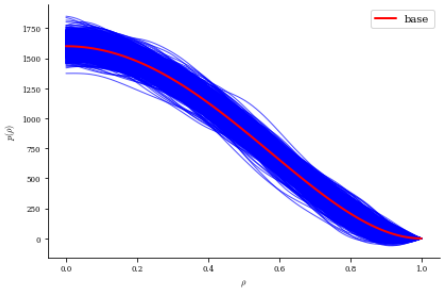


Figure 7: Pressure Profile Perturbations, length scale $l = 0.5$

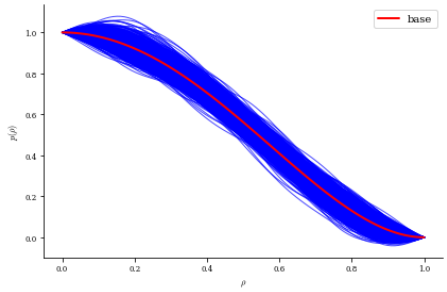


Figure 8: Normalized Pressure Profile Perturbations

distribution. There are several potential avenues for future work in our project. One direction is to extend our analysis to higher-dimensional input functionals, such as R_b and Z_b, using the Gaussian Random Field method. We can also explore the output distributions of other quantities of interest, such as the magnetic field. Additionally, we could employ additional methods to approximate our results, such as polynomial chaos expansion, pseudo-spectral projection, and point allocation methods.

7 REPLICATION OF RESULTS

The code for this work can be found on GitHub (link).

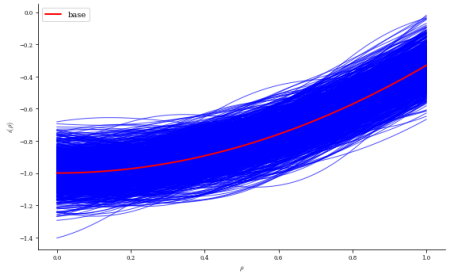


Figure 9: Rotational Transform Profile Perturbations

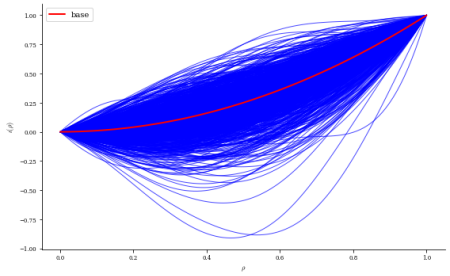


Figure 10: Normalized Pressure Profile Perturbations

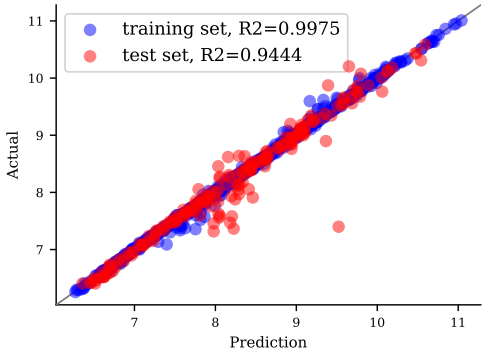


Figure 11: Validation for the neural network

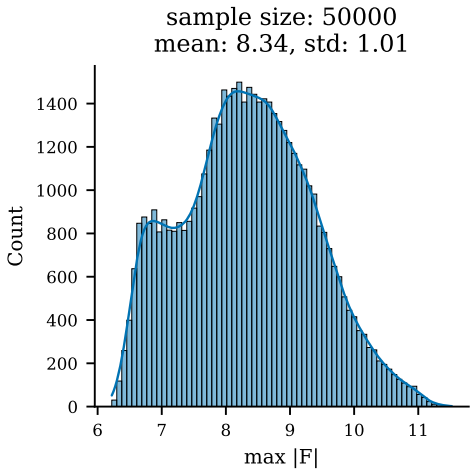


Figure 12: Distribution of the output sampled from the neural network with $N = 50000$

