A HERMITE LEAST SQUARES METHOD FOR MINIMIZING FAILURE PROBABILITIES

M. Fuhrländer*, **S. Schöps***

* Computational Electromagnetics, Technische Universität Darmstadt, Germany
mona.fuhrlaender@tu-darmstadt.de, sebastian.schoeps@tu-darmstadt.de

Keywords: uncertainty quantification, failure probability, derivative-free optimization.

Abstract

In this work we propose a new Hermite least squares optimization method for problems in electrical engineering. The aim is to solve bound constrained non-linear optimization problems, where the derivatives of the objective function are available with respect to some optimization variables, while for others they are not. This method is highly relevant for failure probability minimization having deterministic and uncertain optimization variables.

1 Introduction

In mass production often there are uncertainties caused by manufacturing imperfections, natural material deviations or environmental influences. These may result in deviations in the design parameters which may lead to deviations in operation violating the performance requirements. To avoid rejections of devices due to malfunctioning, the design can be optimized to reach higher reliability in case of uncertainties. Later we will consider a waveguide as depicted in Fig. 1. Depending on the type of uncertainties and models, the derivatives of the objective function with respect to the optimization variables may be available or not. Using gradient based optimization methods would require the computationally expensive approximation of the missing partial derivatives (e.g. by finite differences), however, using derivative-free optimization (DFO) methods would waste available information. We propose a Hermite least squares optimization method, which is a modification of Powell's BOBYQA (Bound constrained Optimization BY Quadratic Approximation) method [1], and is well suited for this case of mixed gradient information.

2 Failure probability

We introduce two types of design parameters: uncertain Gaussian distributed parameters $\boldsymbol{\xi} \sim \mathcal{N}(\overline{\boldsymbol{\xi}}, \boldsymbol{\Sigma})$, with $\overline{\boldsymbol{\xi}} \in \mathbb{R}^{n_{\boldsymbol{\xi}}}$, $\boldsymbol{\Sigma} \in \mathbb{R}^{n_{\boldsymbol{\xi}} \times n_{\boldsymbol{\xi}}}$, and probability density function $\varphi(\boldsymbol{\xi})$, and deterministic parameters $\mathbf{d} \in \mathbb{R}^{n_{\mathbf{d}}}$. Let $Q : \mathbb{R}^{n_{\boldsymbol{\xi}}+n_{\mathbf{d}}} \to \mathbb{R}$ denote a quantity of interest (QoI) and $c \in \mathbb{R}$. Then, we can denote the performance feature specification for a device by

$$Q(\boldsymbol{\xi}, \mathbf{d}) \leq c. \tag{1}$$

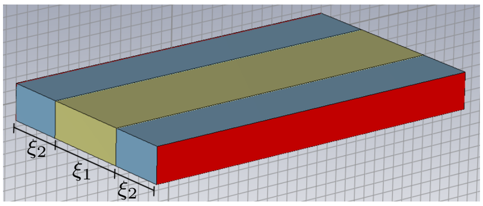

Figure 1: CST model of a waveguide with two uncertain geometry parameters ξ_1 and ξ_2.

We define the failure domain as the set of all parameter combinations of $\boldsymbol{\xi}$ such that (1) is violated, i.e.,

$$\mathcal{F}_{\mathbf{d}} := \{\boldsymbol{\xi} : Q(\boldsymbol{\xi}, \mathbf{d}) > c\}. \tag{2}$$

Then, the failure probability is defined by

$$\mathbb{P}_{\mathrm{f}}(\overline{\boldsymbol{\xi}}, \mathbf{d}) \equiv \mathbb{P}[\boldsymbol{\xi} \in \mathcal{F}_{\mathbf{d}}] = \mathbb{E}[\mathbf{1}_{\mathcal{F}_{\mathbf{d}}}(\boldsymbol{\xi})] = \int_{\mathbb{R}^{n_{\boldsymbol{\xi}}}} \mathbf{1}_{\mathcal{F}_{\mathbf{d}}}(\boldsymbol{\xi}) \varphi(\boldsymbol{\xi}) \, \mathrm{d}\boldsymbol{\xi}, \tag{3}$$

