



**HAL**  
open science

# Optimization and reliability analysis for electrical machines modeled by finite element method

Reda El Bechari

► **To cite this version:**

Reda El Bechari. Optimization and reliability analysis for electrical machines modeled by finite element method. Other. Centrale Lille Institut, 2020. English. NNT : 2020CLIL0007 . tel-03152165

**HAL Id: tel-03152165**

**<https://theses.hal.science/tel-03152165v1>**

Submitted on 25 Feb 2021

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

**CENTRALE LILLE**

**THESE**

**Présentée en vue  
d'obtenir le grade de**

**DOCTEUR**

**En**

**Spécialité : Génie Électrique**

**Par**

**Reda El Bechari**

**DOCTORAT DELIVRE PAR CENTRALE LILLE**

**Titre de la thèse :**

**Optimisation et analyse de fiabilité des machines électrique modélisées par la  
méthode des éléments finis**

**Optimization and Reliability Analysis for Electrical Machines modeled by  
Finite Element Method**

**Soutenue le 17/09/2020 devant le jury d'examen :**

<b>Président et Rapporteur</b>	Frédéric MESSINE, Professeur, ENSEEIHT Toulouse, France
<b>Rapporteur</b>	Piergiorgio ALOTTO, Professeur, Université de Padova, Italie
<b>Examinatrice</b>	Maya HAGE-HASSAN, Maitre de conférences, CentraleSupélec Paris-Saclay, France
<b>Examinatrice</b>	Jocelyne ERHEL, DR émérite, INRIA Rennes Bretagne Atlantique, France
<b>Directeur de thèse</b>	Stéphane BRISSET, Professeur, Centrale Lille, France
<b>Directeur de thèse</b>	Stéphane CLÉNET, Professeur, Arts et Métiers Sciences et Technologies, France
<b>Encadrant</b>	Frédéric GUYOMARCH, Maitre de conférences, Université de Lille, France
<b>Encadrant</b>	Jean Claude MIPO, Docteur, Valeo Powertrain Systems, France
<b>Invité</b>	Thierry COOREVITS, Maitre de conférences, Arts et Métiers Sciences et Technologies, France

**Thèse préparée dans le Laboratoire L2EP  
Ecole Doctorale SPI 072**



# Dedication

To Mom and Dad.

To me for surviving the coronavirus pandemic of 2020, and for writing most of this dissertation during the lockdown (4 months) while alone in a small apartment.





# Declaration

I, Reda El Bechari, declare that this thesis titled, *Optimization and Reliability Analysis for Electrical Machines modelled by Finite Element Method* and the work presented in it are my own. I confirm that:

- This work was done mainly while in candidature for a Ph.D. degree at Centrale Lille.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work.
- I have acknowledged all main sources of help.
- Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.

Signed:

---

Date:



# Acknowledgements

First of all, I would like to thank my thesis advisors, Pr. Stéphanne Brisset, Pr. Stéphanne Clenet, Associate Pr. Frédéric Guyomarch, and Dr. Jean Claude Mipo, for their availability, great mastery of the different subjects treated in this thesis, and ability to formulate complex ideas in simple words.

I would like to thank the jury members for making the defense day possible and pleasant despite the Corona pandemic. Thank you Pr. Frédéric Messine and Pr. Piergiorgio Alotto for accepting the duty of rapporteur to my thesis and your relevant suggestions and useful interrogations; Jocelyne Erhel (DR émérite), Associate Pr. Maya Hage-Hassan, and Associate Pr. Thierry Coorevits for providing insightful comments and questions as jury members.

Then I would like to thank all my colleagues of the L2EP: Emre, Nabil, Wissem, Jérôme, and especially those with whom I shared the office in Centrale Lille at first and then the ESPRIT building afterward, Siyang, Bilquis, the other Reda, Xin, and Lorraine.

Finally, I would like to thank all my family members who have supported me during these three years. Last but certainly not the least, most sincere thanks to Mariem for her encouragement and support.



# Abstract

This dissertation deals with the approaches for optimization and reliability analysis of electrical machines modelled by the finite element method. The finite element method is the most sophisticated tool to model the electromagnetic phenomenon. However, it is computationally expensive. Thus, its usage for optimization and reliability analysis (iterative processes) should be made with caution since only a limited number of evaluations of the model can be tolerated. Furthermore, the impact of the manufacturing process on the electrical machines is scarcely studied in the literature. The integration of this aspect in the design phase is one of the contributions of this thesis alongside the main contribution, which is the development and comparison of optimization approaches for electrical machines.

We present the approaches adapted to the subject and develop new ones. On the one hand, the finite element model can be seen as a "black-box" for which we develop a non-intrusive approach based on Kriging meta-models. On the other hand, we consider an intrusive approach as we look inside the "black-box," we upgrade the model to provide the derivatives of the quantities of interest. The derivatives are essential to some optimization and reliability analysis tools. They are computed efficiently using the adjoint variable method. Finally, the methods are compared to give insight into the advantages and the shortcomings of each of them.

Lastly, a real case study is considered; it consists of studying the impact of the manufacturing process on a claw-pole machine manufactured by Valeo. From the production line, machines are withdrawn to measure their dimensions and characterize their deviation from the nominal one. Then a statistical analysis is conducted to assess the reliability and impact on the performances.



# Résumé

Ce mémoire aborde les approches d'optimisation et d'analyse de la fiabilité des machines électriques modélisées par la méthode des éléments finis. La méthode des éléments finis est l'outil le plus sophistiqué pour modéliser le phénomène électromagnétique. Cependant, elle est coûteuse en temps de calcul. Ainsi, son utilisation pour l'optimisation et l'analyse de fiabilité (processus itératifs) doit être faite avec prudence car seul un nombre limité d'évaluations du modèle peut être toléré. De plus, l'impact du processus de fabrication sur les machines électriques est peu étudié dans la littérature. L'intégration de cet aspect dans la phase de conception est l'un des apports de cette thèse aux côtés de la principale contribution, qui est le développement et la comparaison des approches d'optimisation pour les machines électriques.

Nous exposons les approches adaptées au sujet de la thèse et en développons de nouvelles. D'une part, le modèle d'éléments finis peut être considéré comme une "boîte noire" pour laquelle nous développons une approche non intrusive basée sur les méta-modèles de Krigeage. D'autre part, nous considérons une approche intrusive car nous regardons à l'intérieur de la "boîte noire", nous améliorons le modèle pour fournir les dérivées des quantités d'intérêt. Les dérivées sont essentielles à certains outils d'optimisation et d'analyse de fiabilité. Elles sont calculées efficacement en utilisant la méthode de la variable adjointe. Enfin, les méthodes sont comparées pour donner un aperçu des avantages et des inconvénients de chacune d'entre elles.

Enfin, une étude de cas réel est abordée; elle consiste à étudier l'impact du procédé de fabrication sur la machine à griffes fabriquée par Valeo. Sur la chaîne de production, les machines sont prélevées pour mesurer leurs dimensions et caractériser leur écart par rapport aux dimensions nominales. Ensuite, une analyse statistique est menée pour évaluer la fiabilité et l'impact sur les performances.





# Résumé substantiel

## Introduction

La méthode des éléments finis est l'outil le plus sophistiqué pour modéliser le phénomène électromagnétique. Cependant, elle est coûteuse en temps de calcul. Ainsi, son utilisation pour l'optimisation et l'analyse de fiabilité (processus itératifs) doit être faite avec prudence car seul un nombre limité d'évaluations du modèle peut être toléré. De plus, l'impact du processus de fabrication sur les machines électriques est peu étudié dans la littérature. L'intégration de cet aspect dans la phase de conception est l'un des apports de cette thèse aux côtés de la principale contribution, qui est le développement et la comparaison des approches d'optimisation pour les machines électriques.

## Etat de l'art

Ce chapitre examine les algorithmes d'optimisation et la méthode des éléments finis pour mettre en perspective les contributions de cette thèse. Tous les algorithmes d'optimisation ne sont pas adaptés aux modèles à base d'éléments finis en raison du coût de calcul de ces modèles. Nous présentons les principales catégories d'algorithmes d'optimisation et leurs extensions au contexte de l'optimisation sous incertitudes. Une brève présentation de la modélisation électromagnétique des machines électriques utilisant la méthode des éléments finis est également présentée. Deux approches adaptées au sujet de cette thèse sont sélectionnées pour être investiguées.

## Méthode non-intrusive

Ce chapitre présente une approche non-intrusive dans laquelle le modèle éléments finis est considéré comme une boîte noire, et un modèle de substitution est créé en évaluant le modèle coûteux en quelques points, puis en utilisant le modèle de substitution peu coûteux pour l'optimisation et la quantification des incertitudes. Nous présentons une introduction approfondie de l'optimisation à l'aide de méta-modèles

---

et des défis posés par la mise en place de "bons" méta-modèles et la manière dont on peut évaluer leur qualité, puis nous proposons une nouvelle stratégie d'exploitation des modèles de krigeage.

## Méthode intrusive

Ce chapitre examine une approche intrusive pour calculer les dérivés d'un modèle à éléments finis. Dans cette approche, la majeure partie du travail est effectuée sur le modèle plutôt que sur l'algorithme. Les dérivées sont utilisées par les algorithmes d'optimisation et les méthodes de quantification de l'incertitude les plus efficaces. Nous présentons la méthode de la variable d'état adjointe et la façon dont elle peut être obtenue à partir d'un code éléments finis. La méthode de la variable adjointe permet de calculer efficacement les dérivées à partir du code d'éléments finis.

## Benchmark et comparaisons

Ce chapitre est consacré aux tests numériques des approches développées dans le cadre de l'optimisation. Pour ce faire, nous abordons deux benchmarks bien connus traités par les chercheurs de la communauté électromagnétique. Il s'agit des cas tests issus des workshops TEAM (Testing Electromagnetic Analysis Methods) [1] [2]. Nous présentons une comparaison entre les approches étudiées et d'autres approches pour l'optimisation des dispositifs électromagnétiques.

## Machine à griffes

Ce chapitre traite l'impact des imperfections du processus de fabrication sur les performances de la machine à griffes. Nous étudions principalement les imperfections géométriques. Dans ce chapitre, nous présentons la machine à griffes et son mode de fonctionnement puis nous introduisons la procédure de métrologie adoptée pour les mesures métrologiques de la machine électrique. Ces mesures sont analysées, et la variabilité de la géométrie de la machine électrique due au processus de fabrication est modélisée. Ensuite, un modèle éléments finis paramétré de la machine est présenté. Puis, la propagation de l'incertitude utilisant la variabilité caractérisée et le modèle de la machine est réalisée en utilisant la simulation de Monte Carlo avec le modèle de Krige.

---

## Conclusion

Cette thèse traitait des approches d'optimisation et d'analyse de la fiabilité des machines électriques modélisées par la méthode des éléments finis. La méthode des éléments finis est l'outil le plus sophistiqué pour modéliser les phénomènes électromagnétiques dans les machines électriques. Elle permet de modéliser finement les champs électromagnétiques dans le domaine étudié et de manipuler des géométries complexes, comme la machine à griffes. Cependant, elle est coûteuse en temps de calcul à cause des matériaux non linéaires, des géométries 3D, de la dépendance au temps. Par conséquent, son utilisation dans le cadre de l'optimisation et de l'analyse de fiabilité (processus itératifs) doit être faite avec précaution, car seul un nombre limité d'évaluations de l'outil de simulation peut être toléré.

Les algorithmes d'optimisation générale ne peuvent pas être appliqués tels quels ; certaines modifications indispensables sont nécessaires pour les rendre utilisables pour l'optimisation des machines électriques. Dans le premier chapitre, nous avons présenté la littérature sur les algorithmes d'optimisation, et nous avons discuté de leur adaptabilité au sujet de cette thèse. À partir des différentes catégories, deux approches ont été envisagées, qui dépendent des capacités du modèle utilisé.

1. Une approche non-intrusive basée sur le méta-modèle de Krigeage
2. Une approche intrusive basée sur la méthode de la variable adjointe

L'approche non-intrusive utilisant des modèles de substitution est largement utilisée dans le contexte de l'optimisation de modèles coûteux. De nombreux chercheurs ont souligné les difficultés que pose l'utilisation d'une telle stratégie. Dans cette thèse, nous avons mis l'accent sur certains des problèmes très connus et proposé une nouvelle méthodologie d'utilisation des méta-modèles et la manière d'accélérer les temps d'optimisation. Nous avons souligné les inconvénients de l'approche conventionnelle, qui consiste à faire tenir un seul méta-modèle sur l'ensemble de l'espace de conception et à l'enrichir séquentiellement à l'aide de critères de remplissage. Nous avons proposé une nouvelle stratégie qui consiste à construire de nombreux méta-modèles sur des régions spécifiques de l'espace de recherche. Ensuite, de manière itérative, on élague les régions qui ne sont pas prometteuses. Ce processus explore tout l'espace de recherche ; il nous permet donc de produire des solutions fiables.

D'autre part, pour l'approche intrusive, nous proposons comment calculer les dérivés d'un modèle d'éléments finis. La plupart des travaux ont été effectués sur le modèle plutôt que sur l'algorithme (nous avons utilisé l'algorithme SQP pour cette thèse). Nous avons présenté la méthode de la variable adjointe et la manière dont le gradient peut être dérivé d'un code FEM. Nous avons développé une méthode efficace pour calculer les dérivées des sensibilités de forme pour les paramètres géométriques, qui sont essentiels pour le calcul du gradient. La méthode de la variable adjointe a

---

été comparée à la méthode des différences finies pour valider et mettre en évidence son efficacité en termes de précision et de temps de calcul.

Pour la comparaison entre des approches intrusives et non-intrusives pour l'optimisation des dispositifs électromagnétiques à l'aide de la méthode des éléments finis, nous avons traité deux problèmes de référence bien connus de la littérature connus sous le nom de TEAM Workshop : les problèmes 22 et 25 [1] [2]. Nous avons utilisé deux métriques pour la comparaison, la première est la qualité de la solution, et la seconde est le coût de calcul.

En termes de performances, l'approche intrusive surpasse les autres stratégies pour les deux cas de test ; cela a été possible grâce au calcul du gradient en utilisant la méthode de la variable adjointe. Cela améliore considérablement la qualité des solutions mais s'accompagne d'un coût de manipulation intrusive du code de la méthode des éléments finis. L'approche non-intrusive demeure une excellente alternative en termes de mise en œuvre et de qualité des solutions. L'approche développée a permis de surmonter certains des problèmes très connus qui se posent lors de l'utilisation de méta-modèles pour l'optimisation.

Dans le dernier chapitre, nous avons présenté une démarche complète pour effectuer une analyse de fiabilité de la machine électrique ; nous avons commencé par la présentation de la machine électrique. Ensuite, nous avons présenté les différentes mesures métrologiques qui ont été effectuées pour caractériser la variabilité de sa forme. Puis, nous avons présenté un modèle éléments finis paramétrique de la machine et nous avons exposé comment la mesure peut être prise en compte dans le modèle numérique.

L'évaluation numérique de l'impact des variables géométriques sur les performances de la machine a été réalisée en utilisant une approche de méta-modèle pour réduire le coût de calcul. La variabilité des performances a ensuite été caractérisée, et les variables qui influencent cette variabilité ont également été identifiées.

En résumé, les objectifs de cette thèse ont été atteints. Une comparaison des approches appropriées pour l'optimisation des machines électriques a été effectuée. Les avantages et les inconvénients de chaque méthode ont été mis en évidence.

L'intégration de l'analyse de fiabilité dans la phase de conception a été évaluée sur un cas test réel (une machine à griffes). L'impact du processus de fabrication sur la géométrie de la machine a été étudié, et une analyse quantitative a été réalisée.

# Contents

<b>Introduction</b>	<b>1</b>
<b>1 Literature Review</b>	<b>7</b>
1.1 Deterministic Optimization . . . . .	7
1.1.1 Optimization problems formulation . . . . .	8
1.1.2 Optimization algorithms . . . . .	11
1.1.3 Discussion . . . . .	25
1.2 Uncertainty . . . . .	26
1.2.1 Uncertainty propagation . . . . .	28
1.2.2 Optimization under uncertainty . . . . .	31
1.2.3 Discussions . . . . .	33
1.3 Electrical machines . . . . .	33
1.3.1 Physical model . . . . .	34
1.3.2 Mathematical model . . . . .	36
1.3.3 Numerical model . . . . .	39
1.3.4 MagFEM - 2D magnetostatic FE code . . . . .	43
1.3.5 Discussion . . . . .	48
1.4 Chapter Summary . . . . .	49
<b>2 Non-intrusive approach</b>	<b>51</b>
2.1 Meta-model design optimization MDO . . . . .	51
2.1.1 Design of experiment DoE . . . . .	53
2.1.2 Expensive evaluations . . . . .	55
2.1.3 Fitting Kriging . . . . .	55
2.1.4 Finding new samples . . . . .	67
2.1.5 Stopping criteria . . . . .	74
2.1.6 Analytical examples . . . . .	75
2.1.7 Results assessment . . . . .	76

---

2.1.8	Discussion . . . . .	80
2.2	Branch and Bound assisted by Meta-models (B2M2) . . . . .	83
2.2.1	Proposed algorithm . . . . .	83
2.2.2	Initialization . . . . .	83
2.2.3	Branching . . . . .	85
2.2.4	Meta-modelling . . . . .	85
2.2.5	Bounding . . . . .	86
2.2.6	Elimination . . . . .	87
2.2.7	Selection . . . . .	87
2.2.8	Termination . . . . .	87
2.2.9	Analytic Examples . . . . .	88
2.2.10	Extension to multi-objective optimization . . . . .	89
2.2.11	Extension to optimization under uncertainty . . . . .	90
2.3	Chapter Summary . . . . .	90
<b>3</b>	<b>Intrusive approach</b>	<b>93</b>
3.1	How to compute the gradient? . . . . .	95
3.1.1	finite-difference . . . . .	95
3.1.2	Mesh-morphing . . . . .	97
3.1.3	Discussion . . . . .	99
3.2	Explanatory example . . . . .	99
3.2.1	Derivation of gradient . . . . .	101
3.2.2	The adjoint variable method . . . . .	102
3.3	Adjoint variable method for FEA . . . . .	104
3.3.1	Finite element model . . . . .	104
3.3.2	Derivatives for state variables . . . . .	105
3.3.3	Derivatives for design variables . . . . .	106
3.3.4	Discussion . . . . .	111
3.4	Examples . . . . .	112
3.4.1	Solenoid model . . . . .	112
3.4.2	TEAM workshop 22 . . . . .	113
3.5	Chapter Summary . . . . .	114
<b>4</b>	<b>Applications and benchmarking</b>	<b>115</b>
4.1	Test cases . . . . .	115
4.1.1	TEAM Workshop Problem 22 . . . . .	115

4.1.2	TEAM Workshop Problem 25 . . . . .	119
4.2	Algorithms settings . . . . .	121
4.2.1	B2M2 algorithm . . . . .	121
4.2.2	SQP algorithm assisted by adjoint variable method . . . . .	122
4.2.3	DIRECT . . . . .	122
4.2.4	Genetic algorithm . . . . .	123
4.3	Results and comparison . . . . .	123
4.3.1	Comparison protocol . . . . .	123
4.3.2	TEAM Workshop Problem 22 . . . . .	124
4.3.3	TEAM Workshop Problem 25 . . . . .	125
4.4	Chapter Summary . . . . .	126
<b>5</b>	<b>Claw-pole machine</b>	<b>129</b>
5.1	Electrical machine . . . . .	130
5.1.1	Structure of the Claw-pole machine . . . . .	130
5.1.2	Operating mode and electrical characteristic . . . . .	133
5.1.3	Variability of output current in claw-pole machine . . . . .	134
5.1.4	Impact of manufacturing process . . . . .	135
5.1.5	Summary . . . . .	136
5.2	Metrology . . . . .	136
5.2.1	Methodology . . . . .	138
5.2.2	Summary . . . . .	143
5.3	Raw data and variability modelling . . . . .	143
5.3.1	Stator . . . . .	143
5.3.2	Rotor . . . . .	145
5.3.3	Virtual assembly . . . . .	150
5.3.4	Variability modelling . . . . .	150
5.3.5	Summary . . . . .	151
5.4	FEA model . . . . .	154
5.4.1	Geometry . . . . .	154
5.4.2	Material properties . . . . .	155
5.4.3	Electrical circuit . . . . .	155
5.4.4	Model validity . . . . .	155
5.4.5	Variability parameters . . . . .	156
5.4.6	Summary . . . . .	157



5.5	Uncertainty propagation . . . . .	158
5.5.1	Sampling . . . . .	158
5.5.2	Meta-modelling . . . . .	159
5.5.3	Statistical inference . . . . .	159
5.5.4	Sensitivity analysis . . . . .	159
5.6	Chapter Summary . . . . .	160
<b>Concluding remarks</b>		<b>163</b>
<b>A Shape sensitivities</b>		<b>177</b>
A.1	Rectangle . . . . .	177
A.2	Circle . . . . .	178
A.3	Other shapes . . . . .	178
<b>B FEM formulation in cylindrical coordinates</b>		<b>179</b>
B.1	General equations . . . . .	179
B.2	Variational Formulation . . . . .	180
<b>C Adjoint equation derivation for TEAM Workshop problem 25</b>		<b>183</b>
C.1	Gradient equations . . . . .	183
C.2	TEAM Workshop problem 25 . . . . .	183
C.2.1	Design variables . . . . .	184
C.2.2	State equations . . . . .	185
C.2.3	Quantity of interest . . . . .	187

# List of Figures

1	Claw-pole alternator . . . . .	3
2	Computational time versus the number of elements . . . . .	4
1.1	Optimization process . . . . .	8
1.2	Example of triangulation . . . . .	41
1.3	Reference element concept . . . . .	42
1.4	Conventional structure of a finite element code . . . . .	44
1.5	Geometry of the inductor . . . . .	44
1.6	Modelled geometry and studied domain of the inductor . . . . .	45
1.7	Identifiers of geometry . . . . .	45
1.8	Coarse mesh (left), Fine mesh (right) . . . . .	46
1.9	Magnetic flux density in the studied domain . . . . .	47
1.10	Example of extreme re-meshing errors. . . . .	48
2.1	Flowchart of Kriging meta-model based optimization. . . . .	52
2.2	Test function . . . . .	53
2.3	Two variables Full-factorial DoE . . . . .	54
2.4	Latin hypercube DoE . . . . .	55
2.5	Sampled Test function . . . . .	55
2.6	The evolution of Kriging research . . . . .	56
2.7	Exponential correlation function . . . . .	57
2.8	Kriging surrogate model . . . . .	61
2.9	Evolution of CLLF with respect to $\theta_1$ . . . . .	62
2.10	Impact of the CLLF on the Kriging model . . . . .	63
2.11	Evolution of CLLF with respect to $\theta_1$ for config. 2 . . . . .	64
2.12	Kriging surrogate model with optimized CLLF . . . . .	64
2.13	Iterations of EGO . . . . .	69
2.14	Constrained test problem . . . . .	71

---

2.15	Iterations of constrained EGO . . . . .	72
2.16	Two-dimensional Rosenbrock function . . . . .	75
2.17	Constrained function . . . . .	76
2.18	Evolution of the expected improvement based algorithm . . . . .	78
2.19	Evolution of the condition number as function of iteration . . . . .	81
2.20	Cost of fitting the Kriging model for Example 1 . . . . .	82
2.21	Flowchart of B&B Kriging . . . . .	84
2.22	Space exploration by the algorithm . . . . .	89
3.1	Infinite solenoid model . . . . .	93
3.2	The FEA of the solenoid . . . . .	95
3.3	Relative error of derivative computed by finite-difference . . . . .	96
3.4	Same geometry, different meshes . . . . .	97
3.5	Same mesh topology, different geometries . . . . .	97
3.6	Relative error of derivative computed by mesh-morphing . . . . .	98
3.7	Electrical circuit . . . . .	100
3.8	Model of the circuit . . . . .	100
3.9	(left) The reference element, (right) an arbitrary element. . . . .	107
3.10	Edges and faces numbering . . . . .	108
3.11	Flowchart of a modular adjoint implementation . . . . .	111
4.1	SMES Device . . . . .	116
4.2	Modelled geometry of the SMES . . . . .	117
4.3	The FEA of the SMES device . . . . .	118
4.4	Model of die press with electromagnet of TEAM problem 25 . . . . .	119
4.5	Modelled geometry of the die press . . . . .	120
4.6	The FEA of the die press . . . . .	121
5.1	Position of claw-pole alternator . . . . .	130
5.2	Detailed view of the claw-pole alternator . . . . .	131
5.3	Stator of claw-pole machine . . . . .	131
5.4	Rotor of claw-pole machine . . . . .	132
5.5	The brackets of claw-pole machine . . . . .	133
5.6	Electrical diagram of claw-pole machine . . . . .	133
5.7	Measured output current as a function of rotation speed . . . . .	134
5.8	Variation of measured output current on 30 machines . . . . .	135

5.9	The Coordinate Measuring Machine . . . . .	137
5.10	The measurement procedure using the CMM . . . . .	137
5.11	Claw-pole machine disassembling . . . . .	138
5.12	Claw-pole machine disassembled . . . . .	139
5.13	Reference frame of the stator . . . . .	139
5.14	Measurement points of the inner surface of the stator . . . . .	140
5.15	Reference frame of the rotor . . . . .	140
5.16	Measurement points of the outer surface of the rotor . . . . .	141
5.17	Virtual assembly of the stator and the bracket . . . . .	142
5.18	Virtual assembly of the stator and the rotor . . . . .	142
5.19	Deviation to nominal of the internal surface of the stator in <i>mm</i> . . . . .	144
5.20	Model of deviation to nominal of the internal surface of the stator . . . . .	145
5.21	Synthesis of the measurements of the internal surface of the stator . . . . .	146
5.22	Deviation from nominal of the external surface of the rotor . . . . .	147
5.23	Synthesis of the measurements of the outer surface of the rotor . . . . .	149
5.24	Virtual assembly of the stator and the rotor . . . . .	150
5.25	Synthesis of the measurements of of virtual assembling (Part 1) . . . . .	151
5.26	Synthesis of the measurements of of virtual assembling (Part 2) . . . . .	152
5.27	Synthesis of the all measurements on claw-pole machines . . . . .	153
5.28	Technical drawing of the electrical machines . . . . .	154
5.29	Some parameters of the geometry . . . . .	155
5.30	$B(H)$ curves of the machine materials . . . . .	155
5.31	Electrical circuit of the claw-pole machine. . . . .	156
5.32	Radial deformation of stator teeth . . . . .	157
5.33	Eccentricity module of JMAG software . . . . .	158
5.34	Variability of the output current resulted from variability on geometry . . . . .	159
5.35	Sobol indices of the variables . . . . .	160
C.1	Molds of the Die press (design region) . . . . .	184



# List of Tables

2.1	Variable and responses scaling . . . . .	60
2.2	Impact of adding samples on the CLLF . . . . .	63
2.3	Performances of EGO on the multidimensional Rosenbrock problem .	77
2.4	Performances of EGO on the constrained problem 2.17 . . . . .	79
2.5	Performances of B2M2 on the analytic examples . . . . .	88
3.1	Comparison of analytic and FEA quantities . . . . .	94
3.2	Comparison of analytic and and adjoint variable gradients . . . . .	112
3.3	Gradient Comparison for TEAM workshop 22 . . . . .	113
4.1	Bounds of design variables for TEAM problem 22 . . . . .	117
4.2	Bounds of design variables for TEAM problem 25 . . . . .	121
4.3	TEAM Workshop problem 22 optimization results . . . . .	124
4.4	TEAM Workshop problem 25 optimization results . . . . .	126
5.1	Variables modeling the stator shape for some machines . . . . .	145
5.2	Comparison of simulation result with respect to experimental one . .	156



# Introduction

## The societal context

Electrification is gaining momentum. If the early twentieth century saw electricity revolutionized our daily lives, especially in static applications, the beginning of the twenty-first century sees electricity also profoundly modify all embedded applications. The Electricity Fairy is a very convenient way of transporting and converting energy. The tremendous advances in power electronics has led to new solutions for the vast majority of energy conversion systems. Not to mention, for example, the significant steps are taken with all-electric cars, *e.g.* Tesla, or battery-powered aircrafts, *e.g.* Airbus E-FAN, making them accessible for mass usage. Nevertheless, they are still currently limited by the storage and charging capacities of much less energy than oil. Electricity spreads actively inside systems throughout the energy conversion chain of embedded applications.

The car is becoming hybrid; an electric traction chain assists the internal combustion engine. Naval propulsion is increasingly done by pods containing electrical machines powered by diesel generators. Hydraulic and pneumatic energies are being progressively replaced in aircraft by electric power, causing high expansion of the power of the alternators driven by the engines of the plane (1.4MVA for the Boeing 787 against 300kVA for A330).