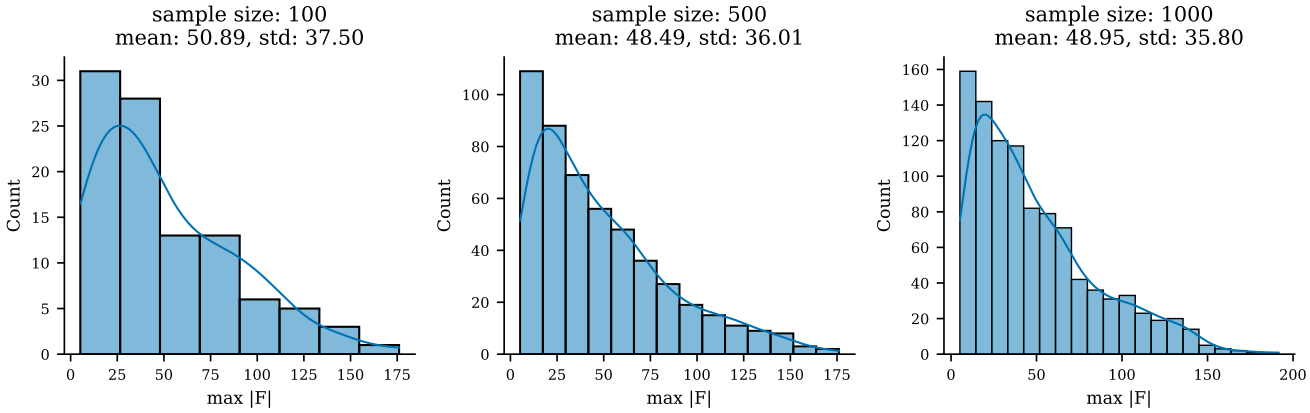
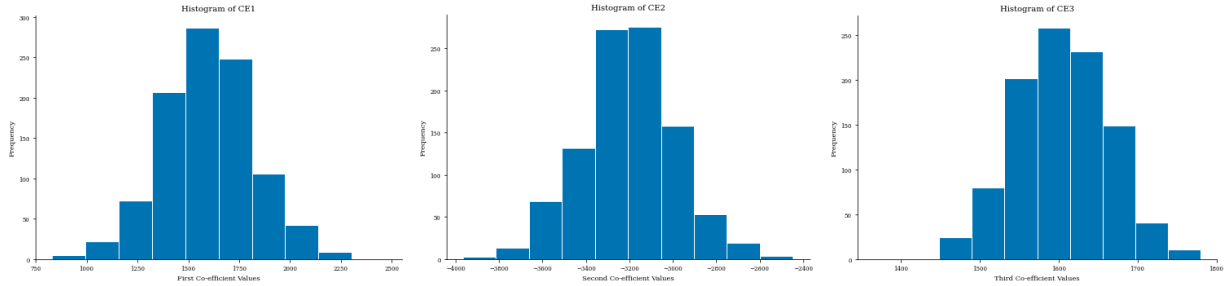
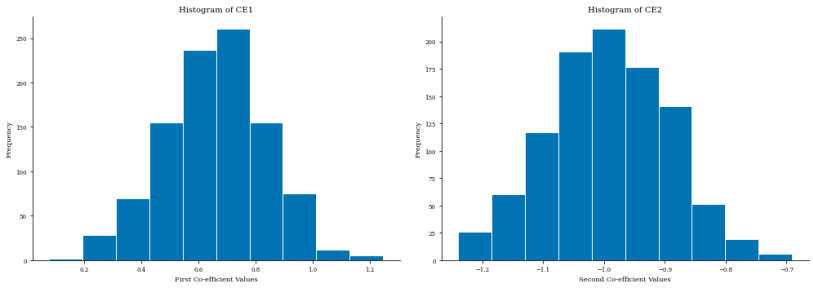


Figure 13: Distribution of $|F|$ for Pressure Profile with Different Sample Size



(a) First Coefficient **(b) Second Coefficient** **(c) Third Coefficient**

Figure 14: Distribution of the Coefficients for the Pressure Profile



(a) Distribution of First Coefficient for Rotational Transform Profile

(b) Distribution of Second Coefficient for Rotational Transform Profile

Figure 15: Distribution of the Coefficients for the Rotational Transform Profile

REFERENCES

[CD22] Rory Conlin and Daniel W. Dudt. The desc stellarator code suite part ii: Perturbation and continuation methods. 2022.

[DK20] D. Dudt and E. Kolemen. Desc: A stellarator equilibrium solver. *Phys. Plasmas 27, 102513*, 2020.

[ea21] Andrea Merlo et al. Proof of concept of a fast surrogate model of the vmec code via neural networks in wendelstein 7- x scenarios. 2021.

[IG22] Lise-Marie Imbert-Gérard. An introduction to stellarators: From magnetic fields to symmetries and optimization. 2022.

[PC22] D. Panici and R. Conlin. The desc stellarator code suite part i: Quick and accurate equilibria computations. *Princeton University*, 2022.