where $\mathbf{1}_{\mathcal{F}_{\mathbf{d}}}(\boldsymbol{\xi})$ denotes the indicator function with value 1 if $\boldsymbol{\xi}$ lies in $\mathcal{F}_{\mathbf{d}}$ and value 0 otherwise. Eq. (3) can be calculated with a Monte Carlo (MC) analysis which requires many evaluations of the QoI, typically involving computationally expensive simulations. Even with efficient modifications of MC, see e.g. [2], each evaluation of (3) remains time consuming. Hence, the main goal of the optimization method we will propose is to reduce the number of objective function calls compared to classic gradient based or DFO solvers.

3 The optimization problem

We aim to minimize the failure probability under bound constraints. The optimization problem reads

$$\min_{\overline{\boldsymbol{\xi}}, \mathbf{d}} \mathbb{P}_{\mathrm{f}}(\overline{\boldsymbol{\xi}}, \mathbf{d}) \quad \text{s.t.} \quad \overline{\boldsymbol{\xi}}_{\mathrm{lb}} \leq \overline{\boldsymbol{\xi}} \leq \overline{\boldsymbol{\xi}}_{\mathrm{ub}} \ \wedge \ \mathbf{d}_{\mathrm{lb}} \leq \mathbf{d} \leq \mathbf{d}_{\mathrm{ub}}. \tag{4}$$

The gradient of $\mathbb{P}_{\mathrm{f}}(\overline{\boldsymbol{\xi}}, \mathbf{d})$ with respect to $\overline{\boldsymbol{\xi}}$ can be easily calculated since the probability density function $\varphi(\boldsymbol{\xi})$ of the Gaussian distribution is an exponential function. Using MC based estimation methods, this gradient can be obtained without additional computing effort [3]. On the other hand, the gradient with respect to the deterministic parameter requires the differentiation of the indicator function. Hence, we consider this gradient as not available.

4 Hermite least squares optimization

Powell's BOBYQA method is a DFO technique, which uses (underdetermined) interpolation to build a quadratic model of the objective function in each iteration. Then, the quadratic subproblem is solved in a trust region. The subproblem's solution is added to the interpolation set, while another point is deleted, ensuring that the interpolation set remains well balanced. Based on the Python implementation PyBOBYQA by Cartis et al. [4], we modify this algorithm in order to handle derivative information. By $\mathbf{x} = [\boldsymbol{\xi}, \mathbf{d}] \in \mathbb{R}^{n_{\boldsymbol{\xi}}+n_{\mathbf{d}}} = \mathbb{R}^{n_{\mathbf{x}}}$ we denote the optimization variable. Let

$$\mathcal{T} = \{(\mathbf{x}^1, \mathbb{P}_{\mathrm{f}}(\mathbf{x}^1)), \dots, (\mathbf{x}^p, \mathbb{P}_{\mathrm{f}}(\mathbf{x}^p))\} \quad (5)$$

denote a training data set and $\Phi = \{\phi_1(\mathbf{x}), \dots, \phi_q(\mathbf{x})\}$ the monomial basis of second degree. We build a system of linear equations

$$\mathbf{M}\mathbf{v} = \mathbf{b}, \quad (6)$$

with $M_{ij} = \phi_j(\mathbf{x}^i)$ and $b_i = \mathbb{P}_{\mathrm{f}}(\mathbf{x}^i)$. We include derivative information with respect to $\boldsymbol{\xi}$ into the training data set, i.e.,

$$\mathcal{T}_{\mathrm{H}} = \Bigg\{ \left(\mathbf{x}^1, \mathbb{P}_{\mathrm{f}}(\mathbf{x}^1), \frac{\partial}{\partial x_1}\mathbb{P}_{\mathrm{f}}(\mathbf{x}^1), \dots, \frac{\partial}{\partial x_{n_{\boldsymbol{\xi}}}}\mathbb{P}_{\mathrm{f}}(\mathbf{x}^1)\right), \dots, \left(\mathbf{x}^p, \mathbb{P}_{\mathrm{f}}(\mathbf{x}^p), \frac{\partial}{\partial x_1}\mathbb{P}_{\mathrm{f}}(\mathbf{x}^p), \dots, \frac{\partial}{\partial x_{n_{\boldsymbol{\xi}}}}\mathbb{P}_{\mathrm{f}}(\mathbf{x}^p)\right) \Bigg\}. \quad (7)$$