The gains are significant in terms of simplicity of installation, modularity, maintenance, and source of failure (an electric cable never leaks). Electric actuators multiply for the sole purpose of comfort: adjusting the electric seats, the electric window of a car, or motorization of rear-view mirrors.

In all these more electrical systems, the ends of the electromechanical chain are often electrical machines; either an actuator that generates electricity while driven by an internal combustion engine or an actuator to generate mechanical energy that is used to achieve the final function. These electrical machines are mostly rotating machines. In this context, their improvement is a vital issue in terms of weight, bulk, efficiency, manufacturing cost, reliability and recyclability.

Furthermore, air pollution has become an urgent problem of environmental sanitation, affecting both developed and developing countries. Increasing amounts of potentially harmful gases and particles are released into the atmosphere, damaging



human health and the environment. They also undermine the long-term resources needed for the sustainable development of our planet.

Studies in France and around the world have shown that the share of transport in total emissions of air pollutants has increased steadily over the last few years. Some countries have taken initiatives to promote the use of clean or low-emission vehicles.

The reduction of emissions requires a significant effort in improving the efficiency of all energy consumers in vehicles and especially auxiliaries such as air conditioning and alternator. The improvement of the performances of electrical devices in cars, which this thesis examines, is in line with this spirit.

## The industrial context

The automotive market is very competitive, which drives automakers to improve their products continually. As a result, electricity and electronics are of great importance in today's automobiles. The electrification of the passenger compartment enhances the comfort and the pleasure of driving and riding the vehicles. Besides, cars are becoming safer thanks to new electrical devices that improve active safety (anti-lock braking system) and passive safety (airbags). Also, the replacement of mechanical or hydraulic devices, such as power steering, by electric actuators can reduce costs and be more comfortable to control.

The increasing demand for electricity in cars is due to the use of these electronic devices. Thus requiring a more significant on-board production of this energy from the alternator on which the constraints necessary for the functioning of the electrical systems, mainly constraints of efficiency, compactness and low-cost manufacturing, will be exercised.

In this context, the company Valeo has performed several studies on claw-pole alternators (Figure 1). The overall goal of these studies was to design and optimize the machine to improve specific performances (Power, noise, ...) while ensuring a low cost and compactness. For the designer to reach these goals, it is necessary to have models that address most of the machine's aspects (electrical, thermal, magnetic, mechanical). Nevertheless, these models are generally expensive in computational time, requiring the solution of large system of algebraic equations. Thus, optimizing the machine with these models is cumbersome and takes a tremendous amount of time.

Additionally, most of these studies did not include uncertainties introduced by the manufacturing process. In the design phase, the variability of the electrical machine geometry and material properties has not been taken into account. This variability can impact electrical machine performances and can lead to non-compliant products. Thus, optimization and uncertainty quantification methodologies are of

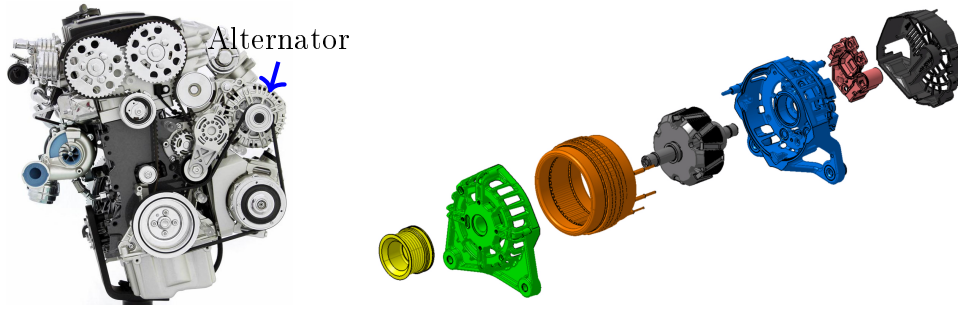


Figure 1: Position of Claw-pole alternator and detailed view.

interest for Valeo to improve the design process by including the manufacturing process constraints.

## The complexity of the design

With the ever-growing complexity of products, modelling and simulation allow us to create an all-digital prototype, to understand and optimize the critical performances in order to ensure that the product will fulfill the specifications correctly during its life cycle. Therefore, the usage of high-fidelity models has become mandatory.

A model is regarded as representative of a system or a phenomenon. It is a fictitious system, which assembles equations associated with particular physical hypotheses to draw specific conclusions. A model is oriented; from input variables, it provides a result. To estimate the effect of the causes, in the same way, the inverse model is the one that reverses this causality.

A model is ideal for design when its inversion is unique; if a single cause produces the desired effect. Indeed, the reversal makes the development faster and less cumbersome since the work is done only once. Nevertheless, the models are generally not invertible, and there are many degrees of freedom that allow meeting the same specifications. Optimization tools can be used to improve a design whose model is not invertible.

Using optimization tools early in the design phase of the development cycle dramatically increases the device potential for improvement. If the designer uses the optimization tools downstream in the development cycle, the main choices will already be made, and the contribution of the optimization will remain moderate. This approach is still not widespread at Valeo, where optimization is used later in the development cycle. Thus, there is a real need for design methodologies to deal with the complexity of the models.

## The computational cost

To be able to model the physics related to the studied phenomena, numerical methods are used; one of the famous and the most robust ones is the Finite Element Method, *i.e.* FEM. It is a method that enables us to determine an approximate solution on a spatial domain by calculating a field (of scalars, vectors) corresponding to the solution of the given equation.

The method consists of splitting the space domain into small elements, also called meshing, and looking for a simplified formulation of the problem on each element, transforming the system of any equations into a system of equations. The latter is represented by a matrix. The matrices for all the elements are assembled, forming a large matrix; the resolution of this global system gives the approximate solution to the problem.

However, the time needed for a FEM simulation depends on the number of small elements. Figure 2 shows the evolution of the computational time versus the number of elements that compose a studied domain. It is worth noting that the data are retrieved by running a finite element code that we have implemented, simulating an electromagnetic device for different element sizes. We can see that the computational time evolves polynomially to the number of elements.

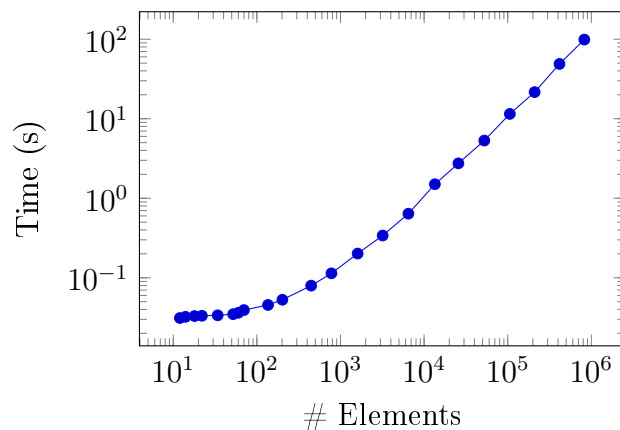


Figure 2: Computational time versus the number of elements

The claw-pole machine has a very complex geometry; consequently, a significant number of elements are needed (around two million) to be able to model its behaviour finely. Therefore, the time needed for the simulations is considerable. The existence of nonlinearity or a dynamic behaviour in the physical phenomena can multiply the computational time considerably. The designer should take this feature into account since it will significantly impact the duration of the design phase by using methodologies that are well-suited.

## The manufacturing process

The manufacturing processes, because of their imperfections, lead to dispersions on the characteristics of the products. In the case of electrical machines, these imperfections can impact its geometry and properties.

These dispersions induce "variabilities" on quantities of interest and the performances for the product. For example, it becomes crucial for an automobile alternator to study the dispersion influence on the output current and efficiency. This kind of study aims to ensure that the product fulfills the specifications contracted with the customer.

The classical approach based on a deterministic model (in the sense that the model entries are entirely known) to address this problem is no longer sufficient. It is necessary to consider the quantities of inputs as uncertain quantities and to place oneself within the conceptual framework of uncertainties.

The uncertainty in the manufacturing process can be reduced by improving the production lines; however, this will induce a high cost. One approach consists of knowing the uncertainty on the geometry on the material process and predict the impact of the uncertainty on the performances of the product. This approach will enable us to control the influence of the variability of the geometry or the material characteristics to see if the design variables respect the specifications and if their variability is not too high. If it is the case, this approach will also enable to determine which parameters on the geometry or the material behaviour have the most impact on the variability of performances, therefore to act on these parameters in order to have a more robust product or to act on the manufacturing process in order to reduce the variability of these influential parameters.

## Thesis organization

The remainder of this dissertation is organized as follows.

**Chapter 1** reviews optimization algorithms and the finite element method to put the contributions of this thesis into perspective. Not all optimization algorithms are adapted for finite element models because of the computational cost of these models. We present the main categories of optimization algorithms and their extensions to the context of optimization under uncertainty. A brief presentation electromagnetic modelling of the electrical machines using the finite element method is also presented. Two approaches adapted to the subject of the thesis are selected to be investigated.

**Chapter 2** deals with a non-intrusive approach where the finite element model is considered as a black-box, and a surrogate model is fit by evaluating the expensive model at few points, then, use the cheap surrogate model for optimization and uncertainty quantification. We present a thorough introduction of optimization

using meta-models and some of the challenges of fitting "good" meta-models and how to assess their quality, then we propose a novel strategy for exploiting Kriging models.

**Chapter 3** examines an intrusive approach to compute the derivatives from a finite element model. In this approach, most of the work is done on the model rather than the algorithm. The derivatives are used by the most efficient optimization algorithms and uncertainty quantification methods. We present the adjoint state variable method and how it can be derived from a finite element code. The adjoint variable method enables to compute the derivatives efficiently from the finite element code.

**Chapter 4** is dedicated to numerical tests of the developed approaches in the context of optimization. For this task, we address two well-known benchmarks treated by electromagnetic community researchers. They are ones of the TEAM (Testing Electromagnetic Analysis Methods) workshops [1] [2]. We present a comparison between the investigated approaches and other approaches for optimizing electromagnetic devices.

**Chapter 5** presents the impact of manufacturing process imperfections on the performance of the claw-pole machine. We mainly study geometry imperfections. In this chapter, we present the claw-pole machine and its operating mode then introduce the metrology procedure adopted for the measurement of the electrical machine. These measurements are analyzed, and the variability of the geometry of the electrical machine induced by the manufacturing process is modelled. After, a parametrized finite element model of the machine is presented. Next, the uncertainty propagation using the variability characterized and the model of the machine is conducted employing Monte Carlo simulation assisted by Kriging meta-model.

The final conclusions and considerations for future work are given in the last chapter.

# Chapter 1

## Literature Review

As described in the introduction, one can use optimization tools and models to improve electrical machines' design. One can categorize their use by the type and how they are integrated into the design phase. This chapter begins by briefly describing several deterministic optimization algorithms. The second section reviews how to incorporate uncertainty in an optimization process. The third section focuses on the modelling of electrical machines modelled by finite element method. The review of optimization tools and finite element method is presented so that the contributions of this thesis are put into perspective.

### 1.1 Deterministic Optimization

Optimization and especially numerical optimization have had a significant boost in the last fifty years with computers' advent. Optimization is a design tool that helps designers identify the optimal solutions from several possible choices or an infinite set of choices. Optimization is increasingly applied in industry since it provides engineers with a reasonable and flexible means to identify optimal designs before physical deployment. It is often the last step of computational analysis. First, one should start by studying the physical phenomenon, describing it by equations, studying these equations, and having shown that they can be solved with a computer. Next, identifying the objective, the variables, and the constraints that should be satisfied. Then, optimize the system by adjusting the variables to change the solution in the desired direction.

**An objective** is a measure of the performance of the system that we want to minimize or to maximize. In electrical machines, we may want to maximize the performances or minimize production cost, or both.

**The variables** are the parameters of the system for which we want to find values. The variables may be the parameters that define the geometry or material properties of the electrical machine.

**The constraints** are the functions that define the allowable values for the variables and the requirements that should be satisfied for the system to be acceptable.

Mathematically, one can express any optimization problem as

$$\begin{aligned} \min_p f(p) \\ \text{s.t. } g_i(p) \leq 0, \quad i = 1 \dots n_c \end{aligned} \quad (1.1)$$

where  $p = (p_1, p_2, \dots, p_{n_v})$  are the variables to be determined,  $f(p)$  is the objective function to be minimized and  $g_i(p)$  are  $n_c$  constraints to be satisfied. By convention, the standard form defines a minimization problem. A maximization problem can be treated by negating the objective function. Likewise for the constraints, in the case of

- $g(p) \geq 0$  then, it is transformed to  $-g(p) \leq 0$
- $g(p) \leq r$  where  $r$  is a real number, then, it becomes  $g(p) - r \leq 0$
- $g(p) = 0$  can be transformed to two constraints  $g(p) \leq 0$  and  $g(p) \geq 0$ . However, some algorithms handle equality constraints naturally.

the problem can be written in the standard form. The optimization problem should be carefully defined based on the performances required and the corresponding constraints that are given by a model.

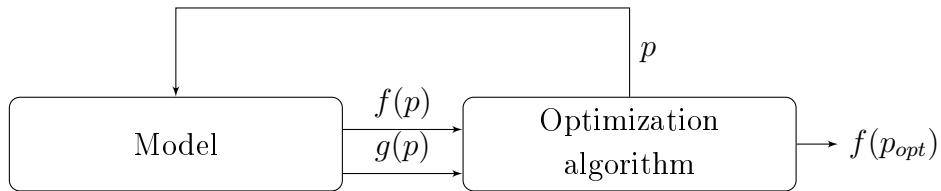


Figure 1.1: General process of optimization.

Figure 1.1 shows the general framework of an optimization process; an optimization algorithm predicts where to evaluate the model to solve the optimization problem. It is an iterative process of model evaluation; The algorithm repeatedly evaluates the model at a variable and gets the values of the objective function and the constraints at this variable. Then, the optimization algorithm uses this information to approach the optimal solution  $p_{opt}$ .

In this section, we present a brief overview of the formulation of optimization problems and the algorithms that enable us to solve them with insight into the pros and cons of each of them.

### 1.1.1 Optimization problems formulation

As noted before, an essential step in the optimization process is classifying the optimization problem, since algorithms for solving optimization problems are dedicated

to a particular type of problems. Indeed, most algorithms developed are designed to solve specific types of problems and are not very efficient for all different types. Here, we provide insights on how optimization problems are classified.

### **A One vs many objectives**

A single objective function defines single-objective problems. Multi-objective problems exist when a compromise is to be sought between several conflicting objectives. For example, optimizing an electrical machine might involve minimizing bulk while maximizing performances. In practice, problems with multiple objectives may be reformulated as single-objective problems by either applying a weighting coefficient on the different objectives or by replacing some of the objectives with constraints. Nevertheless, there exist some dedicated algorithms that solve multi-objective problems.

### **B Unconstrained vs constrained problems**

It is vital to identify problems where constraints exist on the variables. Constrained optimization problems result from applications in which there are constraints on the variables. The constraints on the variables can vary from simple bounds or systems of equalities and inequalities to a more complex nonlinear relationship among the variables. Constrained optimization problems can be reformulated and expressed as unconstrained ones by adding a penalty term in the objective function. Naturally, problems with constraints are more complicated to solve and use dedicated algorithms.

### **C Continuous vs discrete problems**

In some cases, the variables are discrete, most often in the form of integers or binaries. The optimization problem is said to be discrete. On the contrary, in continuous optimization problems, variables can take any real value. Continuous optimization problems are usually easier to solve. An optimization problem mixing continuous variables and discrete variables is called a mixed-integer optimization problem.

Continuous optimization problems are more accessible to solve than discrete ones; the values of the objective function and constraint functions at a point  $x$  can be used to deduce information about points in a neighbourhood of  $x$ . However, advancements in computing technology have dramatically increased the size and complexity of discrete optimization problems that can be solved efficiently. Continuous optimization algorithms are necessary for discrete optimization because many discrete optimization algorithms generate a sequence of continuous subproblems.



## D Linear vs nonlinear problems

Linear optimization deals with optimization problems whose data, *i.e.* objective function and constraints, are linear functions. On the opposite side, nonlinear optimization copes mainly with optimization problems where the objective function or the constraints or both are nonlinear functions. Nonlinear problems are a complicated matter in the field of optimization because it is challenging to converge to the best solution. Quadratic optimization is a particular case of nonlinear optimization where the objective function is quadratic, and constraints are linear functions. Quadratic optimization is widely used to solve nonlinear problems by solving a sequence of quadratic problems.

## E Explicit vs black-box models

Black-box optimization models are models for which the functions defining the objective and constraints are complex, difficult to evaluate, or their analytical expressions are unknown. For most industrial problems, this is a sophisticated computer program that is often inaccessible. They are exploited by analyzing the responses to given inputs; they are considered as black-boxes without any hypothesis on the properties of the functions. The classical mathematical methods based on rigorous hypotheses such as continuity, differentiability, convexity are not straightforwardly applicable in black-box optimization.

Thus, several algorithms designed to solve this type of problems have been developed that do not require any hypothesis on the function. The use of black-box models, as well as the application of suited algorithms, made it possible to solve challenging optimization problems. Nevertheless, there still some challenging aspects that make black-box optimization a thriving research area.

## F Intuitions on the classifications of optimization problems

It is worth noting that there are no problems that are easier to solve than others; for example, there are some continuous linear problems that are more difficult to solve than nonlinear problems.

Nevertheless, there are some problems for which we can guarantee the optimality of the solution, *e.g.* quadratic problems; that is to say, the solution found is a global one. Generally, we approach hard problems with "many" of these guaranteed optimality problems, such as approaching nonlinear problems with a series of quadratic problems. These approximations generally lead to acceptable solutions.

### 1.1.2 Optimization algorithms

The objective of this section is to present different types of optimization algorithms. We will present five classes of algorithms: derivative-based, heuristic-based, population-based algorithms, exact algorithms and approximation-based algorithms. In every class, we can find many algorithms that exploit different ideas.

To simplify the presentation of algorithms, we focus on the presentation of single objective unconstrained optimization algorithms, as shown in equation (1.2); then, we present possible extensions to other types of problems such as constrained, multi-objective and black-box problems.

$$\min_p f(p) \tag{1.2}$$

#### A Derivative-based algorithms

As the name suggests, this class of algorithms needs the derivatives of the objective function, mainly the first derivative, *i.e.* gradient  $\nabla f$ , and sometimes even the second derivative, *i.e.* hessian  $H$ .

$$\nabla f(p) = \left( \frac{\partial f}{\partial p_1}, \frac{\partial f}{\partial p_2}, \dots, \frac{\partial f}{\partial p_{n_v}} \right) \tag{1.3}$$

$$H(p) = [H_{i,j}(p)] = \left[ \frac{\partial^2 f}{\partial p_i \partial p_j} \right]_{ij} \tag{1.4}$$

It should be noted that whenever the derivatives are available and are easy to compute, a derivative-based method should be preferred. A typical algorithm gradually improves the estimate of the solution step by step. It needs to decide at each iteration the location of the new estimate. However, it has only very local information about the landscape of the objective function – the current function value and the derivatives. The derivatives express the slope of the function at a point. Hence, it seems natural that the derivative information is used to define a search direction, and the algorithm searches along this direction for a new (better) point. Derivative-based algorithms are summarized in the algorithm below.

The designer chooses a starting point  $p^0$  at iteration  $k = 0$ , the algorithm exploits the derivatives to find the "best" direction  $d^0$  and the step  $\alpha^0$ . Consequently, the next iterate  $p^1$  is found, and the algorithm repeats the same process until convergence.

The main differences among derivative-based algorithms are :

- How to look for the direction  $d^k$ ?
- What is the step length  $\alpha^k$  to take in direction  $d^k$ ?

**Algorithm 1:** Derivative-based algorithms

---

Choose  $p_0$  and initialize  $k = 0$ ;  
**while** *Not converged* **do**  
    Find a descent direction  $d^k$ ;  
    Determine a step length  $\alpha^k$  that reduces the function  $f$  sufficiently;  
    Compute the new iterate  $p^{k+1} = p^k + \alpha^k d^k$ ;  
    Set  $k = k + 1$ ;  
**end**

---

- What are the convergence criteria that enable to stop the algorithm?

The direction  $d^k$  has to be a descent direction, that is to say, the function  $f$  is decreasing in the direction  $d^k$ , mathematically, it is expressed as

$$f(p^k + \varepsilon d^k) < f(p^k)$$

for a sufficiently small  $\varepsilon$  or equivalently  $d^{kT} \nabla f(p^k) < 0$ .

The step length  $\alpha^k$  is also a key ingredient for the success of the algorithm. In the literature, it is referred as line search. The line search consists of determining the step  $\alpha^k$  to be taken along the descent direction  $d^k$ . It is often chosen in order to verify the following two conditions:

1. the function  $f$  must decrease sufficiently along the descent direction;
2. the step  $\alpha^k$  should not be too small in order to avoid slowing down the convergence.

To find the optimal step length (neither too small nor too big) an **exact line search** can be performed by minimizing the following problem.

$$\min_{\alpha \geq 0} f(p^k + \alpha d^k) \tag{1.5}$$

However, this type of line search generally requires a lot of computation time without significantly improving the convergence of the algorithm. **The inexact line search** provides an efficient way of computing an acceptable step length  $\alpha^k$  that reduces the objective function sufficiently rather than minimizing the objective function exactly [3]. One of the most famous inexact line search methods is the Wolfe conditions [4].

A step length  $\alpha^k$  is said to satisfy the Armijo and Wolfe conditions if the following two inequalities hold:

1.  $f(p^k + \alpha^k d^k) \leq f(p^k) + c_1 \alpha^k d^{kT} \nabla f(p^k)$

$$2. -d^{kT} \nabla f(p^k + \alpha^k d^k) \leq -c_2 d^{kT} \nabla f(p^k)$$

with  $0 < c_1 < c_2 < 1$ .  $c_1$  is usually chosen to be quite small, *e.g.*  $10^{-4}$ , while  $c_2$  is much larger, *e.g.* 0.9.

Both conditions can be interpreted as providing an upper and lower bound on the admissible step length values. Iterative algorithms are used to solve the inequalities; usually, the first value of  $\alpha^k$  that verifies both inequalities is chosen.

All line search methods are carried out along descent directions. Different ways to determine these directions are discussed: the gradient descent, the conjugate gradient method, Newton's method and its generalization, quasi-Newton methods.

### Gradient descent

This method uses the opposite of the function gradient as a descent direction

$$d^k = -\nabla f(p^k) \tag{1.6}$$

This direction is obviously a descent direction because we have

$$d^{kT} \nabla f(p^k) = -\|\nabla f(p^k)\|^2 < 0$$

More importantly, it is the steepest descent direction at position  $p^k$ .

The gradient descent method can take many iterations to compute a local minimum, and the line search can also be very long. Other methods, such as conjugate gradient, can be good alternatives. Generally, such methods converge in fewer iterations, but the cost of each iteration is higher [5].

### Conjugate gradient

Conjugate gradient construct the descent direction by adding the opposite of the gradient to the previous descent direction multiplied by a real scalar  $\beta^k$

$$d^k = -\nabla f(p^k) + \beta^k d^{k-1} \tag{1.7}$$

At the first iteration, there is no previous direction then the algorithm takes simply the opposite of the gradient [6]. Several methods exist for calculating the term  $\beta^{k+1}$ , among them the Polack-Ribière method emerges [7].

$$\beta^k = \frac{\nabla f(p^k)^T (\nabla f(p^k) - \nabla f(p^{k-1}))}{\nabla f(p^{k-1})^T \nabla f(p^{k-1})} \tag{1.8}$$

An additional issue arises for this method to ensure that  $d^k$  is a descent direction, *i.e.*  $d^{kT} \nabla f(p^k) < 0$

$$d^{kT} \nabla f(p^k) = -\|\nabla f(p^k)\|^2 + \beta d^{k-1T} \nabla f(p^k) \tag{1.9}$$

Equation (1.9) is negative if  $d^{k-1T} \nabla f(p^k) = 0$ . To satisfy this condition, an **exact line search is necessary**. Thus, the method becomes costly.

**Newton method**

Newton method constructs the descent direction based on a second-order Taylor approximation of around the iterates [8]. The second Taylor expansion of  $f$  around  $p^k$

$$f(p^k + d^k) \approx f(p^k) + d^{kT} \nabla f(p^k) + \frac{1}{2} d^{kT} H(p^k) d^k \quad (1.10)$$

The minimum of the approximation can be found by setting the first derivative to zero.

$$0 \approx \nabla f(p^k) + H(p^k) d^k \quad (1.11)$$

Then, the Newton descent direction can be deduced

$$d^k = -H(p^k)^{-1} \nabla f(p^k) \quad (1.12)$$

This direction is a descent direction if  $H(p^k)$  is a definite positive matrix.

Newton's method can guarantee convergence to local optima under mild assumptions, *e.g.* convexity, positive definiteness,  $\dots$ . For the case where the hessian  $H$  of the function is not positive definite, the algorithm may diverge. Thus, another category of algorithms was developed to cope with this drawback, *i.e.* quasi-Newton methods.

**Quasi-Newton method**

The quasi-newton method has the same iterates as Newton's method except for the hessian, which is not the actual hessian of the objective function but an approximation of it [9]. This method is efficient when the actual hessian is not definite positive, or it is not very easy to compute.

$$d^k = -B^{k-1} \nabla f(p^k) \quad (1.13)$$

where  $B^k$  is an approximation of the hessian. There are different methods to approximate the hessian the most famous ones is BFGS for Broyden - Fletcher - Goldfarb - Shanno [8]. The approximation is written as

$$B^{k+1} = B^k + \frac{y^k y^{kT}}{y^{kT} s^k} - \frac{B^k s^k s^{kT} B^k}{s^{kT} B^k s^k} \quad (1.14)$$

where  $y^k = \nabla f(p^{k+1}) - \nabla f(p^k)$  and  $s^k = p^{k+1} - p^k$ . It has been proved that  $B^{k+1}$  is definite positive if  $B^k$  is of the same nature and  $y^{kT} s^k > 0$ . Generally,  $B^0$  is set equal to the identity matrix.

### Extension to constrained problems

In constrained optimization, the general aim is to transform the problem into an easier subproblem that can then be solved and used as the basis of an iterative process. The general idea relies on the solution of the Karush-Kuhn-Tucker (KKT) conditions [10][11]. The KKT enables the transformation of the constrained problem to an unconstrained one by including KKT multipliers (equivalent to the Lagrange multipliers for equality constraints).

To recall, the constrained problem is written as

$$\begin{aligned} \min_p f(p) \\ \text{s.t. } g_i(p) \leq 0, \quad i = 1 \dots n_c \end{aligned} \quad (1.15)$$

Using KKT, it can be transformed

$$\min_{p, \lambda} L(p, \lambda) = f(p) + \sum_{i=1}^{n_c} \lambda_i g_i(p) \quad (1.16)$$

where  $L$  is called the Lagrangian function and  $\lambda$  are the KKT multipliers.

The KKT equations are necessary conditions for the optimality of constrained optimization problems and stated as follows.

1. Stationarity  $\nabla L(p, \lambda) = 0$
2. Feasibility  $\lambda_i \geq 0$  and  $g_i(p) \leq 0$  for all  $i = 1, \dots, n_c$
3. Complementary slackness  $\lambda_i g_i(p) = 0$

The optimal solution should satisfy these conditions. These conditions are necessary but not always sufficient, and additional information is required. The necessary conditions are sufficient if the objective function and the constraints are convex functions. Thus, for solving complex nonlinear problems, many algorithms solve successive "simple" problems for which the KKT condition are sufficient to prove optimality. From which, we will detail sequential quadratic programming (SQP), one of the most robust derivative-based algorithms for nonlinearly constrained optimization problems.

**SQP** The SQP method can be viewed as a generalization of quasi-Newton's method for constrained optimization [12][13]. It finds a step away from the current point by minimizing a quadratic model of the problem.

$$\begin{aligned} \min_{d^k} f(p^k) + d^{kT} \nabla f(p^k) + \frac{1}{2} d^{kT} B^k d^k \\ \text{s.t. } g_i(p^k) + d^{kT} \nabla g_i(p^k) \leq 0, \quad i = 1 \dots n_c \end{aligned} \quad (1.17)$$

Different algorithms can solve this problem; one of the most used is the active-set. The active-set algorithm solves the KKT conditions using "guess and check" to find critical points. Guessing that each inequality constraint is inactive ( $g_i(p^k) + d^{kT} \nabla g_i(p^k) < 0$ ) is conventionally the first step. After solving the remaining system for  $d^k$ , feasibility can be checked. If any constraints are violated, they should be considered active ( $g_i(p^k) + d^{kT} \nabla g_i(p^k) = 0$ ) in the next iteration. If any multipliers  $\lambda_i$  are found to be negative, their constraints should be considered inactive in the next iteration.

