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

ABSTRACT

In this study, the robustness of stellarator design and optimization is studied and potentially improved by quantifying forward uncertainties. The numerical experiments will be conducted using DESC code that solves the three-dimensional magnetohydrodynamics equilibrium. Such discretized equilibria governing equations are solved using spectral method. Perturbation method is used to incrementally solve for the plasma state of a complex geometry configuration by sequentially perturbing from a standard simple geometry. Different orders of perturbation can be defined based on the requirement of accuracy. In terms of the study of uncertainty quantification, various approaches including Gaussian process, polynomial chaos approximation and reduced basis approximation are proposed to use. The comparison among these different approaches will be presented in the final report.

1 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 [1], 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.

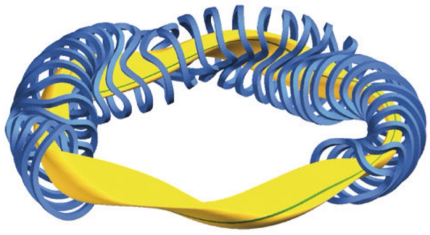

Figure 1: the Wendelstein 7-X stellarator

2 LITERATURE REVIEW

2.1 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. 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 sub-section.

2.2 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 [3]. 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

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

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

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

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

3 PROPOSED PROJECT IDEAS

3.1 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 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.

3.2 Improvements, Expectations and Conclusions

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. 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, reduced basis approximation, 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.

In summary, the current perturbation method has some limitations such as lack of inputs control and having a bargain between time complexity and accuracy. We want to improve this UQ model by assigning distributions for input parameter sets and implementing different UQ methods to solve the model and make a reasonable comparison. Yulong plans to finish first task (assigning distributions) by the end of March, and Yicong plans to finish the second task (implement different UQ methods) by the end of March. Then we will compare and do result analysis together and generate final report by the end of April. We believe our final experiment results and report can contribute to the current Stellarator DESC coding work from the view of Uncertainty Quantification.

4 REFERENCES

[1] DESC: A Stellarator Equilibrium Solver, D. Dudt and E. Kolemen, Phys. Plasmas 27, 102513 (2020)
[2] The DESC Stellarator Code Suite Part I: Quick and Accurate Equilibria Computations, D. Panici and R. Conlin (2022)
[3] The DESC Stellarator Code Suite Part II: Perturbation and Continuation Methods, Rory Conlin and Daniel W. Dudt (2022)
[4] An Introduction to Stellarators: From Magnetic Fields to Symmetries and Optimization, Lise-Marie Imbert-Gérard (2020)