We build the Hermite system of equations

$$\mathbf{M}_{\mathrm{H}}\mathbf{v}_{\mathrm{H}} = \mathbf{b}_{\mathrm{H}}, \quad (8)$$

where

$$\mathbf{M}_{\mathrm{H}} = \begin{bmatrix} \mathbf{M} \\ \mathbf{M}^{(1)} \\ \vdots \\ \mathbf{M}^{(n_{\boldsymbol{\xi}})} \end{bmatrix} \quad \text{and} \quad \mathbf{b}_{\mathrm{H}} = \begin{bmatrix} \mathbf{b} \\ \mathbf{b}^{(1)} \\ \vdots \\ \mathbf{b}^{(n_{\boldsymbol{\xi}})} \end{bmatrix} \quad (9)$$

with $M_{ij}^{(k)} = \frac{\partial}{\partial x_k}\phi_j(\mathbf{x}^i)$ and $b_i^{(k)} = \frac{\partial}{\partial x_k}\mathbb{P}_{\mathrm{f}}(\mathbf{x}^i)$. The system (8) is overdetermined for $q < p(n_{\boldsymbol{\xi}}+1)$ and can be solved with least squares regression. We obtain a quadratic subproblem, which is solved in a trust region, analogously to the PyBOBYQA method, update the training data set and proceed with the next iteration. For details see [5].

5 Numerical results

We design a simple dielectrical waveguide with two uncertain geometry parameters and two deterministic material parameters. The initial failure probability is 0.57. We compare PyBOBYQA and Hermite least squares in their default settings. The results are provided in Table 1. The computational effort, measured by the number of objective function calls, is $27\,\%$ lower when using the proposed Hermite least squares method.

Method	$\mathbb{P}_{\mathrm{f}}^{\mathrm{opt}}$	# fct. calls
PyBOBYQA	0.02	73
Hermite l.s.	0.0036	53

Table 1: Optimization results for waveguide.

Further, we benchmarked the method on a test set of 29 non-linear optimization problems, assuming different numbers and combinations of derivative directions being available. The average computing effort for the five-dimensional problems is summarized in Fig. 2. We observe a significant decrease of effort by $35-52\,\%$.

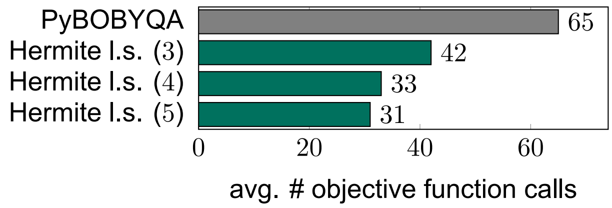

Figure 2: Computing effort considering different numbers of derivative directions as available.

6 Conclusion

We proposed the Hermite least squares method well suited to handle optimization problems with mixed gradient information. This method is highly relevant for failure probability optimization with deterministic and Gaussian distributed uncertain optimization variables. We showed numerically that the computing effort can be reduced by up to $52\,\%$ compared to the state of the art derivative-free PyBOBYQA method.

References

[1] M. J. D. Powell. "The BOBYQA algorithm for bound constrained optimization without derivatives". In: *Cambridge NA Report NA2009/06* (2009).

[2] M. Fuhrländer and S. Schöps. "A Blackbox Yield Estimation Workflow with Gaussian Process Regression Applied to the Design of Electromagnetic Devices". In: *Journal of Mathematics in Industry* 10.1 (2020).

[3] Helmut E. Gräb. *Analog Design Centering and Sizing*. Springer, 2007.

[4] C. Cartis et al. "Improving the Flexibility and Robustness of Model-Based Derivative-Free Optimization Solvers". In: *ACM Transactions on Mathematical Software* 45.3 (2019).

[5] M. Fuhrländer and S. Schöps. "Hermite-type modifications of BOBYQA for optimization with some partial derivatives". In: *arXiv:* 2204.05022 (2022).