The solution to the problem gives the descent direction for the iterate.

$$d^k = -B^{k-1} [\nabla f(p^k) + \sum_{i=1}^{n_c} \lambda_i \nabla g_i(p)] \quad (1.18)$$

The multipliers  $\lambda_i$  for inactive constraints are taken to be zero.

The choice of distance  $\alpha^k$  to move along the direction generated by the subproblem is not as evident as in the unconstrained case, where we simply choose a step length that approximately minimizes  $f$  along the search direction. For constrained problems, we would like to have the next iterate not only to decrease  $f$  but also to satisfy the constraints. Often these two aims conflict, so it is necessary to weigh their relative importance and define a merit or penalty function, which we can use to determine whether or not one point is better than another.

$$M(p^k) = f(p^k) + \sum_{i=1}^{n_c} r_i^k \max(0, g_i(p^k)) \quad (1.19)$$

where  $r_i^k$  are sufficiently positive parameters.

This merit function is used to find the step length  $\alpha^k$  to take in the descent direction  $d^k$  found in (1.18) and, thus, finding the next iterate.

$$p^{k+1} = p^k + \alpha^k d^k \quad (1.20)$$

There exist many algorithms for nonlinearly constrained problems such as interior-point and trust-region algorithms. There is not one general algorithm that will work effectively for every kind of nonlinear problems. We chose to detail SQP because it proved to be robust for the cases that we treated.

### Extension to multi-objective problems

The ideas swing around transforming a multi-objective problem into many single objective ones. We present two methods: weighted-sum and  $\epsilon$ -constraint.

- weighted-sum is the most used method; it transforms the objective functions into a weighted sum of them [14]. Then, to use a single objective optimization algorithm. Changing the weights can lead to different solutions (but not necessarily) that form the Pareto front. This approach is straightforward to conduct, but it cannot lead to a complete description of the Pareto front if the latter is not convex.
- $\epsilon$ -constraint method only considers one objective function while subjecting the others to constraints [15]. A Pareto front can be obtained by varying the parameters of the constraints.  $\epsilon$ -constraint method tackles the drawback of the weighted-sum approach.

### Extension to black-box problems

The gradient is a fundamental ingredient for derivative-based algorithms; nevertheless, this information is not always available when using black-box models. Finite difference (FD) can be used to estimate the derivatives using the formula below

$$\frac{\partial f}{\partial p_i}(p) \approx \frac{f(p_1, \dots, p_i + \varepsilon, \dots, p_{n_v}) - f(p_1, \dots, p_i, \dots, p_{n_v})}{\varepsilon} \quad (1.21)$$

where  $\varepsilon$  is the finite difference parameter. This equation is called the forward finite difference; other approximations also exist, such as backward FD and centred FD. They exploit the same idea as forward FD nevertheless centred FD may lead to better results than the others.

The choice of  $\varepsilon$  is essential for the success of the approximation. A very small value will lead to significant numerical roundoff errors, and high value will not lead to the correct approximation of the gradient. Furthermore, the objective function and/or the constraints can be noisy, which renders the derivatives' computation using finite difference not reliable.

### B Derivative-free algorithms

A heuristic algorithm is one that is intended to solve a problem in a faster and more efficient fashion than traditional methods by sacrificing optimality, accuracy or precision for speed. From these, we can cite Nelder-Mead simplex, Pattern search, DIRECT (DIviding RECTangle), ...

Engineers and scientists widely use heuristic algorithms. They are a class of optimization methods that are easy to implement, do not require derivatives and are often claimed to be robust to deal with problems with discontinuities or where the functions are noisy.

#### Nelder-Mead

The Nelder-Mead (NM) is a heuristic numerical method that exploits the concept



of a simplex [16]. A simplex can be a line segment on a line, a triangle on a plane, a tetrahedron in three-dimensional space, *etc.* Starting initially from such a simplex, the simplex undergoes simple transformations during iterations: it deforms, moves and progressively reduces until its vertices approach a point where the function is locally minimal. The full description is shown in the algorithm below.

---

**Algorithm 2:** Nelder-Mead algorithm

---

```
set the initial simplex ;
while Not terminated do
  order according to the values at the vertices:
     $f(p^1) \leq f(p^2) \leq \dots \leq f(p^{n_v+1})$ ;
  calculate  $p^o$  the centroid of all points except  $p^{n_v+1}$ ;
  compute reflected point  $p^r = p^o + \alpha(p^o - p^{n_v+1})$ ;
  if the reflected point is the best point,  $f(p^r) < f(p^1)$  then
    compute expanded point  $p^e = p^o + \beta(p^o - p^{n_v+1})$ ;
    if the expanded point is better than the reflected point,  $f(p^e) < f(p^r)$ 
      then
        replace the worst point  $p^{n_v+1}$  with the expanded point  $p^e$ ;
      else
        replace the worst point  $p^{n_v+1}$  with the reflected point  $p^r$ ;
      end
    else if  $f(p^r) \geq f(p^{n_v})$  then
      Compute contracted point  $p^c = p^o - \gamma(p^o - p^{n_v+1})$ ;
      if the contracted point is better than the worst point,  $f(p^c) < f(p^{n_v+1})$  then
        replace the worst point  $p^{n_v+1}$  with the contracted point  $p^c$ ;
      else
        replace all points except the best  $p^1$  with  $p^i = p^1 + \delta(p^i - p^1)$ ;
      end
    else
      replace the worst point  $p^{n_v+1}$  with the reflected point  $p^r$ ;
    end
  end
```

---

In  $n$  dimensions, the NM method maintains a set of  $n+1$  test points arranged as a simplex. It then extrapolates the behaviour of the objective function measured at each test point in order to find a new point and to replace one of the old test points with the new one, and so the technique progresses.  $\alpha$ ,  $\beta$ ,  $\gamma$  and  $\delta$  are parameters of the algorithm, and they have a significant effect on the performances of the algorithm. Standard values are  $\alpha = 1$ ,  $\beta = 2$ ,  $\gamma = \frac{1}{2}$  and  $\delta = \frac{1}{2}$ .

The Nelder-Mead method frequently gives notable improvements in the first few iterations and quickly produces satisfying results. Also, it typically requires only

one or two objective function evaluations per iteration. This feature is needed in applications where each function evaluation is time-consuming. Aside from being simple to understand and use, this is the main reason for its reputation in practice.

Contrarily, the lack of convergence theory is often considered in practice as a numerical breakdown of the algorithm, even for smooth and well-behaved functions [17]. The constraints handling is also an issue as will be discussed in the following.

## **DIRECT**

The modification of Lipschitzian optimization motivated the DIRECT optimization algorithm development [18]. It was created to solve challenging global optimization problems with bound constraints. DIRECT is a sampling algorithm; it requires no knowledge of the objective function derivatives. Alternatively, the algorithm samples points in the search domain and uses the information it has obtained to decide where to search next. The name DIRECT originates from the shortening of the expression "Diving RECTangles," which outlines the way the algorithm moves towards the optimal solution.

In the first step, DIRECT transforms the search space to be the unit hypercube. The objective function is then sampled at the center-point of this cube. The latter is afterwards divided into smaller hyper-rectangles whose center-points are also sampled. DIRECT identifies a set of potentially optimal rectangles at each iteration. All potentially optimal rectangles are further divided into smaller rectangles. The procedure described above is repeated for a predefined number of iterations. When there is no limitation for the number of iteration, DIRECT will exhaustively sample the domain, it will cluster sample points around local and global optima [19] [20].

## **Other algorithms**

There exist other well-known algorithms that exploit different ideas.

- The pattern search (PS) method samples a set of points around the current point, looking for the one where the value of the objective function is lower than the value at the current point. The set of points is called a mesh. The mesh is formed by adding the current point to a scalar multiplied by a set of vectors called a pattern. If the pattern search algorithm finds a point in the mesh that improves the objective function at the current point, the new point becomes the current point at the next iteration of the algorithm [21].
- Additionally to his significant contributions to the development of quasi-newton algorithms, Michael JD Powell developed many derivative-free algorithms from which we cite COBYLA, BOBYQA, ... [22].

## **Extension to constrained problems**

Different strategies can treat the constrained problems depending on the algo-

rithm; one of the most used is the penalty method. The penalty method replaces a constrained optimization problem by one or a sequence of unconstrained problems whose solutions ideally converge to the solution of the original constrained problem.

To recall, the constrained problem is written as

$$\begin{aligned} \min_p f(p) \\ \text{s.t. } g_i(p) \leq 0, \quad i = 1 \dots n_c \end{aligned} \quad (1.22)$$

Using the penalty method, it can be transformed

$$\min_p f(p) + P(p) \quad (1.23)$$

where  $P$  is the penalty function. In the literature, we can find different penalty methods such as the augmented Lagrangian function, barrier function, interior point, *etc.* Each of them exploits different ideas with more or less complicated implementation [23]. The exterior penalty method is commonly used because of its simple implementation. The exterior penalty function is written as

$$P(p) = M \sum_{i=1}^{n_c} \max(0, g_i(p)) \quad (1.24)$$

$M$  is the penalty parameter. The idea is to start with a relatively small value of  $M$ . Subsequently, we will solve a series of unconstrained problems with monotonically increasing values of  $M$  chosen so that the solution of each new problem is “close” to the previous one. This will prevent any notable difficulties in finding the minimum from one iteration to the next. Sometimes, a well-chosen value of  $M$  can lead to the optimum in one iteration.

### **Extension to multi-objective problems**

As for the derivative-based algorithms, heuristic-based algorithms can be extended to multi-objective problems through weighted-sum or  $\epsilon$ -constraint methods.

### **Extension to black-box problems**

Heuristic-based are well suited for black-box problems since no additional information is needed except the response of the model to a set of variables.

## **C Population-based algorithms**

The previous optimization algorithms maintain a single current solution. Population-based algorithms maintain multiple solutions. They are usually motivated by biological evolution, such as the evolution of species, behaviour of fireflies, *etc.* [24] [25].

One of the most popular is the genetic algorithm (GA), which is inspired from natural selection [26]. GA starts by creating a random initial population, then creates a sequence of new populations. At each iteration, the algorithm uses the individuals in the current generation to create the ones of the next population. For this goal, the algorithm performs different operations, *i.e.* elitism, selection, crossover and mutation. The algorithm is detailed below.

---

**Algorithm 3:** Genetic algorithm

---

```
create a random initial population ;
while Not terminated do
    evaluate the objective function at the current population ;
    rank the population based on the objective function values ;
    select the parents based on the ranking ;
    choose an elite from the current population to pass the next one ;
    produce children from the parents ;
    replace the parents with the children to form the next population;
end
```

---

**Extension to constrained problems**

As for the heuristic-based algorithms, the penalty method can be used to handle constraints in population-based algorithms. Nevertheless, some other strategies were developed, such as the one developed by K. DEB [27].

**Extension to multi-objective problems**

Population-based algorithms are well suited to multi-objective problems; they are sometimes dedicated to only multi-objective optimization since population-based algorithms dominate multi-objective optimization literature. This is explained by the fact that there are many solutions in each population. These solutions ideally converge the desired Pareto front. NSGA-II is one of the most known optimization algorithms to deal with multi-objective problems [27].

**Extension to black-box problems**

No additional modifications are needed for the population-based algorithms to handle black-box problems, except that the fact that most black-box problems are costly and special measures should be undertaken to make the optimization tractable in a limited time.

**D Exact algorithms**

This category of algorithms is less addressed in the literature because of the implementation complexity and the computational cost needed for solving specific problems. However, exact methods are the only ones that can guarantee global

optimality of the solution. This feature is highly needed but generally sacrificed for the sake of computation time by using other algorithms. Branch and bound (B&B) is the most known of exact algorithms [28]. Initially, it was used to solve integer-linear problems. The process of resolution is simple. First of all, the problem is solved as if the variables are continuous (relaxed problem); if the solution variables are integers, then it is the optimal solution. Otherwise, one variable is chosen to split the initial problem into two sub-problems, then solve the corresponding relaxed problems to compute the bounds. The process continues until a satisfactory solution is found.

An extension to optimizations of the general case (nonlinear and continuous) was introduced, it relies on the estimation of bounds using interval arithmetic [29]. The performances of interval B&B algorithms depend on the efficient estimation of the lower and upper bounds on the solutions. Thus, it can be applied to only a limited number of situations where the objective function and constraints are explicit.

---

**Algorithm 4:** Branch and bound algorithm

---

```
set the initial domain  $S$  ;
set  $\tilde{f} = +\infty$  the lower bound of the objective function  $f$ ;
set a list  $L = \{(+\infty, S)\}$  initialized by the space and the lower bound of  $f$ ;
repeat
    extract from  $L$  the domain with the smallest lower bound;
    split the element into two sub-domains  $V_1, V_2$  with respect to a
    dimension ;
    for  $i = 1$  to  $2$  do
        compute the lower and upper bounds  $\underline{f}_i, \overline{f}_i$  of  $f$  on  $V_i$ ;
        if  $\tilde{f} \geq \underline{f}_i$  then
            insert  $(\underline{f}_i, V_i)$  in  $L$ ;
            if  $\tilde{f} \geq \overline{f}_i$  then
                update  $\tilde{f} = \overline{f}_i$ ;
                eliminate from  $L$  all the domains  $(z, Z)$  for which  $z > \tilde{f}$ ;
            end
        end
    end
until  $\tilde{f} - \min_{(z,Z) \in L} z < \varepsilon$ ;
```

---

B&B is a systematic enumeration of candidate solutions by splitting the search space into smaller spaces. The algorithm explores each smaller space then estimates the upper and lower bounds on each one of them. A space is pruned if it cannot produce a better solution than the best one found so far by the algorithm.

### Extension to constrained problems

Constraints can be handled relatively easy by modifying the previous algorithm

slightly. The constraints bounds need to be calculated, then the elimination will be based on the objective function bounds and the constraints bounds. The sub-domains with at least one of the constraints that have a lower bound higher than 0 are eliminated.

### **Extension to multi-objective problems**

Multi-objective B&B is less addressed in the literature because of the double complexity, first, for the multi-objective aspect and secondly the cost of the exact search [30].

### **Extension to black-box problems**

Minimal cases are presented in the literature where exact algorithms were able to deal with black-box problems [29] [31] [32]. The estimation of the bounds of the functions is difficult or even impossible. Thus, the usage of exact algorithms is not suitable since these bounds are key-ingredient for their success.

## **E Approximation-based algorithms**

The basic idea is that if the function is noisy and likely expensive to evaluate, then that function can be sampled at some points, and a fit of it is created. Then, the optimization is not performed on the original function, but on the cheap to evaluate and smooth fit [33]. These fits are referred to by many names such as approximation models, predictors, surrogate models, meta-models, and response surfaces. These terms will be used synonymously throughout this dissertation. The goal of using a surrogate model is to provide a smooth functional relationship of satisfactory fidelity to the real function with the added advantage of quick computational speed. The approximation could be used in combination with a derivative-based algorithm or other algorithms. The details on how to build and to exploit approximations effectively are still keeping black-box optimization a thriving research area.

It is worth noting that the approximation itself is not an optimization algorithm; however, its combination with another or many optimization algorithms aforementioned is an optimization algorithm.

Many approximation methods exist, such as polynomial approximation, radial basis function, neural networks, *etc.* Nevertheless, the underlying usage procedure is almost the same with minor discrepancies, and the abstract algorithm is presented below.

Because the process of fitting the models and locating the new sample points can be viewed as optimization problems themselves, the burden associated with approximation-based algorithms can be significant. On the other hand, other methods, such as genetic algorithms or derivative-based algorithms, require minimal computational effort in determining where to evaluate the functions next. However, they

**Algorithm 5:** Approximation

---

```

create a initial sample using a design of experiment ;
while Not terminated do
    | evaluate the objective function and the constraint at the new sample ;
    | fit the surrogate model ;
    | find new sample using the surrogate model;
end

```

---

require a large number of function evaluations to converge to a good solution. The benefit of approximation algorithms is that each iteration uses as much information as possible in determining where to evaluate the functions next, enabling them to locate good solutions with fewer iterations. This makes it best-suited to situations where the functions are expensive, and the designer cannot afford to perform a large number of function evaluations.

**Response surface methodology**

The most simple surrogate model is a second-order polynomial function; it is used by the response surface methodology to have a local approximation of the real function [34]. The surrogate model is written as

$$\hat{f}(p) = a + \sum_{i=1}^{n_v} b_i p_i + \sum_{i=1}^{n_v} \sum_{j=i}^{n_v} c_{ij} p_i p_j \quad (1.25)$$

where  $\hat{f}$  is the approximation or the predictor of the real function  $f$ . The hat notation will be commonly used throughout this dissertation to denote the predictors.  $a$ ,  $b_i$  and  $c_{ij}$  are the parameters of the predictor. They are identified using a regression strategy based on  $n_{evals}$  evaluations of the real function. The regression aims to reduce the error between the real function and its predictor in the evaluated points. It can be written as an optimization problem.

$$\min_{a, b_i, c_{ij}} \sum_{k=1}^{n_{evals}} (f(p^k) - (a + \sum_{i=1}^{n_v} b_i p_i^k + \sum_{i=1}^{n_v} \sum_{j=i}^{n_v} c_{ij} p_i^k p_j^k))^2 \quad (1.26)$$

The number of evaluations  $n_{evals}$  should be at least  $\frac{(n_v+1)(n_v+2)}{2}$  to be able to capture the behaviour of the real function. These evaluations should be chosen carefully to prevent numerical problems and also to enable the predictor to represent as best as possible the real function in a local region [34].

Once the parameters are found, the predictor  $\hat{f}$  can be used to optimize the real function. Generally, the second-order polynomial cannot capture the behaviour of the real function on the whole domain; thus, multiple response surfaces can be fit in different locations to capture the maximum of information about the real function.

Other surrogate models enable to fit only one response surface on the whole domain and capture the global behaviour such as radial basis function and its generalization Kriging.

### **Kriging**

Kriging is a data interpolation scheme that has its roots in the field of geostatistics [35]. The technique was adapted for data coming from deterministic computer simulations. This form of data collection and approximation is known as Design and Analysis of Computer Experiments (DACE), Kriging is based on a regression term and a stochastic term [36]. The regression part is a polynomial function, while the stochastic term aims to compensate for the error due to regression and is constructed based on the location of the sampled points. While the idea behind Kriging is simply put here, the details are left for the next chapter.

### **Extensions**

In recent years, numerous advances have been made in approximation-based algorithms. Constraint handling strategies and extension to multi-objective optimization have been discussed [37][38][39][40][36].

A significant area where little research has been done to date is parallelization. El Bechari *et al.* discussed the limitations of some of the strategies [41]. The parallelization strategies represent one of the main contributions of this dissertation and are discussed in the next chapter.

### **1.1.3 Discussion**

The models of electrical machines are usually considered as computationally expensive black-box; the time needed for one simulation can vary from minutes to hours or even days for more complex phenomena. Thus, not all the algorithms are suited for this kind of model. The choice of the algorithm is critical for the success of optimization, mainly in terms of the quality of the solution and the computational effort needed for obtaining a reliable solution. Additionally, these models can be noisy due to the discrete aspect which furthermore limits the choice of the algorithms.

The ideal algorithm is the one that

- has the least cost
- leads to the global solution
- handles the constraints efficiently
- is capable of dealing with numerical noise



Nevertheless, such an algorithm does not exist. The **No free lunch theorem** demonstrates that if an algorithm performs well on a particular class of problems [42], then it necessarily pays for that with degraded performance on the set of all remaining problems. Thus an inevitable compromise should be found. We consider that the cost is a higher priority criterion than the other criteria and to choose the algorithms that offer a margin of improvement to the studied problem.

The derivative-based algorithms offer the best performances in term of cost and constraints handling, thanks to the rigorous foundations of the algorithms nevertheless they perform poorly for problems with many local minima due to their local properties, and also they perform inadequately in terms the noise handling since the derivatives are approximated by finite difference. At the same time, this approximation is highly sensitive to numerical noise.

Exact and population-based algorithms are not adapted to our problem because of the excessive computational cost needed for performing an optimization.

Heuristic-based algorithms are good at handling the numerical noise since no derivatives are needed. However, they perform very bad at constraints handling because the choice of the penalty function highly conditions the results. This is caused when the constraints do not have the same magnitude. If some constraints are dominant, the algorithm will steer towards a solution that satisfies those constraints at the expense of searching for the minimum. Similarly, the value of the penalty parameter should be fixed so that the penalty term's magnitude is not much smaller than the magnitude of the objective function. If an imbalance exists, the objective function's influence could direct the algorithm to head towards an unbounded minimum, even in the presence of unsatisfied constraints. In either case, convergence may be exceedingly slow.

As noted before, approximation-based algorithms are very flexible since the surrogate model can be combined with one or many of the algorithms mentioned above. Their development aims to reduce the computational cost needed for optimization. Different ideas were studied in the literature. One particular optimization algorithm is known as Efficient Global Optimization (EGO), developed by Jones et al. [43]. It employs Kriging surrogate models as the approximation method. Other ideas, such as genetic algorithm assisted by surrogate models, are also of interest.

## 1.2 Uncertainty

The optimization problem types described in the previous section implicitly assume that the variables for the given problem are known precisely. For many actual problems, however, the problem variables cannot be known deterministically for various reasons such as measurement error or tolerances on the manufacturing process. In most cases, deterministic optimization without considering the uncertainties of vari-

ables will find an optimum that lies on one or several constraint boundaries. With a small deviation of the solution, this one could easily violate one or more constraints and fall into the failure domain. Moreover, if the optimum lies on a very narrow valley of the objective function, even a small variation in the variables could result in a significant change for the performance.

The uncertainties can be reduced but never eradicated completely. For that reason, ways to decrease the influence of uncertainties with existing techniques are entirely worth studying. Especially in engineering applications, less influence leads to less cost or more stable performance, which are quite valuable in the real world.

The variability due to uncertainty is inherent in all systems, and have different natures that can be categorized as follows:

- **Model uncertainty** Also known as model inadequacy, or model discrepancy, this arises from a lack of knowledge on the underlying physics in the problem. It depends on how accurately a mathematical model represents the real-life situation because models are always approximations of the reality.
- **Numerical uncertainty** This type originates from numerical errors and numerical approximations in the implementation of the computer model. Most models are complex to solve accurately. For example, the finite element method is used to approximate the solution of a partial differential equation (which introduces numerical errors).
- **Experimental uncertainty** This comes from the variability of experimental measures. The experimental uncertainty is unavoidable and can be noticed by repeating a measurement many times using the same settings for all inputs/variables.
- **Variable uncertainty** This arises from the variability of the input variables of the model. For example, the dimensions of a workpiece in the manufacturing process may not be exactly as designed, which would induce variability in its performance.

This dissertation deals with the last category of uncertainties that enables one to get relationships between the model input variables and their impact on its responses. Thus, the uncertainty on the variables induces uncertainty on the response of the model. This means that the uncertainty propagates through the model to generate uncertainty on the responses. This process is called uncertainty propagation, and it is performed using simulation and numerical tools.

The models are generally deterministic in the sense that model inputs, *i.e.* variables, are assumed to be known. However, we have seen that this is not always true. Deterministic models are then no longer directly adapted to handle this type

of input data. It is necessary to use efficient methods to propagate the uncertainties on the input data to the model's output quantities.

One of the challenges to using models is to capture the effect of uncertainty on the model output in an efficient manner. The problem can be stated as: given the probability density function (PDF) of the random variables (e.g. those of the design variables and parameters), what are the statistics of a system output? The issue is how to propagate the effect of uncertainty through the model. This process is often referred to as uncertainty quantification. In this section, we will start by detailing the uncertainty propagation methods and, afterwards, the integration of these methods in an optimization procedure is presented.

### 1.2.1 Uncertainty propagation

As stated before, uncertainty on the variables leads to uncertainty on performances. Thus, the first stage of any uncertainty propagation procedure is to characterize the input of the model. The input variables are considered as random variables characterized by their PDF. Other statistics such as mean, variance, range, skewness and covariance can also be characterized. The pdf of random variables are not always needed but generally turn out to be useful for the variables characterization.

Once the random input variables are characterized, numerical tools are needed for the propagation of uncertainty, that is to say, compute statistics on the indicators of performance. In the literature, many algorithms enable to propagate uncertainty. We cite five groups.

- Monte-Carlo simulation and variants [44].
- Approximations methods [45][46][47].
- Perturbation methods (first and second order) [48][49].
- Most probable point (MPP) [50].
- Worst case method [51].

It is worth noting that each group is adapted to the performance metric needed. Many performance metrics describing the behaviour of output under uncertainty have been suggested in engineering design. Perhaps the most common performance metric is the expected value of output under probabilistic uncertainties, also known as the mean. Another property of interest is a measure of the spread of the outputs, since if a design has a good expected value of performance but a high spread such that it can also give poor performance it could be undesirable. The standard measure of the spread is the variance. If a system is being designed for reliability, or the output is to be constrained, then the probability of failure will be of interest. A common

measure of this sort is that the probability of failure is higher than some threshold value.

### A Monte-Carlo simulation

One of the most widely used techniques for uncertainty propagation is the Monte Carlo (MC) sampling technique, which is based on a pseudorandom generator used to approximate the desired distribution. MC evaluates the deterministic model at the distribution realizations and, lastly, aggregates the model's outputs to get insights on the uncertainty of the performance. One could compute the mean, the variance and probability of failure with ease [44]. MC sampling is a useful propagation technique because its convergence is independent of the dimensionality of the problem.

While this approach is attractive due to its simplicity, it has numerous deadfalls. For example, a large number of samples must be taken in order to determine low probability events, *e.g.* thousands or millions of trial design variable vectors. The computational expense of this approach can quickly become intractable.

Other computational cost reduction techniques exist, such as Importance sampling, Latin Hypercube Sampling (LHS), and Hammersley Sequence Sampling. They enable one to sample the space in a smart way to reduce the number of sampled points. Nevertheless, sampling methods remain costly when considering computationally expensive models, such as a finite element analysis.

### B Approximation methods

When the number of variables is low, more efficient methods are available than MC sampling. They primarily involve building a surrogate model as an approximation to the actual output, as presented for deterministic optimization, using relatively few model evaluations. Such a surrogate model can subsequently be sampled and analyzed at a negligible computational cost compared to evaluating the actual output or sometimes computing the performance metrics analytically. Polynomial Chaos expansion is widely used in engineering design [52] [53] [47], among other surrogate models.

### C Perturbation methods

These methods have advantages when dealing with relatively small input variability and outputs that do not express high nonlinearity [54][48]. The model must usually be linearized by approximation to a first-order or second-order Taylor series expansion.

The first-order expansion of a function  $f$  with respect to a random variable  $X$  of mean  $\mu_X$  and variance  $\sigma_X^2$  is written.

$$f(X) \approx f(\mu_X) + (X - \mu_X)^T \nabla f(\mu_x) \quad (1.27)$$

The mean value  $\mu_f$  and the variance  $\sigma_f^2$  of the function  $f$  are computed as follows.

$$\begin{aligned} \mu_f &= \mathbf{E}(f(X)) \\ &\approx \mathbf{E}(f(\mu_X) + (X - \mu_X)^T \nabla f(\mu_x)) \\ &= f(\mu_x) + \mathbf{E}(X - \mu_x)^T \nabla f(\mu_x) \\ \mu_f &= f(\mu_x) \end{aligned} \quad (1.28)$$

$$\begin{aligned} \sigma_f^2 &= \mathbf{V}(f(X)) \\ &\approx \mathbf{V}(f(\mu_x) + (X - \mu_X)^T \nabla f(\mu_x)) \\ &= \mathbf{V}(X - \mu_x)^T (\nabla f(\mu_x)) \\ &= \mathbf{V}(X)^T (\nabla f(\mu_x)) \\ \sigma_f^2 &= \sigma_X^T (\nabla f(\mu_x))^2 \end{aligned} \quad (1.29)$$

The uncertainty propagation is done easily. The mean value of the function is obtained by evaluating the deterministic function at the variable mean value. Similarly, the variance is the product of the gradient squared of the function and the variance of the variables.

As noted before, the gradient is not always a piece of accessible information; thus, this method can be applied only if a gradient of good quality is available. It is also important to note that this formula is based on the linear characteristics of the gradient of  $f$ . Therefore, it is a reasonable estimation for the standard deviation of  $f$  as long as  $\sigma_X$  are small enough.

## D Most probable point MPP

The MPP concept is utilized to compute the whole cumulative distribution function (CDF) of model output by evaluating the probability estimates at a serial of limit states [50]. The MPP method was initially developed in the field of reliability analysis. It requires that limit-state (constraints) functions be defined, and the probability of the limit-state functions less than zero can be evaluated approximately. Generally, the MPP method is used to estimate the probability of an inequality function's failure and not the whole CDF. If a function of a random variable is needed to be less than zero, then the probability of failure is written as

$$P_f = 1 - P(g(X) \leq 0) = P(g(X) > 0) \quad (1.30)$$

The first-order reliability method (FORM) has been widely used in the probability of failure estimation. The method involves Taylor's expansion of the inequality function, *i.e.* the linearization of the limit-state function, not performed around the mean value of the function, but at the MPP. The selection of an appropriate linearization point is a crucial consideration and leads to an iterative solving procedure, *i.e.* an optimization algorithm.

## E Worst-case method

With the worst-case analysis, all the uncertainties are assumed to lead to the worst possible combinations [51]. Based on this assumption, the worst-case considers the most severe possible outcome that can occur for the uncertainty of the given variable. It is a non-probabilistic approach that is based on the estimation of the worst possible outcome. There is no need to determine the distributions of functions as opposed to the MPP method that identifies a probability of failure. In the worst case, the probability of failure is zero.

### 1.2.2 Optimization under uncertainty

The formulation of an Optimization under Uncertainty (OuU) primarily concerns how to compare the model's behaviour under uncertainty between designs. The formulation defines the underlying landscape that an optimizer attempts to navigate and therefore influences the design obtained from performing an optimization. Therefore, the performances are related to two types of variables: first, the optimization  $p$ , which are deterministic and secondly, the random uncertain variables  $X$ . Often, the deterministic variables are nothing but the mean value of the random variables. Thus for the sake of presentation simplicity, this assumption, *i.e.*  $p = \mu_X$ , is holding throughout this dissertation unless stated otherwise.

The original development of methods for taking account of uncertainty in the design process of engineering systems is often credited to Taguchi and his "robust design" approach. This framework initiated an interest in controlling robustness through the design of the system instead of trying to reduce the uncertainties on the inputs directly. Taguchi's method is not necessarily an optimization-based approach; instead, it makes use of the design of experiments. The design space is sampled, and the best design from these samples is selected. Since this initial progress, a mixture of methods for design under uncertainty has been proposed for which robustness is the priority.

The recent growth of interest in uncertainty quantification can perhaps be attached to the importance of rigorously quantifying the influence of uncertainties for engineering. The field of OuU grew from the desire to use this rigorous uncertainty quantification within design optimization. In the past few decades, OuU techniques have been applied to various engineering applications, and therefore a variety of

approaches for formulating an OuU problem have been proposed.

### A Robustness, reliability or worst-case

The appropriate formulation depends on the application. In engineering applications, the constrained quantities can represent an extreme failure of the system, and so the likelihood of constraint satisfaction is a key design driver; this is often known as "reliability-based design optimization" (RBDO). RBDO formulations often make use of the MPP method for uncertainty propagation [55][56].

$$\begin{aligned} \min_p \quad & \mu_f(p) \\ \text{s.t.} \quad & P(g_i(p, X) > 0) \leq P_f, \quad i = 1 \dots n_c \end{aligned} \quad (1.31)$$

In contrast, "robust design optimization" (RDO) formulations focus on the likelihood of obtaining excellent performance [57]. Therefore RDO mainly deals with the influence of uncertainties on the performances to be optimized instead of the constrained quantities. Often, "robust design" refers explicitly to simultaneously improving the expected performance represented by  $\mu_f(p)$  and reducing the variability represented by  $\sigma_f(p)$  and  $\sigma_{g_i}(p)$ .

$$\begin{aligned} \min_p \quad & [\mu_f(p), \sigma_f(p), \sigma_{g_i}(p)] \\ \text{s.t.} \quad & \mu_{g_i}(p) \leq 0, \quad i = 1 \dots n_c \end{aligned} \quad (1.32)$$

These "robust" and "reliability" paradigms are not mutually exclusive and can be combined.

On the other hand, worst-case optimization may be expressed as a minimax problem. This means the original values of the objective function and the constraint functions are substituted by their worst values, which is the maximum over the uncertainty set.

$$\begin{aligned} \min_p \quad & \max_{x \in \Omega(X)} f(p, x) \\ \text{s.t.} \quad & \max_{x \in \Omega(X)} g_i(p, x) \leq 0, \quad i = 1 \dots n_c \end{aligned} \quad (1.33)$$

### B Nested-loops to sequential

The general formulation of OuU is mathematically a nested two-level structure, consisting of a design optimization loop that repeatedly calls uncertainty quantification analyses in a series of inner loops. This approach is computationally costing and

therefore has limited applicability for real engineering problems. In order to surpass the computational difficulties, many formulations have been proposed in the literature [49].

The "sequential optimization and reliability assessment" (SORA) method decouples the RBDO process into a sequence of deterministic design optimization followed by a set of reliability evaluation loops [58]. It uses the reliability information from the previous iteration to shift the deterministic constraints. The process continues until convergence of both sub-problems is reached. SORA has a structure of two iterative loops. However, the two loops are decoupled; hence it is significantly more cost-effective than the nested two-level process.

In the same essence, a worst-vertex-based WCO was proposed. It aims to observe the values of the bounds of  $\Omega(X)$  and predicts the worst vertex value [59]. The objective is to determine the directions of ascent in which the values of objective function and constraints increase and then evaluate at the bounds of  $\Omega(X)$ . Consequently, the inner maximization problems are replaced by a few evaluations on the uncertainty space's bounds.

### 1.2.3 Discussions

Well-known methods for uncertainty propagation have been reviewed in this section. Methods based on the derivatives remain the best compromise between accuracy and efficiency, especially when the coefficients of variation of the random input variables are not too large, and the model is almost locally linear.

Derivatives can be used for different performance metrics, such as the perturbation method, MPP and, in some cases, the worst-case method. Nevertheless, they require models that implement the derivatives of the response quantities.

Approximation methods are preferred if the derivatives are not available, and the model is considered a black-box, as it is often the case of commercial electromagnetic software.

## 1.3 Electrical machines

An electrical machine is a broad term for machines using electromagnetic fields, such as motors, generators, and transformers. Some are electromechanical energy converters: an electric motor transforms electricity into mechanical power while a generator converts mechanical power into electricity. The moving components in a machine can be rotating (rotating machines) or linear (linear machines). The third category of electrical machines is transformers, which, although they do not have any moving components are also energy converters, changing the voltage level of an alternating current.



Electrical machines are based on the electromagnetic principles that generate force or induce voltages and currents. Five principles allow us to understand this link.

1. Current flowing in a conductor will produce a magnetic field around the conductor.
2. Magnetic materials, with high relative permeability, provide a mean to direct and focus magnetic flux.
3. Motor Action: if current flows in a conductor subject to an external magnetic field, a force will be exerted on the conductor.
4. Generator Action: if a conductor is moved through a magnetic field, a voltage will be induced in the conductor.
5. Transformer Action: if the magnetic flux passing through a coil changes, there will be a voltage induced in the coil.

The ability to produce a magnetic field using coils and electric current can be coupled with ferromagnetic materials to direct and concentrate magnetic flux. Combined with transformer, generator and motor actions, these concepts form the foundation of electrical machine design and operation.

A rigorous physical model was given to model the electromagnetic phenomena, *i.e.* Maxwell's equations. This model is expressed as partial differential equations (PDE). These PDE are generally difficult to solve analytically due to various reasons such as a complex studied domain, nonlinear material properties and dynamic time behaviour. Thus, numerical methods that discretize the space and time domain are used. The most used one is the finite element method. Such methods are costly in computational time. In this section, we will present the physical model, its discretization and the resolution using the finite element method.

### 1.3.1 Physical model

Maxwell's equations can describe electromagnetic phenomena. They are separated into two groups.

Firstly, fundamental equations contain only fields.

$$\nabla \times \mathbf{E} = -\frac{\partial \mathbf{B}}{\partial t} \quad (1.34)$$

$$\nabla \cdot \mathbf{B} = 0 \quad (1.35)$$

where  $\mathbf{E}$  is the electric field vector ( $V/m$ ) and  $\mathbf{B}$  is the magnetic flux density vector ( $T$ ).

The second group links the fields to the sources.

$$\nabla \times \mathbf{H} = \frac{\partial \mathbf{D}}{\partial t} + \mathbf{J} + \mathbf{J}_s \quad (1.36)$$

$$\nabla \cdot \mathbf{D} = \rho \quad (1.37)$$

where  $\mathbf{H}$  is the magnetic field vector ( $A/m$ ),  $\mathbf{D}$  is the electric flux density vector ( $C/m^2$ ),  $\mathbf{J}$  and  $\mathbf{J}_s$  are the eddy and imposed current densities respectively ( $A/m^2$ ) and  $\rho$  is the electric charge density ( $C/m^3$ ).

For electrical machines, the quasi-static approximation is widely used. It states that displacement currents are negligible in comparison with the imposed currents. Moreover, electric charges are also negligible.

$$\left| \frac{\partial \mathbf{D}}{\partial t} \right| \ll |\mathbf{J}_s|, \quad |\rho| \ll 1$$

Maxwell's equation in the quasi-static assumption are summarized as follows.

$$\nabla \times \mathbf{H} = \mathbf{J} + \mathbf{J}_s \quad (1.38)$$

$$\nabla \times \mathbf{E} = -\frac{\partial \mathbf{B}}{\partial t} \quad (1.39)$$

The system of equations (1.38 - 1.39) is under-determined. To solve such a system, more equations should be given. Constitutive Laws link these quantities throughout electromagnetic material behaviour.

We particularly define two equations

$$\mathbf{B} = \mu(\mathbf{H}, t, T, \varepsilon, \dots)(\mathbf{H} + \mathbf{H}_c) \quad (1.40)$$

$$\mathbf{J} = \sigma(t, T, \dots)\mathbf{E} \quad (1.41)$$

where  $\mathbf{H}_c$  is the coercive field of magnets,  $\mu$  is the magnetic permeability and  $\sigma$  the electric conductivity of the material.  $\mu$  and  $\sigma$  depend on electromagnetic fields and other parameters like time  $t$ , temperature  $T$ , and mechanical strain  $\varepsilon$ .

In this work, all the parameters other than electromagnetic fields are considered constant. Thus, these laws are written as

$$\mathbf{B} = \mu(\mathbf{H})(\mathbf{H} + \mathbf{H}_c) \quad (1.42)$$

$$\mathbf{J} = \sigma\mathbf{E} \quad (1.43)$$

Note that equation 1.43 corresponds to the local form of Ohm law. Equation 1.42 transcribe the magnetic behaviour of materials.

The physical model of the electromagnetic phenomenon is a combination of Maxwell's equation(1.38 - 1.39) and constitutive laws (1.42 - 1.43)

$$\nabla \times \mathbf{H} = \mathbf{J} + \mathbf{J}_s \quad (1.44)$$

$$\nabla \times \mathbf{E} = -\frac{\partial \mathbf{B}}{\partial t} \quad (1.45)$$

$$\mathbf{B} = \mu(\mathbf{H})(\mathbf{H} + \mathbf{H}_c) \quad (1.46)$$

$$\mathbf{J} = \sigma\mathbf{E} \quad (1.47)$$

### 1.3.2 Mathematical model

For the sake of presentation simplicity, we will only detail the two-dimensional magnetostatic case. The electromagnetic fields do not change in time and are invariant in the dimension perpendicular to the two-dimensional domain. For more information about the general formulation, we refer the reader to [60] [61]. The resolution of electromagnetic PDE was and still extensively studied.

Maxwell equation in the magnetostatic case decouple the magnetic and the electric fields and are reduced to

$$\nabla \times \mathbf{H} = \mathbf{J}_s \quad (1.48)$$

$$\mathbf{B} = \mu(\mathbf{H})(\mathbf{H} + \mathbf{H}_c) \quad (1.49)$$

$$(1.50)$$

The quantities are then reduced to  $\mathbf{B} = [B_1, B_2, 0]^\top$ ,  $\mathbf{H} = [H_1, H_2, 0]^\top$ ,  $\mathbf{H}_c = [H_{c1}, H_{c2}, 0]^\top$  and  $\mathbf{J} = [0, 0, J_{s3}]^\top$ .

We denote the studied domain as  $\mathcal{D} \subset \mathbf{R}^2$ . Thus the quantities depend on the two coordinates of the two-dimensional domain. On the other hand, boundary conditions should be applied to solve Maxwell equations on the domain  $\mathcal{D}$ . Furthermore, additional quantities are introduced to tackle issues. These quantities are called the potentials; they enable to reduce the number of unknowns of the original Maxwell equation and to have functionals that are sufficiently smooth since the fields (magnetic and electric) in the Maxwell equation present discontinuities on the material interfaces.

#### A Potential formulation

Since  $\mathbf{B}$  is divergence-free *i.e.*  $\nabla \cdot \mathbf{B} = 0$ , we can find a magnetic vector potential  $\mathbf{A}$  such that

$$\mathbf{B} = \nabla \times \mathbf{A} \quad (1.51)$$

Thus, equations (1.48 - 1.49) can be combined in one equation

$$\nabla \times \left( \frac{1}{\mu} \nabla \times \mathbf{A} \right) = \mathbf{J}_s + \nabla \times \mathbf{H}_c \quad (1.52)$$

The solution  $\mathbf{A}$  of this equation is not unique. Any quantity  $\mathbf{A}'$  is a solution for a scalar function  $u$

$$\mathbf{A}' = \mathbf{A} + \nabla u \quad (1.53)$$

On the other hand, we have

$$\mathbf{B} = \begin{pmatrix} \frac{\partial A_3}{\partial x_2} - \frac{\partial A_2}{\partial x_3} \\ \frac{\partial A_1}{\partial x_3} - \frac{\partial A_3}{\partial x_1} \\ \frac{\partial A_2}{\partial x_1} - \frac{\partial A_1}{\partial x_2} \end{pmatrix} \quad (1.54)$$

$\mathbf{A}$  is invariant in third dimension, then  $\frac{\partial A_2}{\partial x_3} = 0$  and  $\frac{\partial A_1}{\partial x_3} = 0$ .

Furthermore, by comparison with  $\mathbf{B} = [B_1, B_2, 0]^\top$ , it follows that

$$\frac{\partial A_2}{\partial x_1} - \frac{\partial A_1}{\partial x_2} = 0 \quad (1.55)$$

This equation has an infinite number of solutions, the choice of  $A_1$  and  $A_2$  to be equal to zeros will enable reducing the number of unknowns to be solved.

The invariance in the third dimension and the last choice enable to ensure the uniqueness of the solutions. And, the quantities are written as

$$\mathbf{A} = \begin{pmatrix} 0 \\ 0 \\ A_3(x_1, x_2) \end{pmatrix} \quad (1.56)$$

$$\mathbf{B} = \nabla \times \mathbf{A} = \begin{pmatrix} \frac{\partial A_3}{\partial x_2} \\ -\frac{\partial A_3}{\partial x_1} \\ 0 \end{pmatrix} \quad (1.57)$$

Therefore, only the third component of  $\mathbf{A}$  ( $A_3$ ) is considered as an unknown. For the sake of presentation homogeneity, we keep the notation  $\mathbf{A}$  to denote the unknown function.

## B Boundary and interface conditions

Let  $\Gamma$  be the boundary of  $\mathcal{D}$ . Boundary condition model the behaviour of the system at the limit of the studied domains.

Boundary conditions on magnetic quantities  $\Gamma_B$  and  $\Gamma_H$  are complimentary on  $\Gamma$  ( $\Gamma_B \cup \Gamma_H = \Gamma$  and  $\Gamma_B \cap \Gamma_H = \emptyset$ )

$$\mathbf{n} \times \mathbf{H}|_{\Gamma_H} = \mathbf{0} \quad (1.58)$$

$$\mathbf{n} \cdot \mathbf{B}|_{\Gamma_B} = 0 \quad (1.59)$$

$\mathbf{n}$  is the outer normal unit normal vector to  $\Gamma$ .

When passing from a domain to another one of different material properties, the electromagnetic fields present discontinuities. Thus, interface conditions ensure to have a certain level of continuity through material interfaces [62].

Between two domains  $\mathcal{D}_i$  and  $\mathcal{D}_j$ , an interface denoted  $\Gamma_{i,j}$  is defined and  $\mathbf{n}$  an outer normal vector on  $\Gamma_{i,j}$ . Interface conditions are stated as follows.

$$\mathbf{n} \times (\mathbf{H}_i - \mathbf{H}_j)|_{\Gamma_{i,j}} = \mathbf{0} \quad (1.60)$$

$$\mathbf{n} \cdot (\mathbf{B}_i - \mathbf{B}_j)|_{\Gamma_{i,j}} = 0 \quad (1.61)$$

With these conditions, the normal component of  $\mathbf{B}$  and the tangential component of  $\mathbf{H}$  are continuous.

### C Finite element method

As mentioned before, the equation (1.52) is a partial differential equation (PDE) (Strong formulation). This equation can be solved analytically for simple systems. For complex ones, weak formulations are used instead [63], *i.e.* Finite Element Method.

The magnetostatic problem is reduced to the following equations.

$$\nabla \times \left( \frac{1}{\mu} \nabla \times \mathbf{A} \right) = \mathbf{J}_s + \nabla \times \mathbf{H}_c \quad \text{in } \mathcal{D} \quad (1.62)$$

$$\mathbf{n} \times \mathbf{A} = \mathbf{0} \quad \text{on } \Gamma_B \quad (1.63)$$

$$\left( \frac{1}{\mu} \nabla \times \mathbf{A} \right) \times \mathbf{n} = \mathbf{0} \quad \text{on } \Gamma_H \quad (1.64)$$

These equations describe the electromagnetic fields and the corresponding boundary conditions.

The problem in 2D is further reduced since

$$\nabla \times \left( \frac{1}{\mu} \nabla \times \mathbf{A} \right) = \begin{pmatrix} 0 \\ 0 \\ -\nabla \cdot \left[ \frac{1}{\mu} \nabla A_3 \right] \end{pmatrix}$$

and  $\mathbf{n} \times \mathbf{A} = \mathbf{0}$  is verified if  $A_3 = 0$  and lastly  $\left( \frac{1}{\mu} \nabla \times \mathbf{A} \right) \times \mathbf{n} = \frac{1}{\mu} \nabla A_3 \cdot \mathbf{n}$ .

Finally, the problem is wholly reduced to 2D, and by summarizing, we have

$$-\nabla \cdot \left[ \frac{1}{\mu} \nabla A_3 \right] = J_{s3} + \nabla \cdot \mathbf{M} \quad \text{in } \mathcal{D} \quad (1.65)$$

$$A_3 = 0 \quad \text{on } \Gamma_B \quad (1.66)$$

$$\frac{1}{\mu} \nabla A_3 \cdot \mathbf{n} = 0 \quad \text{on } \Gamma_H \quad (1.67)$$

with  $\mathbf{M} = \begin{pmatrix} -H_{c2} \\ H_{c1} \end{pmatrix}$ .

The interface conditions are implicitly verified in the two-dimensional magnetostatic problems.

### Variational formulations

Multiplying the PDE from 1.65 by an arbitrary test function  $v$  such that  $v|_{\Gamma_B} = 0$  and integrating over the domain leads to

$$-\int_{\mathcal{D}} \left[ \nabla \cdot \left( \frac{1}{\mu} \nabla A_3 \right) \right] v = \int_{\mathcal{D}} J_{s3} v + \int_{\mathcal{D}} \nabla \cdot \mathbf{M} v \quad (1.68)$$

Applying Green formulas on the l.h.s [64].

$$\begin{aligned} -\int_{\mathcal{D}} \left[ \nabla \cdot \left( \frac{1}{\mu} \nabla A_3 \right) \right] v &= \int_{\mathcal{D}} \left( \frac{1}{\mu} \nabla A_3 \right) \cdot \nabla v \\ &\quad - \int_{\Gamma} \left[ \left( \frac{1}{\mu} \nabla A_3 \right) \cdot \mathbf{n} \right] v \end{aligned} \quad (1.69)$$

since  $v$  vanishes on  $\Gamma_B$  (Dirichlet boundary) and  $\left( \frac{1}{\mu} \nabla A_3 \right) \cdot \mathbf{n}$  vanishes on  $\Gamma_H$  (Neumann boundary), the boundary integral vanishes too.

In the same manner, Green formula is applied to the term with  $\mathbf{M}$ .

$$\int_{\mathcal{D}} \nabla \cdot \mathbf{M} v = \int_{\mathcal{D}} \mathbf{M} \cdot \nabla v \quad (1.70)$$

This leads to the following variational formulation.

$$a(A, v) = \langle \mathbf{F}, v \rangle \quad (1.71)$$

with

$$a(A, v) = \int_{\mathcal{D}} \frac{1}{\mu} \nabla A \cdot \nabla v \quad (1.72)$$

$$\langle \mathbf{F}, v \rangle = \int_{\mathcal{D}} J_{s3} v + \int_{\mathcal{D}} \mathbf{M} \cdot \nabla v \quad (1.73)$$

Using the Lax-Milgram theorem, we can guarantee the existence and the uniqueness of the solution of equation 1.71.

In the next section, we will show how we solve this equation using the finite element method.

### 1.3.3 Numerical model

Thanks to the Galerkin method, we assume having a finite-dimensional space  $V_h$ , on which a discrete problem is associated.

$$\text{Find } A_h \in V_h \text{ such that } a(A_h, v_h) = \langle \mathbf{F}, v_h \rangle \quad \forall v_h \in V_h \quad (1.74)$$

Since the space  $V_h$  is of finite dimension denoted  $N_h$ , we let  $\alpha_i$  denote the basis functions, i.e.  $V_h = \text{span}\{\alpha_i / i = 1, \dots, N_h\}$ . Consequently, the solution  $A_h$  is of the form

$$A_h(\mathbf{x}) = \sum_{i=1}^{N_h} u_i \alpha_i(\mathbf{x}), \quad \mathbf{x} \in \mathcal{D} \quad (1.75)$$

Replacing  $A_h$  by 1.75 and  $v$  by the basis functions  $\alpha_i$ , the problem becomes

$$\text{Find } u = (u_i)_{i=1}^{N_h} \text{ such that } \sum_{i=1}^{N_h} a(\alpha_i, \alpha_j) u_i = \langle \mathbf{F}, \alpha_j \rangle \quad j = 1, \dots, N_h \quad (1.76)$$

We can write

$$\text{Find } u \text{ such that } Ku = b \quad (1.77)$$

with

$$K_{ij} = a(\alpha_i, \alpha_j) = \int_{\mathcal{D}} \left( \frac{1}{\mu} \nabla \alpha_i \right) \cdot \nabla \alpha_j \quad (1.78)$$

$$b_j = \langle \mathbf{F}, \alpha_j \rangle = \int_{\mathcal{D}} J_{s3} \cdot \alpha_j + \mathbf{M} \cdot \nabla \alpha_j \quad (1.79)$$

Now we face the problem of how to construct the space  $V_h$  and corresponding basis functions. In the following paragraph, we are discussing this matter.

### A Special choice of $V_h$

The FEM is a particular case of the Galerkin method in the construction of  $V_h$  in three main aspects:

1. A discretization  $\mathcal{T}_h$  of the domain  $\mathcal{D}$  has to be established.
2.  $V_h$  is constructed such that for each element  $T \in \mathcal{T}_h$ ,  $\alpha_i \in V_h$  is a polynomial.
3. There is a canonical basis of  $V_h$  can be easily described and have small support.

The discretization  $\mathcal{T}_h$  of the domain  $\mathcal{D}$  is called a triangulation or mesh.  $\mathcal{T}_h$  must satisfy the following properties

- Each  $T \in \mathcal{T}_h$  is a closed set with a nonempty, connected interior  $\overset{\circ}{T}$ .
- Each  $T \in \mathcal{T}_h$ , its boundary  $\partial T$  is Lipschitz continuous.
- $\mathcal{D} = \cup_{T \in \mathcal{T}_h} T$ .
- For each distinct  $T_1, T_2 \in \mathcal{T}_h$ , the intersection is of measure zero.

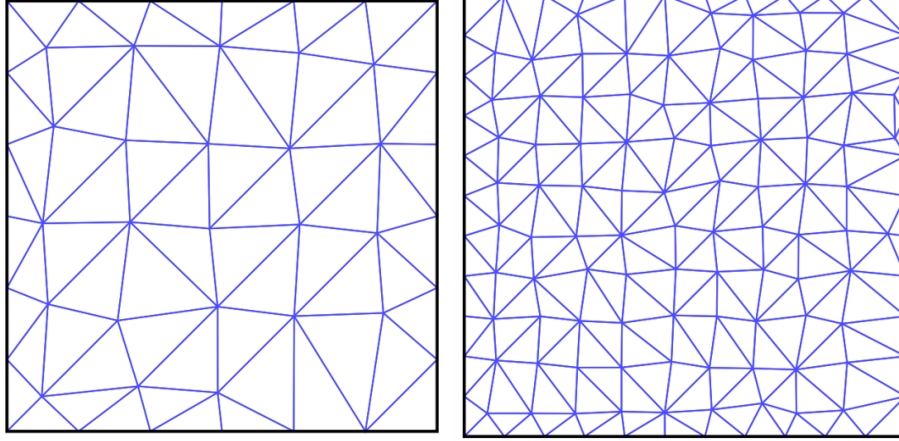


Figure 1.2: Example of 2D square triangulation, (left) coarse triangulation, (right) fine triangulation.

Different kinds of triangulation exist. Figure 1.2 shows possible triangulation of a squared domain using small triangles, rectangles could also be used.

The most simple but widely used elements are the linear and quadratic elements. The domain is meshed, *e.g.* triangles for 2D domains, on which the functions  $\alpha_i$  should be linear or quadratic. The construction is as follows.

- Let the vertices of the element (called nodes) be enumerated by an index set  $Id_n$  and let  $x_i$ ,  $i \in Id_n$  denote the corresponding nodes.
- We set

$$\alpha_i(x_j) = \delta_{ij}, \quad \forall i, j \in Id_n,$$

and with the restriction

$$\alpha_i|_T \text{ first or second order polynomial, } \forall T \in \mathcal{T}_h$$

Such a basis is called the nodal basis since the unknowns are the function values on the nodes. Likewise, we can define edge basis and facet basis in 2D and 3D.

Constructing  $V_h$  in this manner enables one to make the matrix  $K$  sparse. Consequently, relatively easy to solve the system 1.77.

The stiffness matrix  $K$  and the load vector  $b_i$  are assembled by summing over the elements of the mesh.

$$K_{ij} = \int_{\mathcal{D}} \frac{1}{\mu} \nabla \alpha_i \cdot \nabla \alpha_j = \sum_{T \in \mathcal{T}_h} \int_T \frac{1}{\mu} \nabla \alpha_i \cdot \nabla \alpha_j \quad (1.80)$$

$$b_j = \int_{\mathcal{D}} J_{s3} \alpha_j + \mathbf{M} \cdot \nabla \alpha_j = \sum_{T \in \mathcal{T}_h} \int_T J_{s3} \alpha_j + \mathbf{M} \cdot \nabla \alpha_j \quad (1.81)$$

The appearing integrals over the elements  $T$  can be evaluated efficiently using the concept of the reference element.



## B Reference element

In the 2D case, an element is a triangle. Figure 1.3 shows a reference triangle  $T_r$  and an arbitrary triangle  $T$ . Every arbitrary element can be mapped by an affine transformation to the reference element as follows

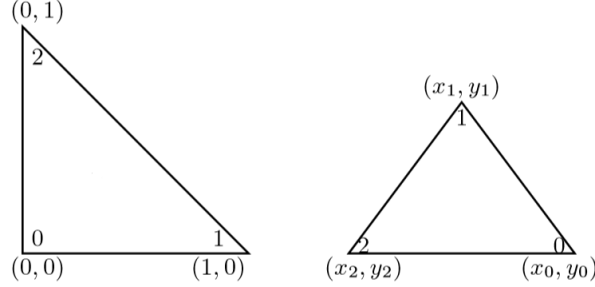


Figure 1.3: The reference element (left), an arbitrary element (right).

$$(x, y) = (x_0, y_0) + (x_r, y_r) \begin{pmatrix} x_1 - x_0 & y_1 - y_0 \\ x_2 - x_0 & y_2 - y_0 \end{pmatrix} \quad (1.82)$$

Consequently, the integrals can be easily transformed into the reference element using the mapping. For efficiently evaluating these integrals, numerical integration is used (Gauss quadrature).

Let us suppose that  $(x_g, y_g)$  and  $w_g$  are the points and weights of a Gauss quadrature of the reference element, then, the basis function and their derivatives are evaluated and written as

$$\alpha_i^g = \alpha_i(x_g, y_g) \quad (1.83)$$

$$\nabla \alpha_i^g = \nabla \alpha_i(x_g, y_g) \quad (1.84)$$

Consequently, the integrals in (1.80) and (1.81) are computed as

$$K_{ij} = \sum_{T \in \mathcal{T}_h} \sum_g \frac{1}{\mu} \nabla \alpha_i^g J_T^{-1} \cdot \nabla \alpha_j^g J_T^{-1} |J_T| w_g \quad (1.85)$$

$$b_j = \sum_{T \in \mathcal{T}_h} \sum_g [J_{s3} \alpha_j^g + \mathbf{M} \cdot \nabla \alpha_j^g J_T^{-1}] |J_T| w_g \quad (1.86)$$

where  $J_T$  is the Jacobian of the mapping from the reference element to the element  $T$  and  $|J_T|$  is its determinant.

$$J_T = \begin{pmatrix} x_1 - x_0 & y_1 - y_0 \\ x_2 - x_0 & y_2 - y_0 \end{pmatrix} \quad (1.87)$$

## C System resolution

Once the assembling of the stiffness matrix and the load vector based on the contribution of each element of the triangulation is done, the system of equations (1.77) has to be solved. There exist multiple solvers. Direct solvers aim to decompose the matrix  $K$  to the product of two or more matrices that are relatively easy to solve; for example, we can cite Cholesky factorization. However, iterative solvers are still the mostly used type of solvers since they usually lead to a considerable reduction of resolution time and memory.

It is worth noting that the system 1.77 may not be linear in the existence of ferromagnetic material, *i.e.*  $K$  is a function of  $u$ . Iterative solvers like Newton-Raphson or fixed-point algorithms are then used for treating the nonlinearity of the system.

## D Results interpretation

Usually, the solution of the system of equation is not of interest; nevertheless, other quantities computed from that solution are the ones of interest such as the magnetic flux density, the energy, the torque, the electromotive force, ...

These quantities are often used to make "colourful" visualization; these are easy to interpret and give insight about the electromagnetic phenomena in the studied domain.

### 1.3.4 MagFEM - 2D magnetostatic FE code

We have developed a two-dimensional finite element code to investigate the challenges of using such a tool to simulate electrical machines. The developed code is far from being a complete interface for the modelling of electrical machines; however, it will enable us to test the developed approaches on some optimization benchmarks from the literature [1][2].

The general structure of any finite element code is presented in Figure 1.4. At first, the geometry of the device should be created, and material properties are assigned to their corresponding parts. Then a triangulation technique is used to subdivide the domains into small elements. Afterwards, the mathematical model is applied to each of these small elements. The latter are then assembled into a global system of equations. This system is solved to compute the quantities of interest.

We chose to explain these steps using a simple analysis of a device taken from FEMM website [65]. The analysis consist in computing the inductance of the gapped inductor shown in Figure 1.5. The electromagnetic device is composed of an E-shaped ferromagnetic core separated from an I-shaped ferromagnetic core by an air-gap of 0.025" (= 0.635 mm) thick and a winding of 66 turns, a current of 1A

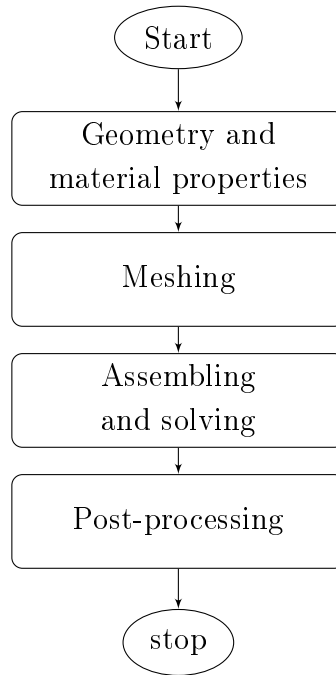


Figure 1.4: Conventional structure of a finite element code

flows in it.

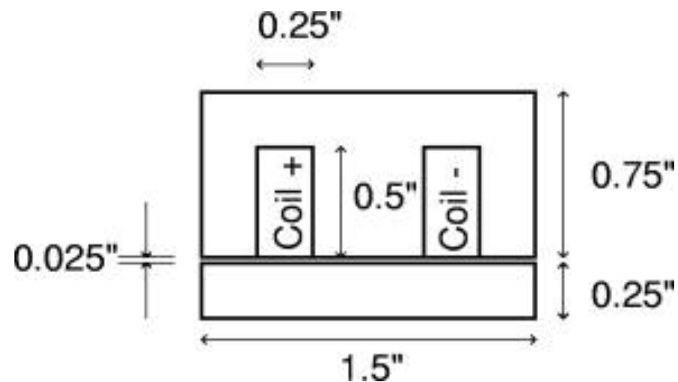


Figure 1.5: Geometry of the inductor [65]

For this model, the ferromagnetic is considered to be linear with a relative permeability of 2500, and we consider that the current is distributed uniformly in the slots.

## A Geometry and material properties

The geometry of the electromagnetic device needs to be created, and material properties are assigned to the corresponding domains. Generally, symmetries may exist in the studied device. In our example, we can model only half of the device since the other half is anti-symmetric. Figure 1.6 shows half of the modelled geometry. Additionally, a boundary domain is defined to impose boundary conditions.

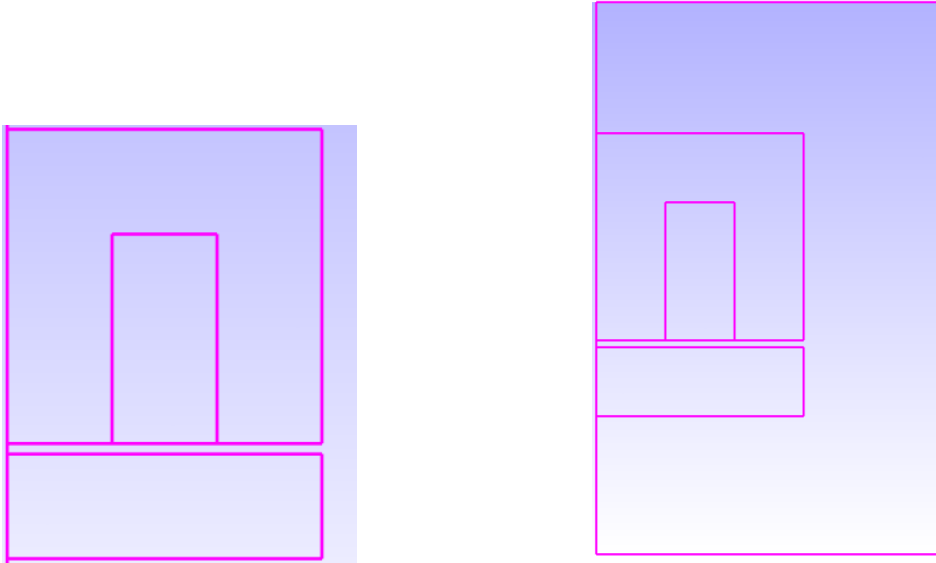


Figure 1.6: Modelled geometry of the inductor (left), Studied domain with boundaries (right)

Generally, the edges and each domain have identifiers that enable to affect boundary conditions, and material properties as shown in Figure 1.7. The edges and the faces are identified by the a number preceded by a letter  $E$  or  $F$  respectively.

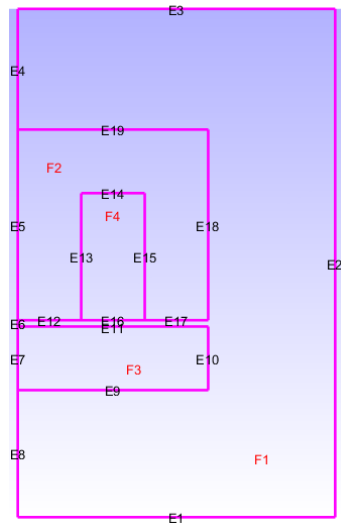


Figure 1.7: Identifiers of geometry

Now, the boundary conditions are assigned to edges from  $E1$  to  $E8$  to impose  $\mathbf{n} \times \mathbf{A} = \mathbf{0}$ . It worth noting that these conditions enable to model the anti-symmetric property and also the boundary of the domain.

Furthermore, we assign a magnetic permeability of 2500 to faces  $F2$  and  $F3$  while we assign a current density of  $0.8184\text{A}/\text{mm}^2$  to face  $F4$ .

## B Mesh

Now, we need to mesh the studied domain. In Figure 1.8, we show two meshes of the studied domain, one with elements that are bigger than the other one. Naturally, the one with smaller elements will lead to more accurate results than the other one. However, it will be more expensive to evaluate.

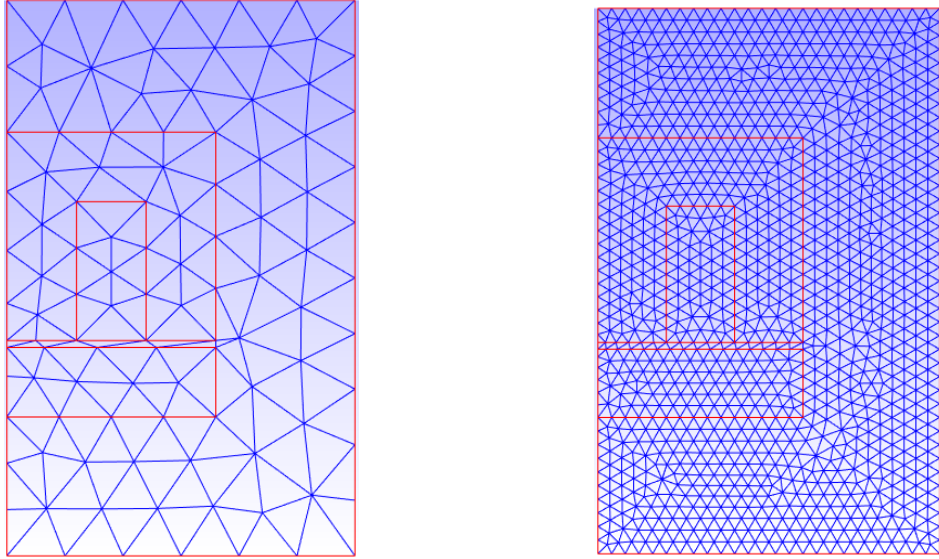


Figure 1.8: Coarse mesh (left), Fine mesh (right)

## C Assembling and solving

Once the material properties are defined and the domain is discretized, the assembling of the matrix and the second member is conducted. Afterwards, the equation is solved; since all the materials are linear, the solution consists in solving a linear system and  $u = K^{-1}b$ .

Figure 1.9 shows the distribution of the magnetic flux density  $\mathbf{B}$  in the studied domain for both the meshes shown in Figure 1.8.

We recall that the flux density  $\mathbf{B}$  is expressed as follows.

$$\mathbf{B} = \begin{pmatrix} \frac{\partial A_h}{\partial x_2} \\ -\frac{\partial A_h}{\partial x_1} \\ 0 \end{pmatrix} \quad (1.88)$$

which can be computed from the solution of the linear system using equation (1.75).

## D Post-processing

As noted before, the solution  $u$  is only an intermediate result; generally, other quantities are computed based on  $u$  are of interest. For example, we have made the plots in Figure 1.9 by computing the magnetic flux density from  $u$ .

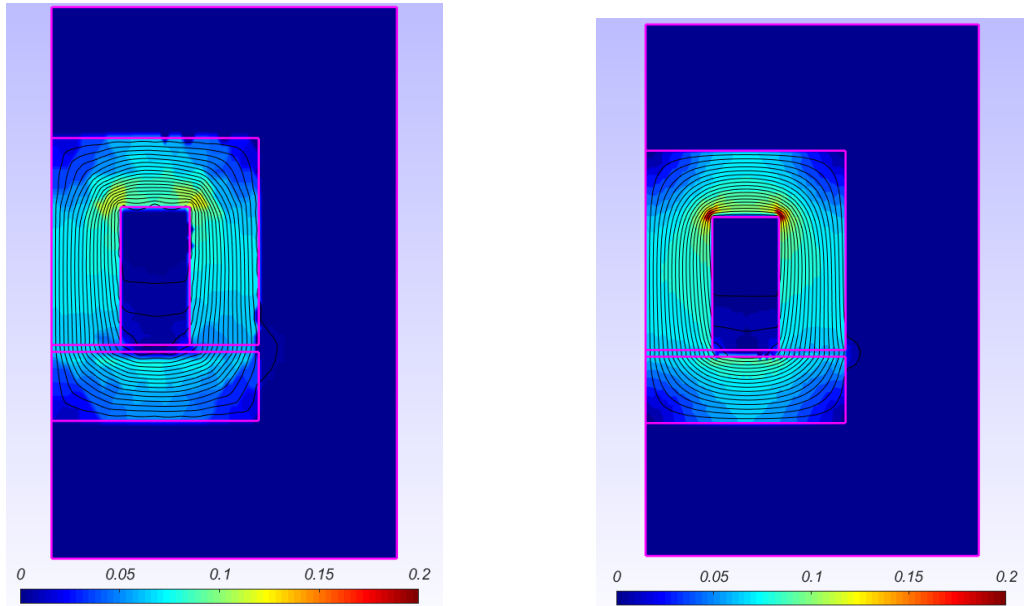


Figure 1.9: Magnetic flux density in the studied domain. Coarse mesh (left), Fine mesh (right)

In this example, we are interested in computing the inductance of the magnetic device. To do so, we use the following formula.

$$W = \frac{1}{2}Li^2 \quad (1.89)$$

where  $W$  is magnetic energy stored in the device,  $L$  is the inductance and  $i$  is the current flowing in the coil.

We recall that the current was chosen to be equal to 1A, Thus, to compute the inductance, we need to compute the magnetic energy using the formula.

$$W = \frac{1}{2} \int_{\mathcal{D}} BHdV \quad (1.90)$$

This quantity is nothing but  $W = \frac{1}{2}u^T Ku = \frac{1}{2}u^T b$ , where  $K$  and  $b$  are resulted from the assembling.

From the reference [65], a value of  $W = 0.865$  mJ is found. In our analysis, we found  $W = 0.779$  mJ for the coarse mesh and  $W = 0.868$  mJ for the fine mesh. The fine mesh gives the closest result to the one from the reference. Then, we can compute the inductance by equation (1.89), we find a value of  $L = 1.736$ mH.

As we solve numerically Maxwell equation, the discretization of the domain, *i.e.* mesh, has an important impact on the solution quality. Ideally, a proper mesh is the one that gives sufficiently acceptable results without being very fine. Thus, some tuning of the mesh is necessary before conducting a finite element analysis.

### 1.3.5 Discussion

In this section, we reviewed the literature related to the electromagnetic modelling of electrical machines starting from the physical model and detailing the mathematical one to present the numerical model finally.

Finite element analysis (FEA) enables one to model the electromagnetic fields in the studied domain finely. FEA is now undeniably the most flexible modelling methodology that is based on a sound mathematical foundation and able to handle different phenomena such as nonlinear material properties, time-varying behaviours and complex geometries.

However, all these advantages come with the expense of the increasing computational cost needed to handle such phenomena. Thus, its usage for optimization, *i.e.* iterative processes, should be made with caution since only a limited number of evaluations of the FEA can be tolerated.

Figure 1.10 shows an objective function  $f$  (function of the flux density) as a function of a geometry parameter  $p$ . An FEA was used to compute the response  $f$  by varying the variable  $p$ . Note that the response is both noisy and has several discontinuities. Numerically, finding the geometry parameter that minimizes the objective function is a design optimization problem that poses several challenges. These discontinuities are explained by the fact that when varying the geometry variable, the topology of the mesh is also changed.

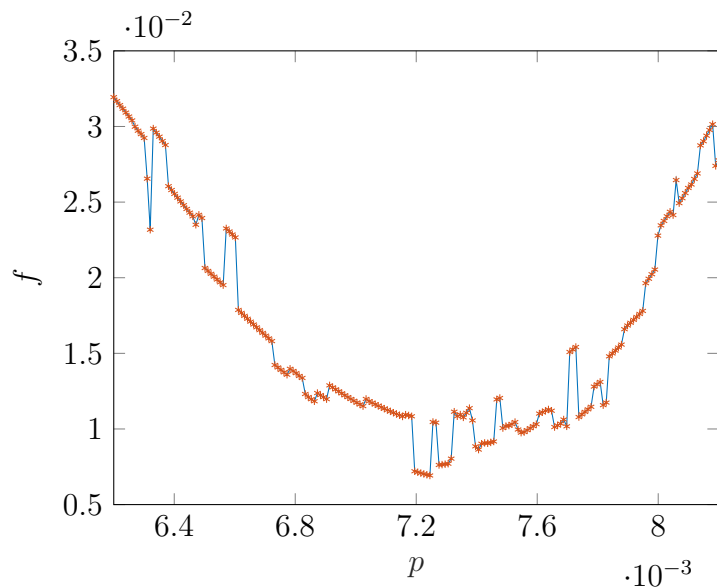


Figure 1.10: Example of extreme re-meshing errors.

Some traditional design optimization techniques use gradient information computed by finite-difference to guide a sequential strategy towards the minimum. The noisy nature of the response shown in Figure 1.10 may render finite difference ineffective at estimating the gradient, thereby preventing the algorithm from successfully finding the minimum.

## 1.4 Chapter Summary

This chapter is considered as the foundation of our research work; at the beginning, we presented the deterministic optimization. Afterwards, the integration of uncertainty related to variables in an optimization loop. Finally, electromagnetic modelling of the electrical machines using the finite element method.

We have highlighted the thesis's main challenges related to the computational cost of the FEA and the numerical error due to re-meshing. These two features limit the choices of well-suited optimization algorithms. Furthermore, we have seen that there are algorithms that are better than others in handling constraints and in treating multi-objective problems.

Consequently, two approaches can be considered, which depend on the capabilities of the model used. A non-intrusive approach where the model is considered as a black-box and surrogate model is fit by evaluating the expensive model at few points, then, use the cheap to evaluate surrogate model for optimization and uncertainty quantification. This approach is detailed in Chapter 2. The second approach (detailed in Chapter 3), which is intrusive where the model evaluates not only the outputs but also their derivatives mainly the first derivative, the derivatives are used either by a derivative-based algorithm or a perturbation method for uncertainty propagation.

The non-intrusive approach using surrogate models is widely used in the context of expensive model optimization. Many researchers had highlighted the challenges of using such a strategy. In this dissertation, we emphasize some of the very known issues and propose new methodologies of using the meta-models and how to speed up the optimization times.

On the other hand, for the intrusive approach, we propose how to compute the derivatives from a finite element model. In this approach, most of the work is done by the model rather than the algorithm. We consider that derivative-based algorithms are cost-efficient, and minor adjustments are needed for improvement, such as a multi-start strategy. The derivatives are efficiently computed from the FEA using the adjoint variable method.





# Chapter 2

## Non-intrusive approach

The term "Black-box" has come into general use as a description or nomination of an undisclosed process or function about which we do not need to know how it operates - just that it does. A black-box is a device, system or object which can be viewed in only through its inputs and outputs (or transfer characteristics), without any knowledge of its internal workings. Its implementation is "opaque" (black). Through this dissertation, the black-box refers to the FEA that models the electromagnetic phenomena of a device, *e.g.* electrical machine.

The non-intrusive approach considers that the FEA as a black-box and all the efforts performed for device optimization is on the optimization algorithms. In the first chapter, we detailed different algorithms for solving optimization problems; all these have pros and cons, but some are more well-suited than others for the problematic dealt with in this thesis. Approximation-based algorithms are of interest since they consider the model as a black-box and fit a meta-model for the optimization process. This approach decouples the black-box model and the optimization process, which enable to reduce the number of calls to the black-box model (expensive) and renders the optimization tractable in a limited time. The meta-model also offers desired characteristics for the most efficient optimization algorithms, such as smoothness and the possibility to compute the gradients since the meta-model is an analytic function. Thus, derivative-based algorithms will excel in optimizing such models.

In this chapter, all the research has been done on how to exploit meta-models in the context of black-box optimization. A discussion of the limitations of the conventional way of usage is conducted, and we propose strategies to tackle these.

### 2.1 Meta-model design optimization MDO

Meta-modelling has led to new areas of research in simulation-based design optimization. Meta-modelling approaches have advantages over traditional techniques

when dealing with the noisy responses and/or high computational cost characteristic of many computer simulations. Meta-models are used in many fields [66], mainly to replace expensive black-box models [67] [68]. In an optimization problem, the objective function and/or constraints are not always cheaply available data. Thus these surrogate models aim to give a model able to approximate the expensive black-box models from a limited number of solutions.

As seen in the first chapter, there is a multitude of approximation techniques in the literature; This dissertation focuses on one particular technique, Kriging. This decision is not baseless. Other researchers such as Jones et al. [33] have made the argument that Kriging provides a suitable approximation for computer simulations and has led a thorough comparison between different approximation techniques.

Optimization using Kriging meta-models was first introduced to tackle unconstrained optimization by Jones et al. [43]. This led to the famous EGO for the Efficient Global Optimization algorithm that forms the basis of this chapter. EGO is classified into a general class of optimization algorithms that is referred to as *Bayesian analysis algorithms* [39].

Meta-model based optimization flowchart is presented in Figure 2.1. The first step aims to determine an initial set of parameter values (initial design) using a design of experiments, *e.g.* Latin Hypercube Sampling (LHS). The black-box(expensive) model is solved for each set of parameters. Afterwards, a meta-model is built based on the initial design and the output data from the black-box. The most important part of the process is to find the infill point (Find new promising samples) in which we look for samples that improve the actual best solution of the optimization algorithm and/or increases the meta-model prediction capabilities. This sample will be evaluated using the expensive model in the next iteration. Finally, some stopping criteria are evaluated to terminate the optimization.

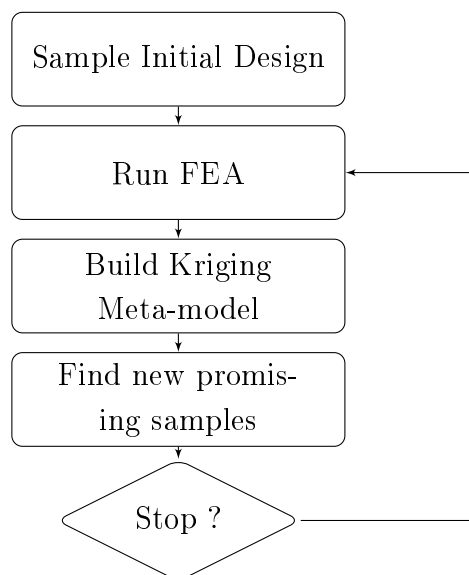


Figure 2.1: Flowchart of Kriging meta-model based optimization.

In this section, we will discuss different steps of the algorithm, starting from initial sampling to stopping criteria, with the corresponding literature and our contributions to the subject. We will use the one-dimensional function

$$f(p) = \cos(0.6p) + \cos(3p) + \sin(6p + 1.5) \quad (2.1)$$

with  $p \in [0, 4]$ .

The function is shown in Figure 2.2 to illustrate the major concepts of how to minimize a black-box model using approximation methods. This function has three local minima and one global minimum (shown as an asterisk in Figure 2.2).

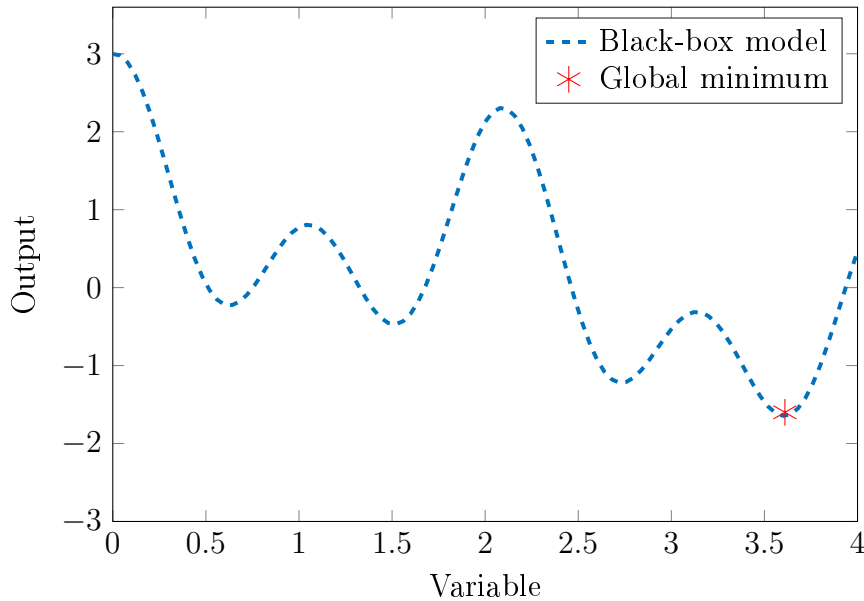


Figure 2.2: Test function

### 2.1.1 Design of experiment DoE

The choice of Design of Experiment (sampling points) plays a critical role in the accuracy of the meta-model and the subsequent use in prediction. The choice of DoE was addressed in the literature by performing empiric comparison on some test cases [69]. Here, we detail two families of design of experiments: classical designs and space-filling designs; moreover, we discuss their adaptability to Kriging models.

#### A Classical DoE

The first family of DoE consists of designs based on geometric considerations. Full-factorial designs and central-composite designs belong to this category. They have been initially developed in the framework of linear regression, *e.g.* quadratic response surface. The idea is to choose the observation points that maximize the quality of statistical inference and minimize the uncertainty of the parameters of the meta-model.

For example, the samples of the full-factorial design are at the boundary of the domain. For the problem defined in Figure 2.2, the samples are  $p^1 = 0$  and  $p^2 = 4$ , *i.e.*; borders of the design space.

In a two-dimensional space, the samples are on the corners of the design space as shown in Figure 2.3.

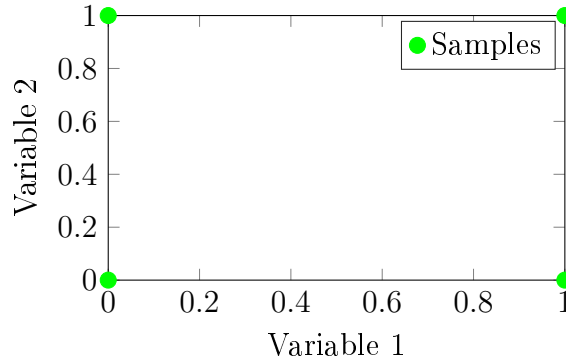


Figure 2.3: Two variables Full-factorial DoE

Although these designs remain reasonable in low dimensions, they require a large number of observations in high dimensions ( $2^{n_v}$ ,  $n_v$  is the number of variables), making them impractical for computationally expensive problems, *i.e.* exponential number of evaluations is needed.

## B Space-filling DoE

A popular alternative to classical DoE is space-filling one. As the name suggests, they aim to spread the sample points evenly on design space. One of the most used designs for the construction of the Kriging response surface is Latin Hypercube Sampling (LHS) [69]. Figure 2.4 shows a two-dimensional sampling using LHS using four samples. All dimensions are subdivided by four intervals, which produces  $4^2 = 16$  sub-spaces. The sub-spaces are sampled in a way that each column and each row contains only one point. Each point is randomly sampled in the sub-space. This enables one to have an acceptable spread of the points on the whole design space. The extension of LHS to  $n$ -dimensional space LHS is relatively easy.

There exist other sampling techniques such as Halton sequence, Hammersley set and Sobol sequence [70]. These methods offer an affordable way of constructing good space-filling DoE, and they are suited for high dimensional problems.

Simpson has performed a comparison of different DoE methods on several test cases [69]. It was found that space-filling designs offer better performances than classical design, and thus, for the rest of this dissertation, we only consider these designs unless stated otherwise.

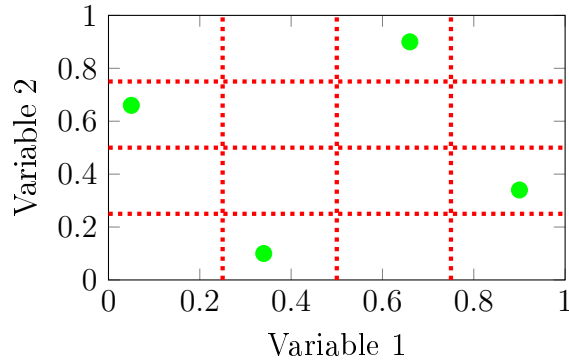


Figure 2.4: Latin hypercube DoE

### 2.1.2 Expensive evaluations

Once the initial DoE is defined, the black-box model is evaluated in the points generated by the DoE. In Figure 2.5, the function was sampled at four points generated using an LHS design.

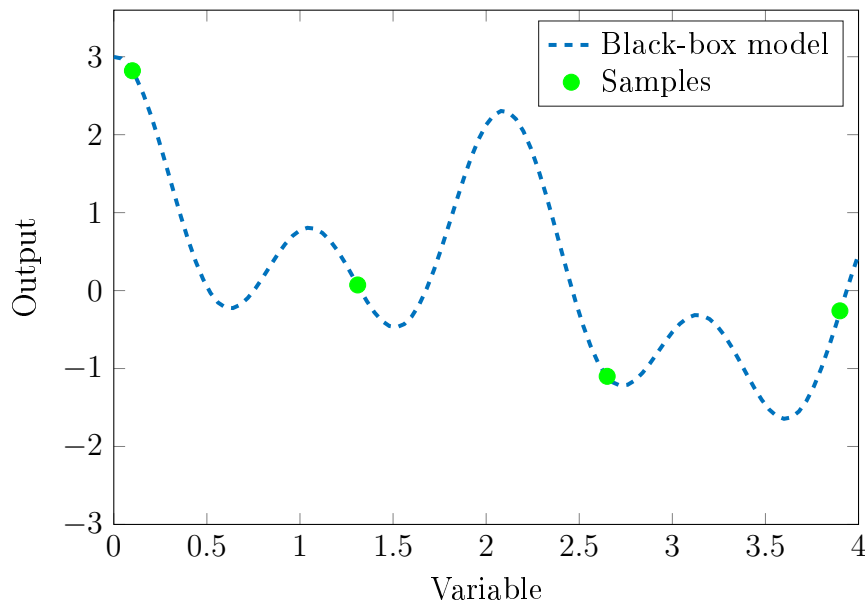


Figure 2.5: Sampled Test function

The evaluation of the black-box model can be performed simultaneously to take advantage of parallel architectures of computers.

### 2.1.3 Fitting Kriging

Kriging has got much interest as a method of curve fitting and prediction. To show this, Figure 2.6 exposes the annual evolution of the usage of Kriging in research papers. The growing trend reflects its usefulness in various engineering and science fields. Kriging offers some highly desired features such as data interpolation and the ability to compute a prediction error.

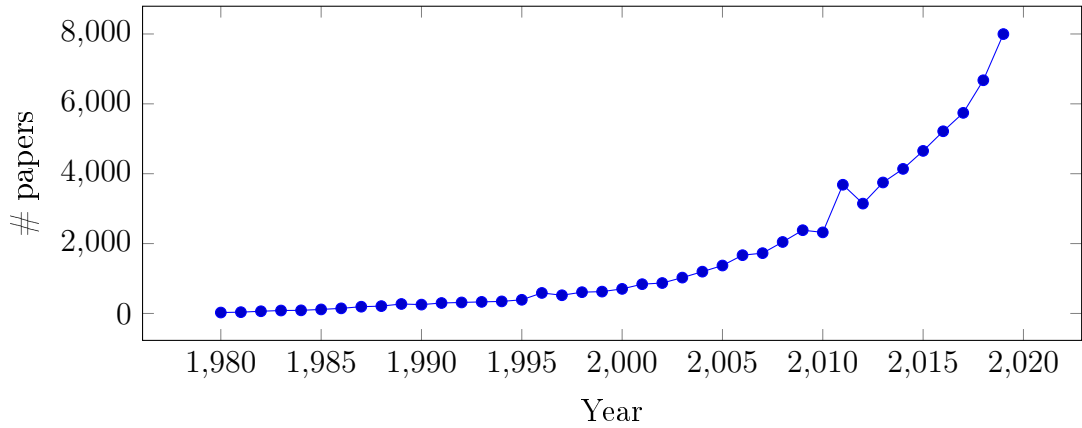


Figure 2.6: The evolution of the number of research papers dealing with Kriging [source: dimensions.ai]

This section is devoted to the development of the equations found in Kriging. We will derive the Kriging formulas using a somewhat different approach. Readers interested in the standard derivation may consult [37][71]. Much of the derivation presented below follows the presentation of Jones [33].

## A Understanding Kriging

Suppose we have two points  $p^k$  and  $p^l$ . Before we have sampled these points using the black-box model  $f(p)$ , we don't know the function values at these points which can be considered as uncertain. Therefore, let us model this uncertainty by a random variable  $F(p)$  of a normal distribution of mean  $\mu(p)$  and variance  $\sigma^2(p)$ . Intuitively, this means that we are saying that the function  $f(p)$  has a typical value of  $\mu(p)$  and is expected to vary in some range like  $[\mu(p) - 3\sigma(p), \mu(p) + 3\sigma(p)]$ .

Furthermore, the function values  $f(p^k)$  and  $f(p^l)$  will tend to be close if the distance between the points  $p^k$  and  $p^l$  is small. Statistically, this means that  $F(p^k)$  and  $F(p^l)$  will be highly correlated if the points  $p^k$  and  $p^l$  are close to each other. Commonly, the exponential function is used to express the correlation between points, and it is given by

$$\text{Corr}[F(p^k), F(p^l)] = \exp\left(-\sum_{i=1}^{n_v} \theta_i |p_i^k - p_i^l|^\alpha\right) \quad (2.2)$$

where  $\theta_k$  and  $\alpha$  are the correlation parameters. Figure 2.7 illustrates the impact of the correlation parameters on the distance between points.

On the left,  $\alpha$  is held fix while varying  $\theta$ . The value of  $\theta_i$  determines how fast the correlation decay as one moves in  $i$ -th dimension. Large values serve to model rapid changes even over small distances. The higher  $\theta$ , the smaller the correlation between two points. It means that the higher  $\theta$  the faster the variation of the function can be between the two points.

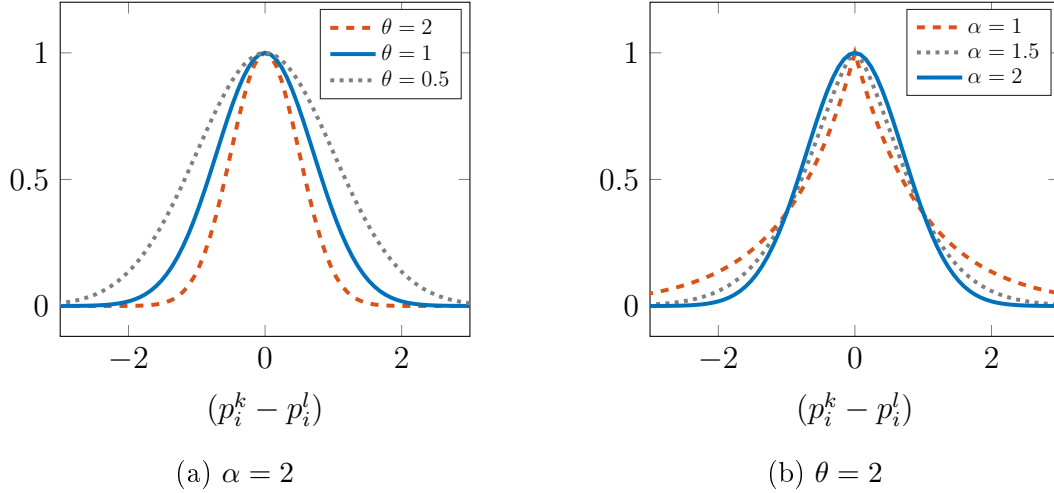


Figure 2.7: Exponential correlation function

On the right, we fix the value of  $\theta$  and vary  $\alpha$ . the value of  $\alpha$  ( $\in [1, 2]$ ) determine the smoothness of the function. The value of  $\alpha = 2$  leads to a Gaussian correlation type that helps to model smooth functions while small values model rough functions.

Going back to the random variable  $F(p)$ , the uncertainty about the black-box model can be represented by sampling the random variable at  $N$  points.

$$\mathbf{Y} = \begin{pmatrix} F(\mathbf{p}^1) \\ \vdots \\ F(\mathbf{p}^N) \end{pmatrix} \quad (2.3)$$

The random vector  $\mathbf{Y}$  has a mean value equal to  $\mathbf{M}$  (vector of  $N \times 1$ ) and covariance matrix equal to

$$\text{Cov}(\mathbf{Y}) = s^2 \mathbf{R} \quad (2.4)$$

where  $\mathbf{R}$  is a  $N \times N$  correlation matrix with  $(i, j)$  element given by equation (2.2) and  $s^2$  is the variance.

The parameters  $\mathbf{M}$ ,  $s^2$ ,  $\theta_i$  and  $\alpha$  characterize how we expect the function to vary when moving in different coordinate directions. These parameters need to be estimated in order to model the behaviour of the function.

The maximum likelihood estimation (MLE) offers one approach to estimate these parameters. The likelihood function is given by

$$\frac{1}{(2\pi s^2)^{\frac{N}{2}} |\mathbf{R}|^{\frac{1}{2}}} \exp \left[ \frac{-(\mathbf{Y} - \mathbf{M})^T \mathbf{R}^{-1} (\mathbf{Y} - \mathbf{M})}{2s^2} \right]. \quad (2.5)$$

One should look for the parameters that maximize the likelihood function (LF). Practically, this means that we look for the predictor that has the same behaviour as the sampled data.

Interestingly, one could easily deduce the values of  $\mathbf{M}$  and  $s^2$  by setting the derivative of the LF with respect to  $\mathbf{M}$  and  $s^2$  to zero. The optimal values can then be expressed as a functions of  $\mathbf{R}$



$$s^2 = \frac{(\mathbf{Y} - \mathbf{M})^T \mathbf{R}^{-1} (\mathbf{Y} - \mathbf{M})}{N} \quad (2.6)$$

$$\mathbf{R}^{-1} \mathbf{M} = \mathbf{R}^{-1} \mathbf{Y} \quad (2.7)$$

The equation (2.7) is written in that form because, generally, the mean cannot capture all the variations of the functions. Otherwise,  $s^2$  is equal to zero, and consequently, there will be no covariance between the sampled random variables; therefore, the Kriging model is meaningless. In the literature, the mean is often assumed to be some polynomial function that captures the global trend of the sampled data. By adopting this hypothesis, the mean  $\mathbf{M}$  is written as

$$\mathbf{M}(p) = \mathbf{f}_{poly}(p)^\top \beta \quad (2.8)$$

where  $\mathbf{f}_{poly}(p)$  is a polynomial vector (generally, polynomials of order two or less), and  $\beta$  are the coefficients of the polynomial vector. Therefore these coefficients  $\beta$  need to be estimated from equation 2.7,

$$\mathbf{R}^{-1} \mathbf{F}_{poly} \beta = \mathbf{R}^{-1} \mathbf{Y} \quad (2.9)$$

$$\beta = (\mathbf{F}_{poly}^\top \mathbf{R}^{-1} \mathbf{F}_{poly})^{-1} \mathbf{F}_{poly}^\top \mathbf{R}^{-1} \mathbf{Y} \quad (2.10)$$

where

$$\mathbf{F}_{poly} = \begin{pmatrix} \mathbf{f}_{poly}(p^1)^\top \\ \vdots \\ \mathbf{f}_{poly}(p^N)^\top \end{pmatrix}. \quad (2.11)$$

Substituting equation (2.6) into equation (2.5), we get the expression of the LF that needs to be maximized.

$$\frac{1}{(2\pi s^2)^{\frac{N}{2}} |\mathbf{R}|^{\frac{1}{2}}} \exp \left[ \frac{-N}{2} \right]. \quad (2.12)$$

Researchers usually refer to the concentrated log-likelihood function (CLLF), in which the log function is applied on LF, and constant terms are ignored. This function simplifies the expression of the LF and changes its range of variation. This will improve upon the performances of the optimization algorithm that solves the problem. The CLLF is written as

$$-\log(s^2) - \frac{1}{N} \log(|\mathbf{R}|) \quad (2.13)$$

The CLLF depends only on the correlation matrix  $\mathbf{R}$  and, hence, on the correlation parameters  $\theta_i$  and  $\alpha$ .

At this stage, we have modelled the sampled data by characterizing the parameters of the correlation function, *i.e.*  $\theta$  and  $\alpha$ . To understand how we can predict at

some point  $p^*$ , we follow the same principle as before. We assume that the parameters of the correlation are known, and we add an artificial sample at  $p^*$  to  $\mathbf{Y}$  that is the unknown predictor  $\hat{f}(p^*)$ . This predictor is found by maximizing the updated LF and is given by

$$\hat{f}(p^*) = \mathbf{f}_{poly}(p^*)^\top \beta + \mathbf{r}(p^*)^\top \mathbf{R}^{-1}(\mathbf{Y} - \mathbf{F}_{poly}\beta) \quad (2.14)$$

where  $\mathbf{r}(p^*)$  is the correlation vector of length  $N$  between the untried  $p^*$  and the sampled data points  $(p^1, \dots, p^N)$ .

The predictor  $\hat{f}$  is made of two terms; the first term is a regression model that enables to capture the global trend of the function, and the second term compensates the residuals produced by the regression; it is often referred as a stochastic term. Therefore, the Kriging predictor interpolates the sampled points.

The mean squared error (MSE) of the prediction can be deduced from the following equation.

$$MSE(p^*) = E \left[ \left( \hat{f}(p^*) - f(p^*) \right)^2 \right] \quad (2.15)$$

The MSE is derived by developing this equation [37], and written as

$$MSE = s^2 \left( 1 + \mathbf{u}^\top (\mathbf{F}_{poly}^\top \mathbf{R}^{-1} \mathbf{F}_{poly})^{-1} \mathbf{u} - \mathbf{r}^\top(p^*) \mathbf{R}^{-1} \mathbf{r}(p^*) \right) \quad (2.16)$$

where  $\mathbf{u}(p^*) = \mathbf{F}_{poly}^\top \mathbf{R}^{-1} \mathbf{r}(p^*) - \mathbf{f}_{poly}(p^*)$ .

Since we have no uncertainty about the point we have already sampled (because we have calculated it using the black-box), the formula for (2.16) has the intuitive property that it is zero at any sampled point. The MSE offers a metric for assessing the meta-model accuracy and can be used to improve it. Indeed, we can sample an additional point where the MSE is high and consequently expecting to improve the global approximation of the function. It will often be convenient to work with the square root of the  $MSE$ , it gives a standard error, or prediction error, for measuring uncertainty in the prediction and referred to as

$$\hat{\sigma}_f = \sqrt{MSE} \quad (2.17)$$

To illustrate the Kriging predictor and its MSE, we use the test function mentioned earlier (2.2). The process of fitting a Kriging model is described step-by-step to foster a better understanding of what is involved in building a Kriging model.

After the function was sampled at the point defined in (2.5), we proceed to the construction of the Kriging meta-model by adopting the following simplifying hypotheses:

- The regression term is first order polynomial

- The correlation function is exponential with parameters  $\theta_1 = 1$  and  $\alpha = 2$

Before fitting the Kriging model to the four sampled points, the variables and responses are scaled, as shown in table 2.1. They are scaled in a way to have a mean of zero and a standard deviation of one, other methods of scaling could be considered. When variables do not have the same order of magnitude, scaling becomes very important to treat all the variables in the same manner and to avoid numerical issues.

No.	$p$	Scaled $p$	$y = f(p)$	Scaled $y$
1	0.1	-1.1489	2.8167	1.4357
2	1.31	-0.41337	0.066438	-0.18091
3	2.65	0.40121	-1.1077	-0.87108
4	3.9	1.1611	-0.27861	-0.38372

Table 2.1: Variable and responses scaling

First of all, we will look for the regression coefficients of  $\beta$ . As we considered the first-order polynomial then

$$\mathbf{f}_{poly}(p) = \begin{pmatrix} 1 \\ p \end{pmatrix}. \quad (2.18)$$

Using the exponential correlation function for the stochastic part of the model, the correlation matrix is particularized for this example as:

$$R_{i,j} = \exp(-|p^i - p^j|^2) \quad (2.19)$$

The resulting correlation matrix is thus

$$\mathbf{R} = \begin{pmatrix} 1 & 0.58214 & 0.09045 & 0.00481 \\ & 1 & 0.51502 & 0.08383 \\ & & 1 & 0.56135 \\ & & & 1 \end{pmatrix}. \quad (2.20)$$

*sym*

The correlation matrix is symmetric and reflects the basic idea discussed at the beginning of this section; close points have a significant correlation, while distant points have a lower correlation.

The regression term of the global model can now be estimated using equation (2.10) :  $\beta = (0.2251 \quad -0.7847)^\top$

Now, new points are predicted using equation (2.14), which is repeated here.

$$\hat{f}(p^*) = \mathbf{f}_{poly}(p^* p^*)^\top \beta + \mathbf{r}(p^*)^\top \mathbf{R}^{-1}(\mathbf{Y} - \mathbf{F}_{poly}\beta) \quad (2.21)$$

where  $\mathbf{r}(p)^\top$  is a the  $4 \times 1$  correlation vector between an untried point and the sampled data points

$$r_i(p^*) = \exp(-|p^* - p^i|^2) \quad (2.22)$$

Notice that  $p^*$  needs to be scaled, as were the variables, before using the predictor, the response also needs to be transformed.

The  $MSE$  can be computed using (2.16), once the parameters of the Kriging model are identified.

A plot of the resulting Kriging model is shown in Figure 2.8 along with the prediction error  $\hat{\sigma}_f$ , the original black-box function, and the four samples.

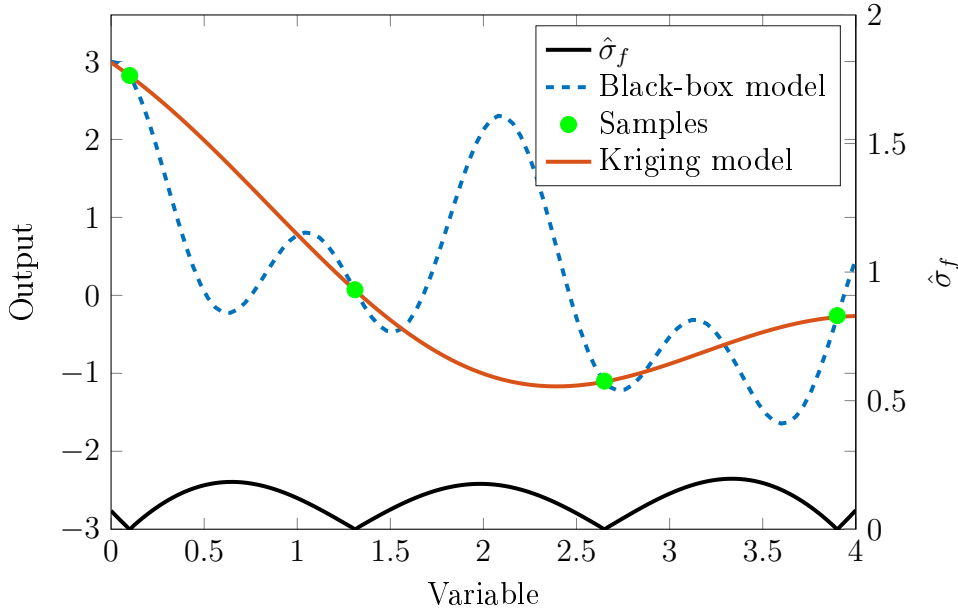


Figure 2.8: Kriging surrogate model

Immediately evident from the figure is the fact that the Kriging model interpolates the sampled points. Furthermore, the prediction error  $\hat{\sigma}_f$  (eq. 2.17) is equal to zero at those points and increases when going far from them.

## B Parameter estimation

Previously, it was noted that Concentrated Log-Likelihood Function needs to be maximized (or minimize its opposite) to determine the parameters of the correlation function  $\theta$  and  $\alpha$  (2.2). Therefore, an optimization algorithm needs to be chosen for this purpose. The optimization problem is given by

$$\begin{aligned} \min_{\theta_l, \alpha} & \left[ \log(s^2) + \frac{1}{N} \log(|\mathbf{R}|) \right] \\ \text{s.t.} & \underline{\theta}_l \leq \theta_l \leq \bar{\theta}_l \\ & 1 \leq \alpha \leq 2 \end{aligned} \quad (2.23)$$

In the example above, we assigned the parameters ( $\theta_1 = 1$  and  $\alpha = 2$ ) to the Kriging model. Here, we will look for the parameters by solving the optimization problem (2.23). First of all, we will look to the evolution of the CLLF versus  $\theta$  while fixing  $\alpha = 2$  as shown in Figure 2.9.

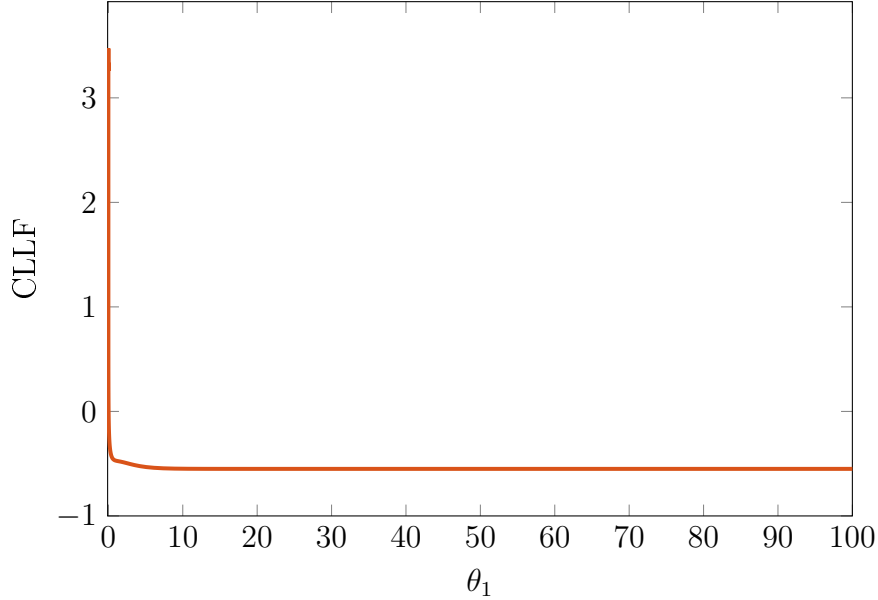


Figure 2.9: Evolution of CLLF with respect to  $\theta_1$

We can see clearly that the CLLF monotonically decreases with  $\theta_1$ , then the "optimal" solution is the maximum allowed value of  $\theta_1$ , *i.e.*  $\bar{\theta}_1$ . With such a large value, the correlation between points will decrease, and the Kriging model will degenerate, producing the behaviour shown in Figure 2.10. We can distinguish that the Kriging model has a linear part corresponding to the first-order polynomial regression term and "bell" shapes around the sampled points to interpolate these.

Obviously, this model is not well suited and should be avoided when optimizing. Sasena explained that the points that are evenly sampled in the design space cause this behaviour and proposed to add new samples near the ones that are already sampled [39] to prevent this erratic behaviour. Furthermore, he proposed a metric to assess if the CLLF is monotonic or not.

An asymptotic study was proposed when  $\theta_1$  approaches infinity. Thus, the quantities mentioned earlier become:

$$\begin{aligned}
 \mathbf{R} &= \mathbf{I} \\
 \beta &= (\mathbf{F}_{poly}^\top \mathbf{F}_{poly})^{-1} \mathbf{F}_{poly}^\top \mathbf{Y} \\
 s^2 &= \frac{(\mathbf{Y} - \mathbf{F}_{poly}^\top \beta)^T (\mathbf{Y} - \mathbf{F}_{poly}^\top \beta)}{N}
 \end{aligned} \tag{2.24}$$

Then, the asymptotic value  $\lim \text{CLLF}$ , when  $\theta$  goes to infinity, is equal to

$$\lim \text{CLLF} = \log \left( \frac{(\mathbf{Y} - \mathbf{F}_{poly}^\top \beta)^T (\mathbf{Y} - \mathbf{F}_{poly}^\top \beta)}{N} \right). \tag{2.25}$$

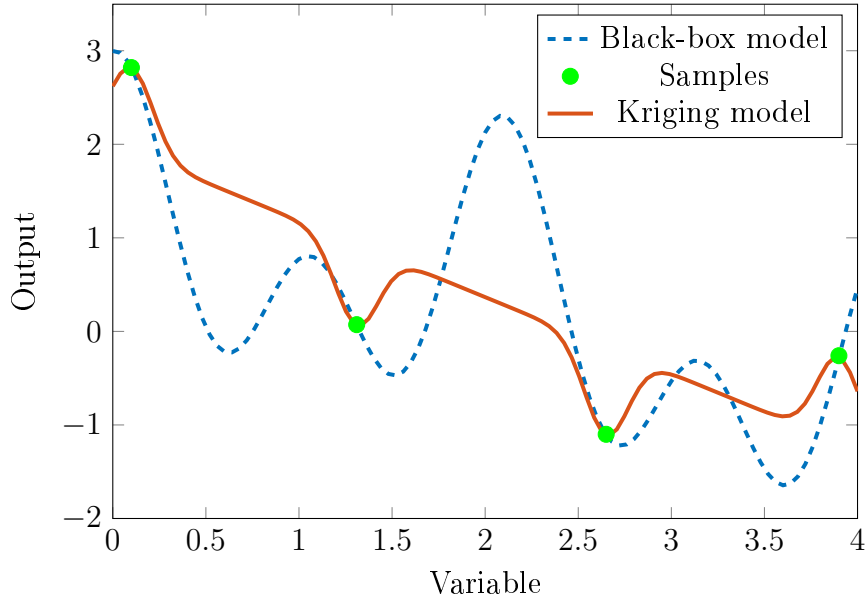


Figure 2.10: Impact of the CLLF on the Kriging model

The optimal value of the CLLF found by the optimization algorithm must be strictly less than  $\text{limCLLF}$ . Otherwise, we conclude that the meta-model is not suitable and special measures should be undertaken. In our example, we found that the optimal value of CLLF is equal to  $\text{limCLLF} = -0.55$ . This explains the poor fitting of the Kriging meta-model.

As proposed in [39], adding additional points near the already sampled ones could improve the meta-model quality. To verify this hypothesis, we considered three configurations. In config. 1, we add a new point  $p = 3.85$  near the point  $p = 3.9$ , in config. 2, we add a point  $p = 2.6$  near the point  $p = 2.65$  and in config. 3, we add a point  $p = 0.85$ .

Config.	Optimal $\theta$	CLLF	$\text{limCLLF}$
1	139.26	-0.64	-0.54
2	11.83	-0.82	-0.52
3	7.57	-0.48	-0.44

Table 2.2: Impact of adding samples on the CLLF

Table 2.2 shows the solution of the CLLF and the corresponding values of  $\text{limCLLF}$ . We can see that adding points near to the sampled ones does not always solve the problem, as stated in [39]. In config. 2, we can notice that indeed the CLLF is not monotonic because the optimal value of CLLF is less than  $\text{limCLLF}$ , and we have a minimal value at  $\theta = 139.26$ . This will have the same behaviour as observed in Figure 2.10; A large value of  $\theta$  is not very desired since it produces a model with as many local extrema as the number of samples.

Nevertheless, adding a sample point has solved the problem for the config. 2, it

has produced an acceptable value of  $\theta$  while verifying the condition that the optimal value of CLLF is less than  $\text{limCLLF}$ .

The config. 3 shows that points other than the ones near the already sampled point could prevent the monotonic behaviour of the CLLF.

Figure 2.11 shows the evolution of CLLF with respect to  $\theta$  in config. 3, and the location of the optimal value. Furthermore, Figure 2.12 shows the Kriging meta-model that corresponds to this configuration.

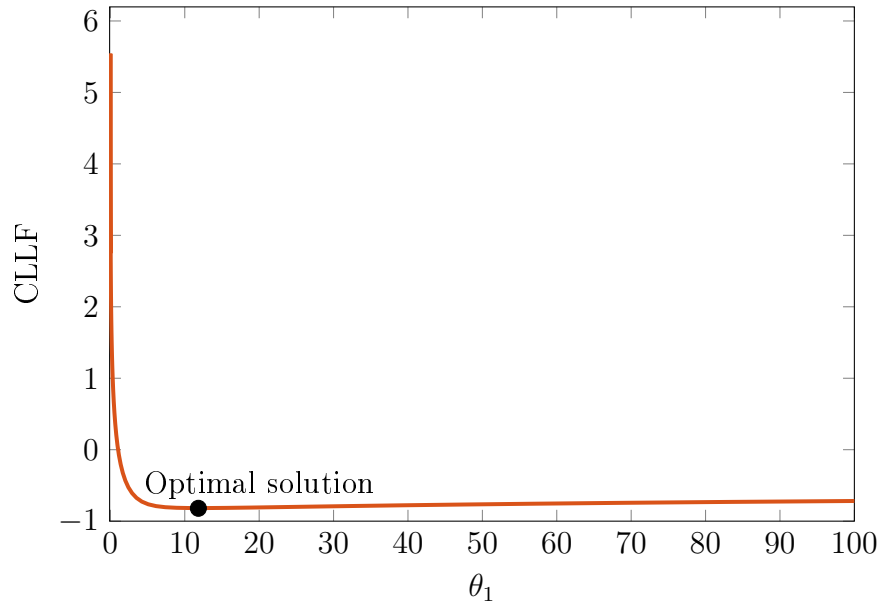


Figure 2.11: Evolution of CLLF with respect to  $\theta_1$  for config. 2

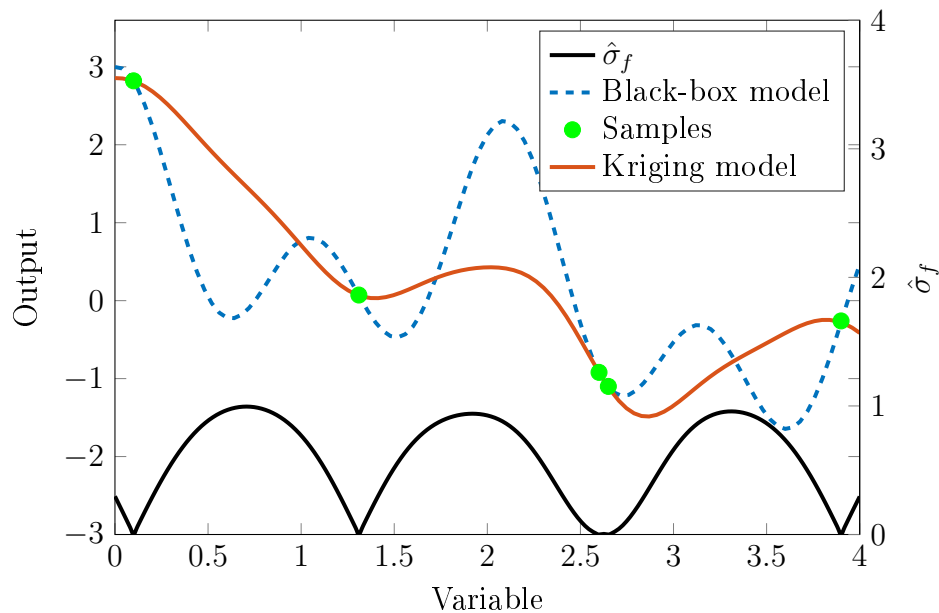


Figure 2.12: Kriging surrogate model with optimized CLLF

These observations beg the questions: What can one do to prevent the mono-

tonic behaviour of the CLLF? Furthermore, where to add the sample points? Two measures should be considered. The first one is to limit the upper value of the  $\theta$  to guarantee a minimal correlation between points. The second one is a diagnosis to test the optimal value of CLLF compared to the limCLLF; Points should be added sequentially until the diagnosis is satisfied.

As we have shown before, adding a single point near sampled one is not sufficient to improve upon the short-range variability, and therefore the CLLF may still be monotonic. Hence, additional points should be added until the condition mentioned above is verified. In the next section, we will discuss how to find those points and add them to the sampled ones.

On the other hand, the maximal value of  $\theta$  should be imposed, but how?

Considering the correlation function, we want at least two sampled points to be correlated with each other, and their correlation has to be higher than some threshold as shown in the equation below

$$\exp(-\theta|p^k - p^l|) \geq c \quad (2.26)$$

In most cases, the samples of the initial DoE are uniformly distributed on the design space, after scaling, all the variables will be bounded between  $-\sqrt{3}$  and  $\sqrt{3}$ . Furthermore, they are almost equidistant in each dimension; thus, the distance between two samples is around

$$d = \frac{2\sqrt{3}}{N+1} \quad (2.27)$$

where  $N$  is the number of samples. Thus, by fixing the strength of the correlation  $c$  we can find an upper value of  $\theta$ .

$$\theta \leq -\frac{\log(c)}{d} \quad (2.28)$$

A reasonable choice of  $c = 0.003$  than will lead to an upper bound of  $\theta$  which is equal to

$$1.68(N+1).$$

It is worth noting that for  $n_v$ -dimensional problem, the upper bound of  $\theta$  in each dimension should be

$$\bar{\theta} = 1.68 \frac{N+1}{n_v} \quad (2.29)$$

The upper bound depends only on the size  $N$  of the initial design and the dimensionality of the problem. Imposing an upper bound on  $\theta$  will guarantee that, in the worst case, at least some small correlations between the samples are imposed.

We can follow the same reasoning for the estimation of the lower bound on  $\theta$ . Imposing a maximal possible correlation between two samples to  $c = 0.997$  will lead



to the lower bound of  $\theta$  in each dimension:

$$\underline{\theta} = 10^{-3} \frac{N + 1}{n_v} \quad (2.30)$$

Now, we fully identified the optimization problem formulated in (2.23), one should choose an optimization algorithm to solve it. As the evaluation of CLLF requires inverting the correlation matrix, which becomes computationally expensive for a large number of samples (size of  $\mathbf{R}$  is equal to the number of samples). A derivative-based algorithm can be used, but the CLLF presents a plateau where the gradient is quasi-null, as shown in Figures 2.9 and 2.11. This will lead to slow convergence of derivative-based algorithms. As an alternative, we use the Nelder-Mead algorithm that requires only one or two evaluations per iteration and does not require derivatives.

The Nelder-Mead algorithm has two drawbacks; handling constraints and finding an adequate initial point. The problem formulated in (2.23) has only bound constraints; these can be easily integrated into the objective function by applying a variable transformation. Nevertheless, the initial point requires more attention since the CLLF can be multi-modal (many local minima) [72].

The main idea for finding an adequate initial point is similar to the one used for identifying the upper and lower bounds of  $\theta$ . We define the initial point, such as the most distant samples are correlated. This will lead to an overestimation of the correlation strength with a small value of  $\theta$ , where the CLLF exhibits large gradients.

The value of  $\alpha = 2$  is best suited for smooth functions but less numerically robust [39]; that is why we choose the initial value of  $\alpha = 1$ . Then, we assume all the samples are correlated by at least a small factor  $c$ . This small correlation corresponds to the most distant samples. As the variables are scaled, this distance between the farthest points is at most equal to  $2\sqrt{3}$ . Then, the initial point of  $\theta^0$

$$\theta^0 = -\frac{\log(c)}{2\sqrt{3}} \quad (2.31)$$

A correlation factor  $c = 0.25$  seems a good value since our experience have shown that underestimated values of  $\theta^0$  usually gives better results than overestimated ones.

$$\theta^0 = 0.4 \quad (2.32)$$

Finally the optimization problem that maximizes the CLLH to find the correlation parameters is written as

$$\begin{aligned}
 & \min_{\theta_l, \alpha} \left[ \log(s^2) + \frac{1}{N} \log(|\mathbf{R}|) \right] \\
 \text{s.t. } & 10^{-3} \frac{N+1}{n_v} \leq \theta_l \leq 1.68 \frac{N+1}{n_v}, \quad l = 1, \dots, n_v \\
 & 1 \leq \alpha \leq 2
 \end{aligned} \tag{2.33}$$

where  $N$  is the size of the initial design and  $n_v$  is the number of variables.

After fitting the Kriging model, additional sample points are added to improve its quality either by looking for the optimum of the Black-box model or globally improving its accuracy. Next, we discuss how we determine the sample points for both approaches.

### 2.1.4 Finding new samples

Optimization algorithms using the Kriging model do not follow a search path as conventional algorithms; it selects points from anywhere in the design space depending on the objective; it is referred to as infill sampling criterion (ISC). To illustrate this, we will start with the famous algorithm EGO developed by Jones et al. [43].

EGO exploits the Expected Improvement (EI) infill sampling criterion. EI was first introduced to tackle unconstrained optimization. Its main advantage lies on sampling points that offer a good trade-off between local exploitation and global exploration of the design space. Indeed, It tends to choose the design points most likely to improve the accuracy of the model and/or have a better function value than the current best point.

Thus, EGO proposes to maximize the expected improvement function, shown in equation ( 2.34), rather than minimizing the meta-model of the objective function.

$$EI(p) = \left( f_{min} - \hat{f}(p) \right) \Phi \left( \frac{f_{min} - \hat{f}(p)}{\hat{\sigma}_f(p)} \right) + \hat{\sigma}_f(p) \phi \left( \frac{f_{min} - \hat{f}(p)}{\hat{\sigma}_f(p)} \right) \tag{2.34}$$

where  $\hat{f}$  and  $\hat{\sigma}_f$  are the Kriging meta-model and the prediction error of the objective function respectively.  $f_{min}$  is the smallest sampled value of  $f$ .

Maximizing  $EI(p)$  leads to the point  $x^*$  with the most significant expected improvement under the uncertainty of the Kriging model, either by sampling toward the optimum or improving the approximation of the meta-model. These characteristics can be justified by the fact that the derivative of EI with respect to  $\hat{f}$  is negative, meaning that the smaller  $\hat{f}$ , the higher EI (exploitation) and the derivative EI with respect to  $\hat{\sigma}_f$  is positive, meaning that the bigger  $\hat{\sigma}_f$ , the higher EI (exploration).

To demonstrate EGO's search strategy for unconstrained optimization problems, the one-dimensional test function shown in Figure 2.2 is used. EI determines where

the algorithm will evaluate the black-box function. It tends to choose the design points most likely to improve the accuracy of the Kriging model and/or have a better value than the current.

In Figure 2.13, The blue dashed line is the true objective function we wish to minimize, while the solid red line is the Kriging approximation conditional to the sample points shown as green dots. The black plot at the bottom is the sampling criterion (EI); its y-axis is on the right.

After the initial sample of four points generated by LHS, the resulting Kriging model is of poor fit compared to the true function. Nevertheless, the EI function leads the algorithm to sample points where it is expected to improve the actual smallest value of the objective function while taking into account the prediction error. After three iterations, the model has improved in the region of the optimum on the right. In iterations 5 and 6, EI finds other points in regions where there is a high probability of finding a better point. By iteration seven, all the region on the right has been explored. However, the left part is not accurate compared to the true function because the Kriging model expects no-improvement from sampling in that region. Nevertheless, the global minimum has been found quite accurately as shown in Figure 2.13.

From this example, one can see that optimization using the Kriging model does not follow a search path. It chooses points from different regions in the design space, depending on where the sampling criterion is the highest.

In general, ISCs have strong influence on how efficiently and accurately the algorithm locates the optimum. They can be tuned to focus either on exploitation and exploration, or a compromise between the two. An exhaustive set of infill criteria was presented in [39].

## A Constraints handling

Constraints are omnipresent in most optimization problems, having ways to deal with these is mandatory. EGO was developed for the optimization of simply bounded problems, that is, with constraints only on the ranges of the design variable values to be considered. Some researchers extended the EGO algorithm to constrained problems. The probability of feasibility ( $PF$ ) criterion is one of the most significant contribution. It quantifies the probability that a constraint is satisfied ( $g(x) \leq 0$ ).

$$PF(x) = \Phi\left(\frac{-\hat{g}(x)}{\hat{s}_g(x)}\right) \quad (2.35)$$

where  $\hat{g}(x)$  and  $\hat{s}_g(x)$  are the Kriging meta-model and the prediction error of the constraint  $g(x)$  respectively.

The ISC is then adapted to consider both objectives; to sample point that improves the actual solution and verify the constraints. Thus, both  $EI$  and  $PF$  should be considered.

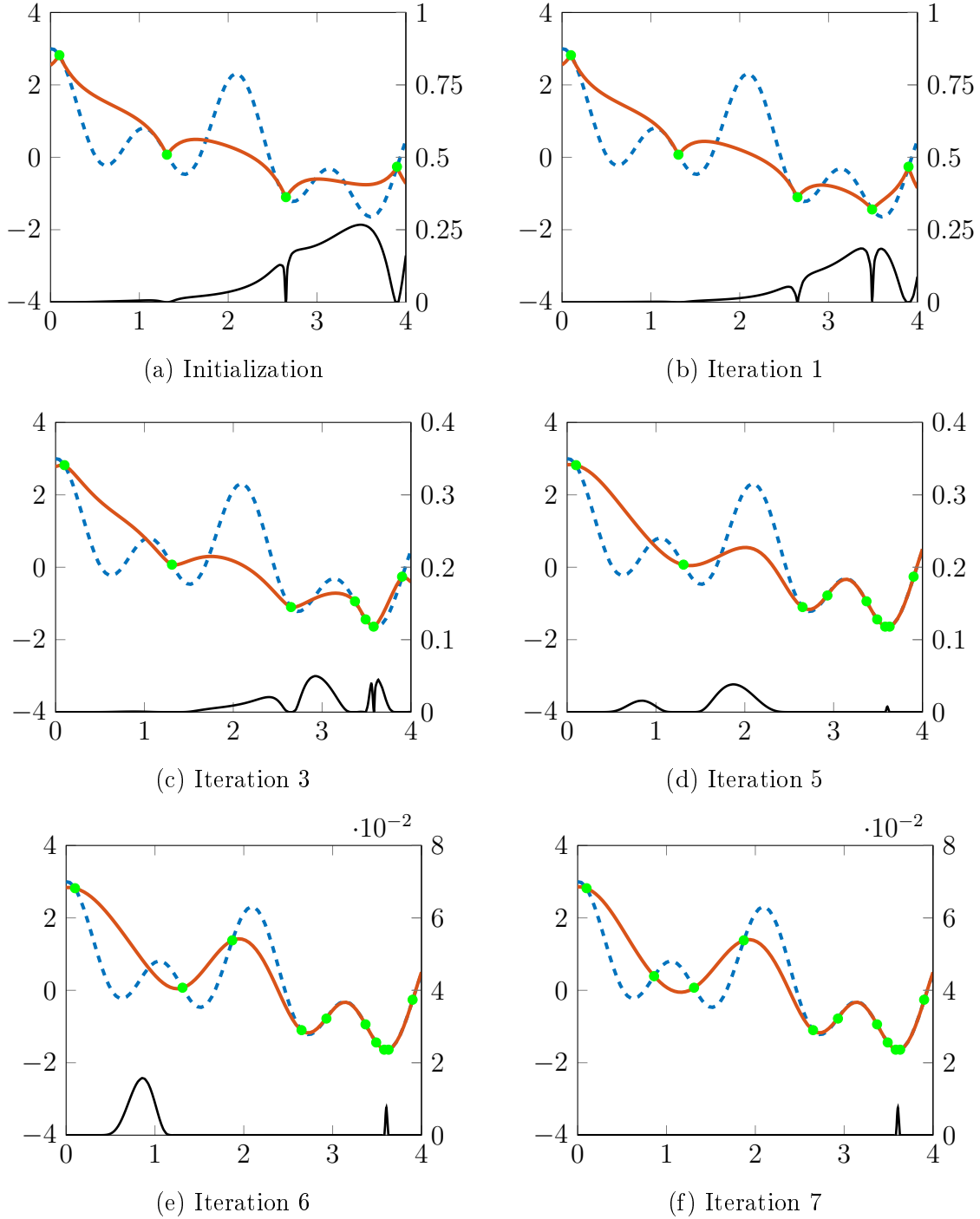


Figure 2.13: Iterations of EGO

The formulation (2.36) has a thoughtful statistical derivation (assuming that the objective function and the constraint are statistically independent). It aims at looking for points that maximize the expected improvement and fulfill constraints. However, the product of  $EI$  and  $PF$  reveals high modality and algorithms maximizing this criterion often fail to find the global optimum.

$$\max_p EI(p)PF(p) \quad (2.36)$$

Thus, another formulation was proposed as in equation (2.37) that considers the infill criterion as bi-objective to reduce the modality [73]. From the Pareto front solution, the point chosen is the one that maximizes the product of  $EI$  and  $PF$ .

$$\max_p [EI(p), PF(p)] \quad (2.37)$$

In the case of many constraints, both formulations consider the global  $PF$  as the product of the probability of feasibility of each constraint. However, this prevents the search close to the boundary of the constraint. So, if the optimum lies on the constraint boundaries, these infill criteria may fail to find it. The maximization of  $PF$  leads to points inside the feasible region and less likely on the constraint boundaries.

Therefore, a new formulation was proposed in equation (2.38). It considers the ISC problem as a constrained one to reduce the modality of the infill criterion and to gain in precision of the solution in case of the optimum lies on the boundary of some constraints [39].

$$\begin{aligned} \max_p EI(p) \\ s.t. PF_i(p) \geq P_{tol}, \quad i = 1 \dots n_c \end{aligned} \quad (2.38)$$

This formulation considers each constraint independently and calculates their respective probabilities of feasibility, ending up with the same number of constraints as the original problem.

A value  $P_{tol} = 0.95$  was recommended but it has an effect on the precision. It fails to locate the points on the constraint boundaries. In our opinion, a value  $P_{tol} = 0.5$  seems more reasonable because  $PF = 0.5$  when  $\hat{g}(x) = 0$ .

To illustrate the sampling strategy for constrained problems, we will add a constraint to our test function. The optimization problem is now written as follows:

$$\begin{aligned} \min_p f(p) &= \cos(0.6p) + \cos(3p) + \sin(6p + 1.5) \\ s.t. g(p) &= \cos(2p) - 0.4 \leq 0 \\ 0 &\leq p \leq 4 \end{aligned} \quad (2.39)$$

In Figure 2.14, the problem is shown graphically. The objective function is the blue dashed line, the infeasible region, where  $g(p) > 0$ , is highlighted by the gray area, and an asterisk shows the global minimum at  $p = 3.721$ . On this figure, we can see that there are two separate feasible zones, one at the centre and the other at the right where the optimum lies.

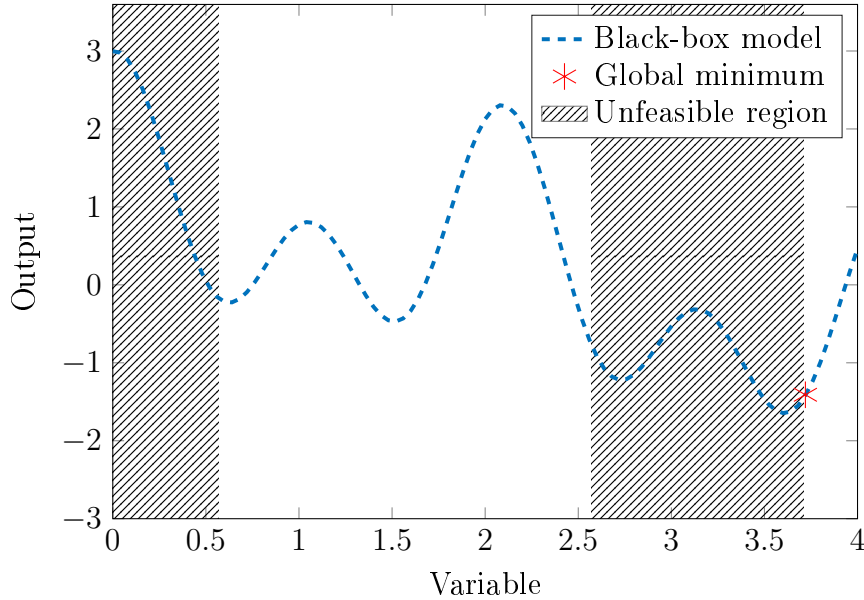


Figure 2.14: Constrained test problem

This constrained test problem is used to show the iterations of the constrained ISC presented in 2.36. In Figure 2.15, The blue dashed line is the true objective function we wish to minimize, while the solid red line is the Kriging approximation conditional to the sample points shown as green dots. The black plot at the bottom is the sampling criterion 2.36; its y-axis is on the right. The Kriging model of the constraints was not put in the figures for better legibility. Visually, we can see that the Kriging was able to approximate quite accurately the constraint after three iterations.

The ISC was able to locate the global solution quite accurately as for the unconstrained case. The new samples are found by maximizing the ISC. In the beginning, the algorithm explores the right region where the optimum lies. In contrast, at the last iterations, it explores the left region where there is a high probability that a better point can be found.

Sasena et al. have discussed that transforming the ISC to a constrained problem, as shown in (2.40), highly improves the accuracy of the solution found [74]. The ISC for constrained problems is then written as

$$\begin{aligned} \min_p EI(p) \\ \text{s.t. } \hat{g}_i(p) \leq 0, \quad i = 1 \dots n_c \end{aligned} \quad (2.40)$$

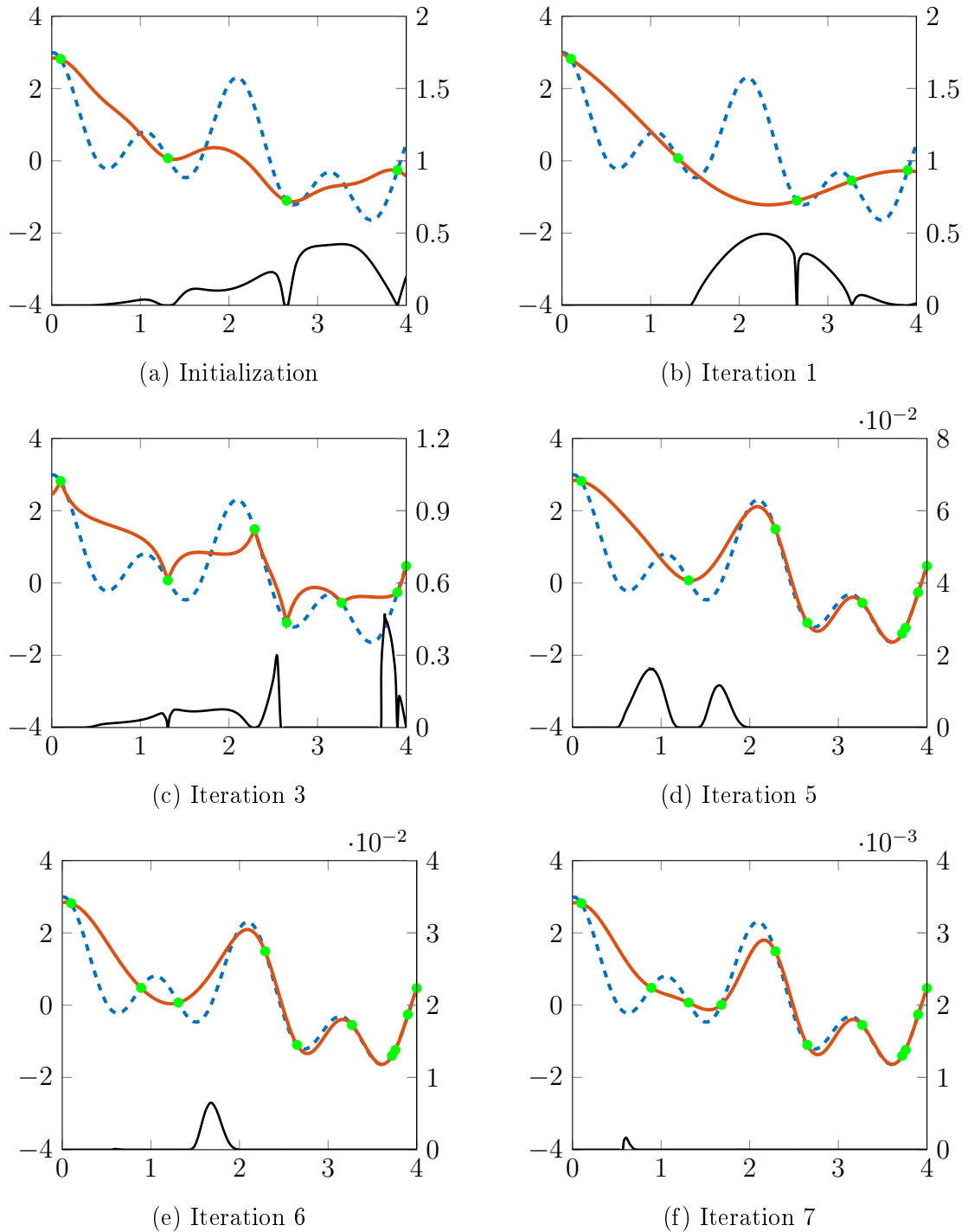


Figure 2.15: Iterations of constrained EGO

Furthermore, they noted that the uncertainty on the constraints meta-model does not bring any gains upon the accuracy of the solution. But, it may play an inverse role by forcing the sampling criteria to sample either in the unfeasible or the feasible regions rather than on the boundary itself where the optimum may be found. They have shown this behaviour on different test cases and recommended using only the Kriging model as a measure of constraint satisfaction.

## **B Extensions to multi-objective optimization**

Treating multi-objective problems with approximation methods is still an active research area, as noted by Rojas et al. [75]. There exist different ways to treat multi-objective optimization using an approximation method such as Kriging. We will distinguish here two significant categories:

1. Infill criteria-based
2. Performance indicator-based

The infill criteria-based category aims to extend the infill sampling criteria from single-objective problems to multi-objective ones. The expected hypervolume improvement is commonly used; it extends the concept of the expected improvement to multi-objective optimization [76].

On the other hand, the performance indicator-based category is used as an assistant to some multi-objective optimization algorithm. For example, K-MOGA uses the prediction error to assist a genetic algorithm [77]. At each generation, a Kriging model is fitted to each objective and used to evaluate each point in the population. If the prediction error is higher than some threshold for any point in this population, the black-box model is used on that point to return the true response values.

It is worth recalling what we presented in Chapter 1, namely that a multi-objective problem could be transformed into a single-objective one (using weighted-sum or  $\varepsilon$ -constraint) and treat it as a single-objective problem.

## **C Extensions to optimization under uncertainty**

Optimization under uncertainty using approximation methods is scarcely treated in the literature because of the double complexity of optimization and uncertainty propagation, Deng has presented a thorough state-of-the-art on the subject [49]. Most of the formulations that deal with uncertainty are extended from the deterministic case. In fact, new ISCs are developed to explore only promising regions of the design space. This research topic has been gaining importance during the last two decades. Many researchers proposed their methods to reduce the computational burden and improve the design while taking into account the uncertainties [78].



## D Solving the infill criteria problem

It was noted that new samples are found by solving the ISC optimization problem. Although different, ISCs often reveal striking similarities. One of the most common is multi-modal behaviour. Indeed the ISC has many local optima. Finding a local minimum is not necessarily a critical matter, but finding the global one is very desired. For example, in Figure 2.13, the expected improvement function has many optima and large regions where it is equal to zero. Optimization algorithms can find the optimum in the first iterations. However, in the last iterations, the non-zero region is tiny, which renders finding the global optimum a very delicate task.

Thus, the choice of the optimization algorithm is a crucial part of the success of the whole process. As noted, the ISC can have multiple local optima and can have zero-gradient on large parts of the design space. Thus, derivative-based algorithms are not well-suited. Nevertheless, the meta-model has an analytic formula and gradients can be computed with little overhead. We recall that derivative-based algorithms are the most efficient when the function and its derivatives are smooth enough, as detailed in Chapter 1. Consequently, we choose to bet on these algorithms for two main reasons:

1. The local search drawback, can be dealt with by adopting a multi-start strategy. Indeed, running multiple optimization with uniformly distributed initial points on the design space will necessary increase the chances of finding the global minimum.
2. Derivative-based algorithms deal efficiently with constraints to enable us to treat constrained ISC problems.

We choose to use the SQP implementation of MATLAB for this purpose. The initial points are generated using an LHS to guarantee a uniform distribution. Additional measures were taken to avoid some specific problems, such as scaling the variables and the responses.

### 2.1.5 Stopping criteria

Proving the convergence of the approximation-based algorithm is very difficult. For most algorithms, the search is terminated by a limit on the number of black-box evaluations. Other stopping criteria have been investigated that involve some statistical property. For example, the expected improvement estimates how the sampled point will improve the actual solution. Thus, a threshold can be defined to stop the algorithm if the maximal value of the expected improvement is less than that threshold.

Nevertheless, these criteria might stop the algorithm prematurely since the optimization algorithm can sometimes fail to find the global optimum of the ISC.

In the remaining of this chapter, the termination criterion is specified as the maximum allowable number of black-box model evaluations unless we state different conditions.

### 2.1.6 Analytical examples

The performances of the algorithms presented above are tested on two analytical examples.

#### A Example 1: Dimensionality

The first example is the Rosenbrock function; H. Rosenbrock introduced it in 1960, which is used as a performance test problem for optimization algorithms. As shown in (2.41), the problem can be extended to many variables; thus, the objective, of using this function, is to study the impact of dimensionality on EGO.

$$\begin{aligned} \min_p f(p) &= \sum_{i=1}^{n_v-1} \left[ (1 - p_i)^2 + 100 (p_{i+1} - p_i^2)^2 \right] \\ \text{s.t. } & -2.4 \leq p_i \leq 2.4, \quad i = 1 \dots n_v \end{aligned} \quad (2.41)$$

Rosenbrock problem has one global minimum at  $p^o = (1, \dots, 1)$  and  $f(p^o) = 0$ . Figure 2.16 shows the contour plot of the function and the global solution is as asterisk.

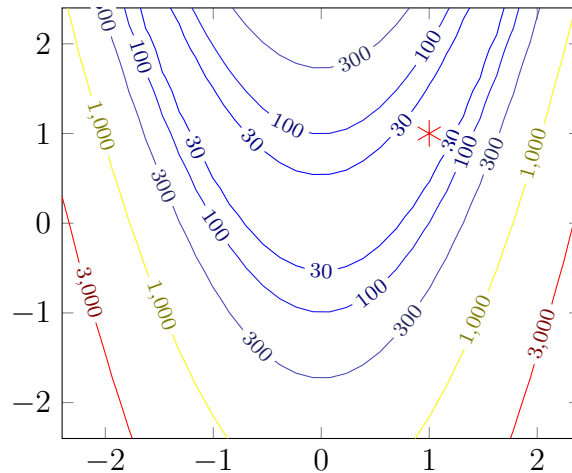


Figure 2.16: Two-dimensional Rosenbrock function

#### B Example 2: Constraints

The second example aims to study constraint handling; therefore, we consider the optimization problem shown in (2.42). Sasena considered this problem as one of his test problems [39].

$$\begin{aligned}
 \min_p f(p) &= (p_1 - 10)^2 + 100(p_2 - 15)^2 \\
 \text{s.t. } g(p) &= \left( p_2 - \frac{5.1}{4 * \pi^2} p_1^2 + \frac{5}{\pi} p_1 - 6 \right) + 10 \left( 1 - \frac{1}{8\pi} \right) \cos(p_1) - 5 \leq 0 \\
 &- 5 \leq p_1 \leq 10 \\
 &0 \leq p_2 \leq 15
 \end{aligned} \tag{2.42}$$

The problem has a global optimum at  $p^o = (9.7775, 4.5238)$  and the constraint is active  $g(p^o) = 0$ . Figure 2.17 represents the cost function and the constraints. The objective function is shown as contour lines; the region inside the black curves is the feasible region while outside is not. This problem has three minima, one in each feasible region. A red asterisk shows the global one.

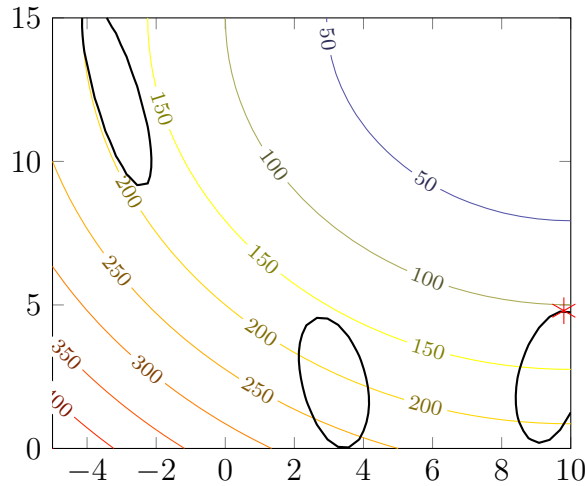


Figure 2.17: Constrained function

### 2.1.7 Results assessment

We choose to consider these examples in order to test EGO algorithms and assess the quality of the solution and the computational burden expected when treating similar problems. Thus, some specific metrics are proposed to quantify the performances.

1. *cost* : The number of black-box function evaluations.
2. *cost*<sub>1%</sub> : The number of black-box function evaluations before a point is sampled within a box with size 1% of the design space range around the true solution.
3. *error* : the normalized Euclidean distance from the best sampled point to the global solution. This quantity is a scaled to design size.
4. *er*<sub>hat</sub> : The normalized Euclidean distance from the solution found by the Kriging model and the global solution.

To reduce the impact of the initial sampling on the result, we choose to use the Hammersley sequence rather than LHS as an initial DoE; it generates well-spread points. The size of the initial sample is chosen to be five times the number of variables.

The Kriging meta-model is fit at each iteration; the parameters of the correlation function are found using the Nelder-Mead algorithm. The upper and the lower bounds and the initial point of the correlation function are determined using the procedure described in the previous section mainly equation (2.33).

We consider the expected improvement criterion as an ISC for the first example and for the second example, the expected improvement is subjected to the meta-model of the constraint. We use Matlab implementation of the SQP algorithm to solve the ISC problem. One thousand points were chosen using an LHS to run SQP using different initial points. This significant number of initial points will enable us to determine a convergence rate. The latter is determined by computing the number of optimization that converged to the best solution (within a radius of  $10^{-8}$ ) divided by the total number of optimizations.

As for the stopping criterion, the iterations are kept going until the maximum value of the expected improvement is less than a threshold for two successive iterations. Imposing two iterations will prevent the algorithm from stopping prematurely in case the optimization algorithm fails to solve the ISC problem once. The threshold is taken to be equal to  $10^{-6}$ .

### A Example 1

Four optimizations are considered with different number of variables  $n_v$ . The results are summarized in Table 2.3.

$n_v$	$cost$	$cost_{1\%}$	$error \%$	$error \%$
2	27	-	1.5	0.06
3	45	-	2.2	11.5
4	87	-	8.2	3.0
5	157	-	3.5	3.6

Table 2.3: Performances of EGO on the multidimensional Rosenbrock problem

The total number of evaluations increases as the number of the variable does. Interestingly, for this problem, the number of evaluations is almost proportional to the power of two ( $cost \approx 6 \cdot 2^{n_v}$ ). As expected, this rate of increase is exponential. The cost can be estimated for problems of higher dimensions.

The empty third column of the table shows that EGO could not sample points near the global solution because the algorithm expected no improvement in doing so. The values of  $error$  confirm this fact, the distance from the best-sampled point

( the sample having the minimum value of  $f$ ) to the global solution is higher than 1.5% for all the cases.

In the two-dimensional case, the algorithms were pretty accurate in the neighbourhood of the global optimum; this is justified by the low value of  $error$ . The minimum found by the Kriging model is very close to the global one.

For the three-dimensional case, the algorithm was misled since the best-sampled point was not the closest to the global solution. Consequently, the solution found by the meta-model is far from the global one  $error = 11.5\%$ .

For other cases, the solution found using the Kriging meta-model was not of good quality, as it was far from it by at least 3% of the design space range.

This example shows two drawbacks of EGO; the global solution is not determined reliably neither by :

1. sampling near enough to the global solution or,
2. having a good approximation near the global optimum

To investigate the causes of these drawbacks, we investigated the evolution of the algorithm iteration-by-iteration. In Figure 2.18, at the left, we see the evolution of the maximum value of the expected improvement (max EI) found by the algorithm, and on the right, the convergence rate (CR) of the algorithm solving the ISC at each iteration. The erratic behaviour in both plots is related to the change in the correlation parameters of the meta-model. But, interestingly, after the tenth iteration, the maximum value of the expected improvement decreases and, at the same time, the rate of convergence.

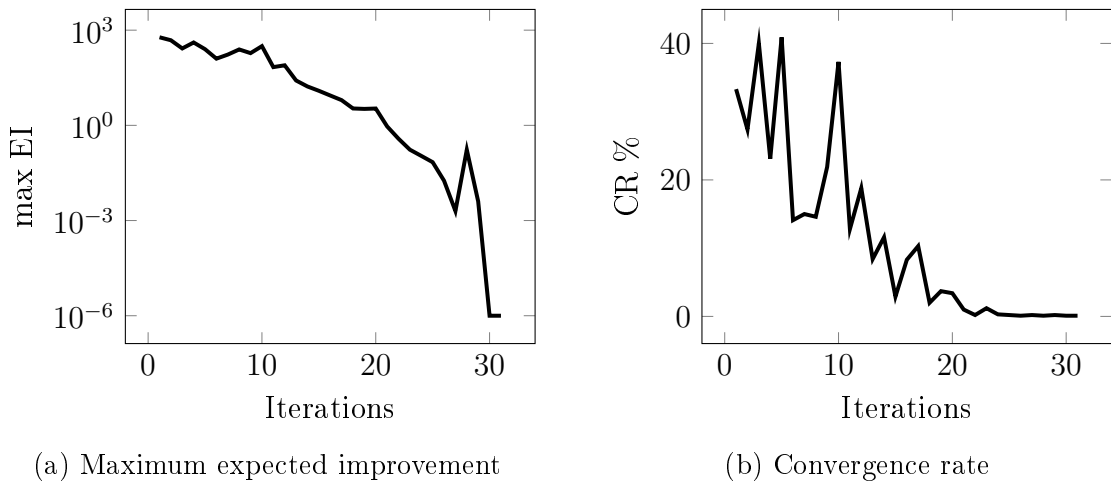


Figure 2.18: Evolution of the maximum expected improvement and the convergence rate of the algorithm

The decreasing trend of the convergence rate highlights the increased complexity of the ISC problem as the iterations progress. Indeed, we have seen for the one-dimensional problem that the expected improvement criterion has zeros-gradient in

the majority of the design space, and this explains why it is becoming cumbersome to solve the ISC as iterations continue.

Putting the chosen optimization algorithm to blame is legitimate, but we have considered one of the most efficient algorithms for local search and enhanced it with the multi-start strategy. In our opinion, no other algorithm could outperform the one we propose for solving these problems except an exact algorithm (which, by the way, explains why a branch and bound algorithm was used in the original paper for the optimization of the ISC [43]). Nevertheless, the estimation of the bounds to be used by an exact algorithm becomes computationally expensive as the number of samples increases; thus, the process becomes intractable and impractical.

All that being noted, we suggest that the EGO is not able to sample points near the global optimum because the algorithm was not able to find the global solution of the ISC.

## B Example 2

In this example, two Kriging models are fit one for the objective function and the other for the constraint function. The result of the optimization is presented in Table 2.4. The algorithm failed severely to solve the problem or even to locate where the global solution is. As shown in Table 2.4, one can see that the best-sampled point is very far from the global solution ( $error = 31.6\%$ ), and even the solution found by the meta-model is also faraway ( $error = 31.6\%$ ).

<i>cost</i>	<i>cost</i> <sub>1%</sub>	<i>error</i> %	<i>error</i> %
19	-	31.6	31.6

Table 2.4: Performances of EGO on the constrained problem 2.17

In fact, the algorithm found a local solution and got stuck with it. It kept sampling points near to each other in one of the feasible regions found. It is worth noting that the inaccuracy of the constraint meta-model caused this behaviour. Admittedly, the constraint function is complicated; thus, the meta-model was not capable of approximating it accurately.

Some research has been done on how to locate all the feasible regions before starting optimization. We proposed to use the probability of feasibility of  $PF$  criterion [41]. Maximizing  $PF$  will lead to solutions where the constraints are more likely to be satisfied.

We also proposed another criterion to look for multiple feasible solutions. This ISC allows sampling points that have the highest probability of feasibility while being distant from each other. This enables identifying multiple feasible zones (if any), and the optimization using another ISC for the optimum search can be conducted afterwards.

However, our experiences have shown that the optimization algorithm, maximizing  $PF$ , has the same problems as observed for the expected improvement ISC. Indeed, the  $PF$  function has a zero-gradient in large part of the design space; this renders the optimization problem harder to solve.

### 2.1.8 Discussion

We have noted some of the difficulties that need to be dealt with when optimizing using Kriging models. Here we will try to summarize these and cite some others that were highlighted in the literature :

1. The ISC optimization problems are highly multi-modal, and their modality is highly correlated to the size of the sample of the black-box model; when the number of samples rises, the number of local optima increases too. Furthermore, some ISCs can express zero-gradient in large part of the design space. Therefore, it becomes highly difficult to solve these optimization problems [38].
2. The infill criteria that focus more on the local search tend to sample points close to each other, which decay the conditioning of the correlation matrix and, in consequence, the time needed for the construction of the meta-model [79][80].
3. The number of samples exponentially increases as the dimensionality of the optimization problem does (the curse of dimension). Thus, the size of the correlation matrix increases and consequently the time needed to fit the meta-model. This time can even exceed the evaluation time of the black-box model.
4. When dealing with constrained optimization, the infill criteria tend to sample inside the feasible region but not on the constraint boundary which affects the solution accuracy [74].

#### A Solving the ISC

We have seen in Figure 2.18 that solving the ISC becomes challenging as the iterations progress. Even at the beginning of the process, the convergence rate did not exceed 40% and decreased as the algorithm iterates. Solving the ISC is vital for the success of the process of finding a global solution. Thus, the choice of the ISC is crucial.

#### B Correlation matrix conditioning

As the correlation matrix is inverted many times in the process of fitting the Kriging model, its conditioning is an indicator of how difficult model fitting will be. The condition number is the quantity that measures the conditioning. A matrix with

a low condition number is said to be well-conditioned, while a matrix with a high condition number is said to be ill-conditioned. In Figure 2.19, we see the evolution of the condition number as a function of the iterations for the examples treated.

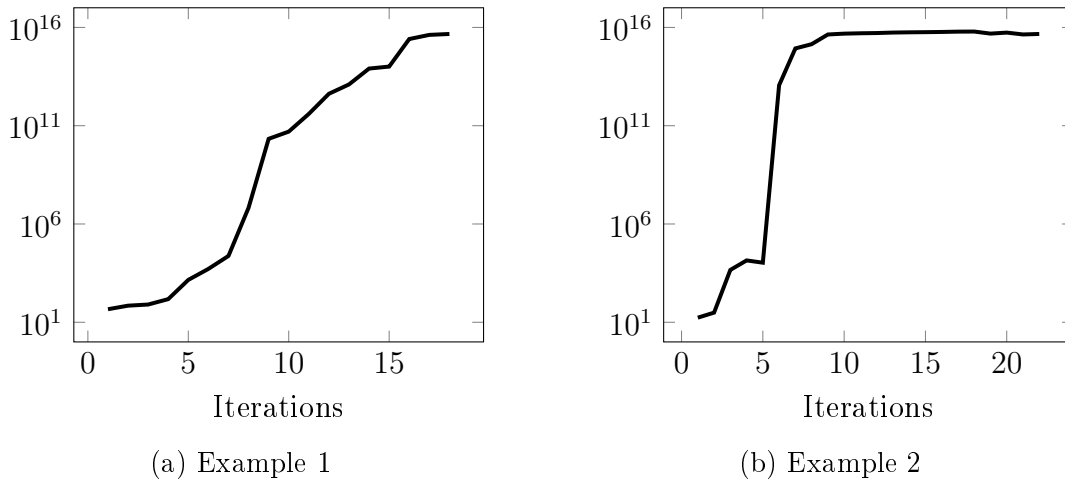


Figure 2.19: Evolution of the condition number as function of iteration

Clearly, the condition number of the correlation matrix increases as iterations increase. The correlation matrix somewhat expresses the distances between the samples. A large condition number exposes that there are samples that are far away from each other, while others that are very close to each other. This shows that the algorithm tends to sample point close to other existing samples as iterations go on.

In Example 2, the condition number is very high from the sixth iteration, as shown in Figure 2.19.b. We have noted that the optimization algorithm was trapped in a local minimum, and the algorithm kept sampling near it, which explains the large value of the condition number.

For both cases, the condition number increases, which means the the construction of the meta-model becomes since the correlation matrix is not guaranteed to be positive-definite due to rounding errors.

### C Fitting cost

Aside from the conditioning, the number of samples also impact the time needed for fitting the Kriging model. To illustrate this, we took the problem of Example 1 and tested it for different sample sizes and different number of variables ( $n_v$ ). To eliminate the impact of the conditioning, the samples are drawn using the Hammer-sley sequence to get a space-filling design. Figure 2.20 shows the evolution of fitting time as a function of the sample size. The figure reports the mean values of multiple runs.

We can see that the time needed for model fitting increases as the sample size grows. The dimensionality seems not to have a significant impact on fitting time. This can be explained by the fact that the inversion correlation matrix is the most



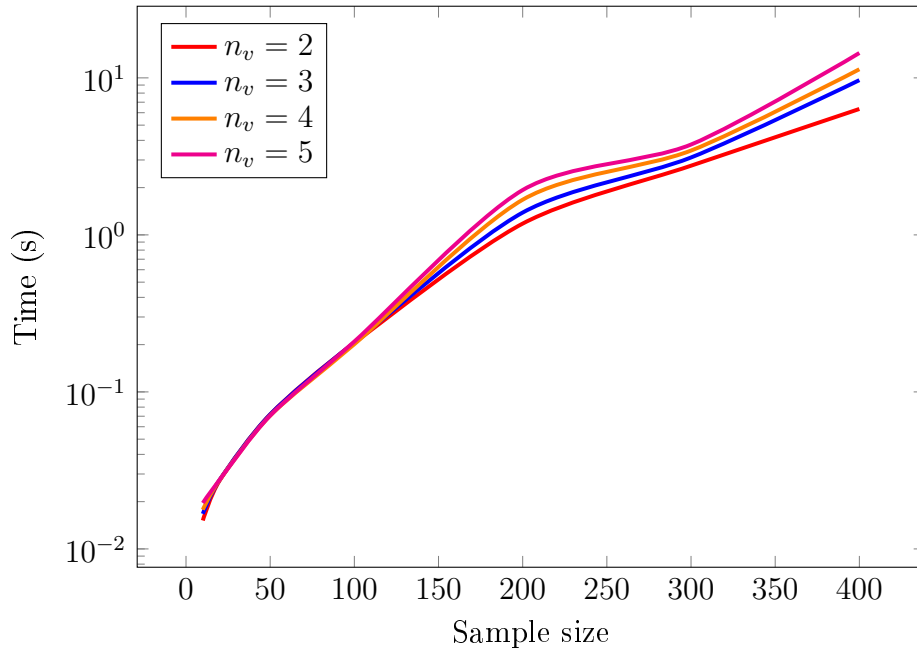


Figure 2.20: Cost of fitting the Kriging model for Example 1

computationally expensive operation. Moreover, its size depends on the sample size but not on the dimensionality of the problem.

Nevertheless, a problem of higher dimension needs more samples to approximate sufficiently the black-box model (the curse of dimension). Therefore, the increased cost related to dimensionality.

## D Synthesis

The conventional way of using the Kriging model seems to be not reliable since the meta-model fitting is a multifaceted process. Indeed, the conditioning of the correlation matrix and size of the samples profoundly impacts the time needed for model fitting and the quality of the fitted model. Additionally, traditional ISC problems increase in complexity as the sample size does.

To solve the problems mentioned above, "simple" meta-models need to be fitted. "Simple" stands for meta-models that are easier to fit and lead to ISC problems that are tractable using classical optimization algorithms. Many optimization algorithms discussed in Chapter 1 solve optimization problems using many relaxed forms of the initial problem. This approach appears promising in the context of approximation-based algorithms and still an unexplored research path.

A divide and conquer strategy appears to be a good starting point; the initial design space is divided into sub-spaces with models easier to fit and problems faster to solve. Then the sub-spaces are compared to further exploration for the most promising ones. The idea is based on the branch and bound (B&B) algorithm.

## 2.2 Branch and Bound assisted by Meta-models (B2M2)

To recall, B&B algorithms consist of a systematic enumeration of candidate solutions by splitting the search space into smaller spaces. The algorithm explores each smaller space then estimates the upper and lower bounds of the objective function on each one of them. A space is pruned if it cannot produce a better solution than the best one found so far by the algorithm. B&B were mainly developed for discrete optimization by other extensions to continuous optimization were proposed. Messine et al. developed an interval B&B algorithm (IBBA) [29] [31]. IBBA computes the bounds of the objective function using interval arithmetic. Its performances depend on the efficient estimation of the lower and upper bounds on the sub-regions. The lower and upper bounds are critical elements for IBBA, But, in the case of an expensive black-box model, the real bounds are, in most cases, difficult or even impossible to compute. Thus, it can be applied to only a limited number of situations with explicit objective function and constraints.

Nevertheless, Kriging enables us to model the uncertainty on a black-box function; This uncertainty is modelled by a random variable of a normal distribution of mean  $\hat{f}(p)$  and variance  $\sigma^2(p)$  often called Gaussian process. Intuitively, this means that we are saying that the function  $f(p)$  has a typical value of  $\mu(p)$  and is expected to vary in some range like  $[\hat{f}(p) - 3\sigma(p), \hat{f}(p) + 3\sigma(p)]$ .

This fundamental property of the Kriging model will enable estimating the bounds to be used by the B&B algorithm.

### 2.2.1 Proposed algorithm

We propose an algorithm that builds many meta-models each on a specified region of the design space; each one of them is relatively easy to fit and easy to use to estimate the upper and lower bounds. Then, iteratively, prune the regions that are not promising. By adopting this algorithm, the drawbacks mentioned earlier are tackled, namely the numerical ones related to the correlation matrix. The flowchart of the algorithm is shown in Figure 2.21.

### 2.2.2 Initialization

In the beginning, the initial space  $S$  is defined, and an empty list  $L$  is set,  $L$  will store the potentially optimal regions. Additionally, a maximum depth level  $l$  is defined to limit the exploration of the algorithm. The depth level can be seen as a metric of how many times the initial design space  $S$  can be subdivided.

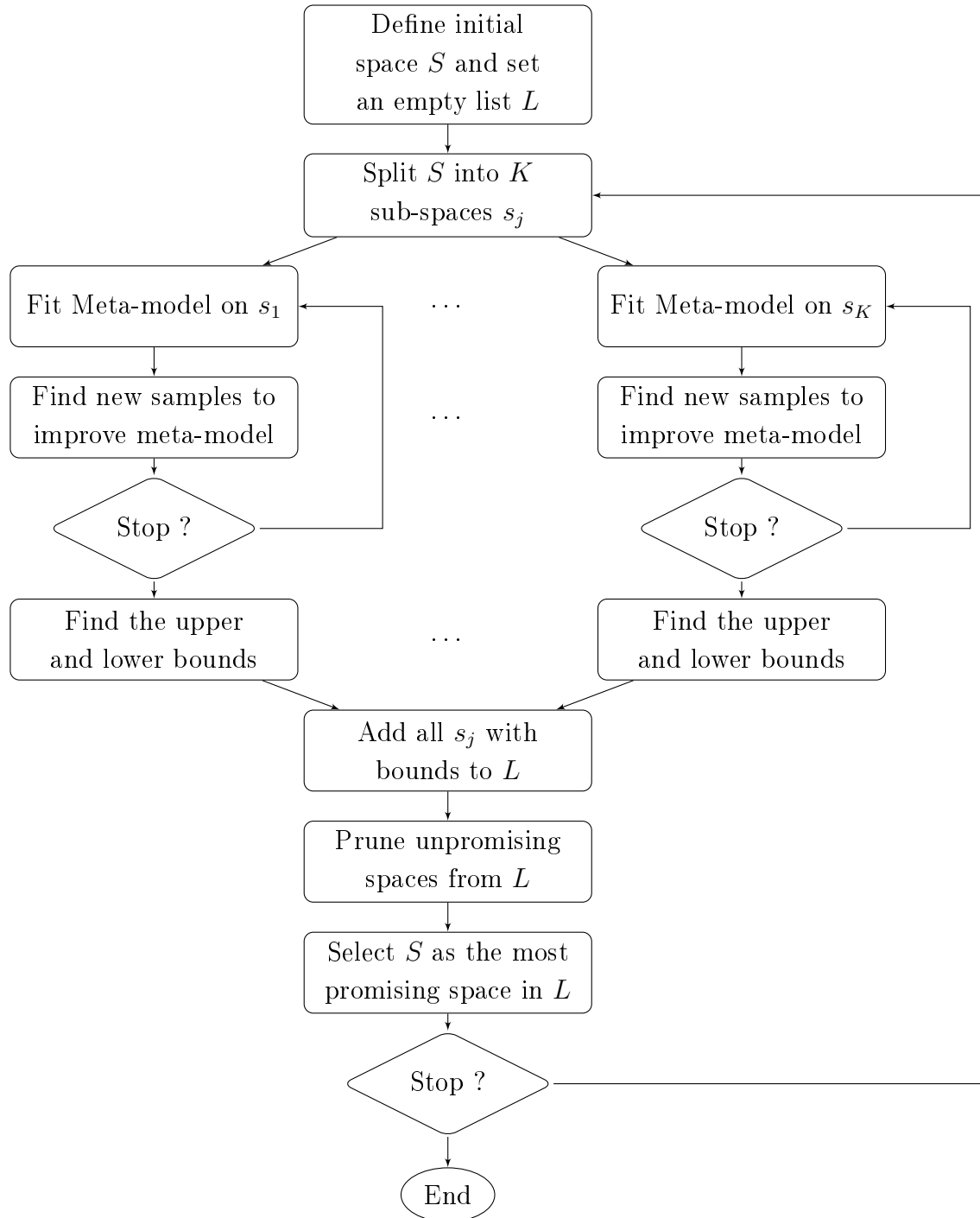


Figure 2.21: Flowchart of B&B Kriging

### 2.2.3 Branching

Branching stands for subdividing the initial design space  $S$ . The most intuitive rule is splitting each dimension of  $S$  by two. This will lead to  $K = 2^{n_v}$  sub-spaces  $s_j$ . Unfortunately, the number of sub-spaces increases exponentially as the number of variables grows. This is not desired, and a more practical rule is proposed.

For meta-models, the purpose behind branching is to improve upon its accuracy by reducing the size of the design space. Thus, the space  $S$  is split, such as the produced sub-spaces are of better accuracy. Therefore, the meta-model on  $S$  is used to identify the dimensions on which high inaccuracy is observed.

We recall that the stochastic term in the Kriging predictor enables us to model the error not captured by the regression term. This term is characterized by the correlation function and thus its parameters  $\alpha$  and  $\theta_1, \dots, \theta_{n_v}$ . The value of  $\alpha$  characterizes the smoothness of the function while values of  $\theta$  enable to model the changes over distances. Large values serve to model rapid changes even over small distances. Hence, the value of  $\theta$  is an indicator of how the function varies.

If the value of  $\theta_i$ , corresponding to a variable  $p_i$ , is the largest. Then, the change in the  $i$ -dimension is more important than other dimensions. Consequently, splitting the space  $S$  by dividing its dimension  $i$  will lead to meta-models on the sub-spaces that are better than the one on  $S$ .

A simple example is used to test this property of the Kriging model, the studied function is a three-dimensional function but that depends on only two of them, and one of the variables present more variation than the other 2.43. From this example, we expect that the value of  $\theta_2$  to be higher than  $\theta_1$  and  $\theta_3$ . So, a Kriging model is fit to a DoE of 15 samples and using a linear regression term.

$$f(p_1, p_2, p_3) = p_1 p_2^2, \text{ with } 0 \leq p_i \leq 10, \quad i = 1, 2, 3 \quad (2.43)$$

Indeed, the meta-model fitting led to  $\theta = (0.072, 0.171, 0.004)$  which confirms the idea we proposed. Ordering the value of  $\theta$  will lead to the variables that have the highest variation. The function does not depend on  $p_3$  then  $\theta_3 = 0.004$  which is the smallest allowed value for  $\theta$ .

More than one variable can be chosen for the branching; these variables are picked in the decreasing order of their corresponding  $\theta$  values. Then each variable is subdivided in two part which will lead to  $2^d$  sub-space where  $d$  is the number of variables picked.

### 2.2.4 Meta-modelling

Once the sub-spaces are determined using the branching rule, Kriging models are fit on each one of them. The initial DoE, such as Hammersley, is determined, and

the black-box-model is evaluated.

Additional samples are added using infill criteria to improve the meta-model approximations and to deal with the monotonic CLLHF issue. The choice here is crucial for the estimation of the bounds; we rely on the prediction error  $\hat{\sigma}$ , new samples are added where the prediction error is high; consequently, we improve the meta-model globally on the sub-space. Additionally, a limited number of points are sampled using expected improvement (EI) criterion to enhance local exploitation, in case the CLLF is monotonic. This will also improve the global performances of the algorithm.

The initial DoE size and number of the points added using  $\hat{\sigma}$  are specified by the designer. In contrast, the number of points added using EI is conditioned by the monotonicity of CLLF. The algorithm stops if the CLLH function is not monotonic.

### 2.2.5 Bounding

Once the meta-model is fit on each of the sub-spaces, we look for the lower and upper bounds on each of them. As Kriging is capable of giving the predictor and its error estimation, we exploit this information for the computation of bounds using the following formulation (2.44) and (2.45):

$$\begin{aligned} \underline{f} &= \min_p \quad \hat{f}(p) - k\hat{\sigma}_f(p) \\ \text{s.t.} \quad & \hat{g}_i(p) - k\hat{\sigma}_{g_i}(p) \leq 0, \quad i = 1, \dots, n_c. \end{aligned} \quad (2.44)$$

$$\begin{aligned} \bar{f} &= \min_p \quad \hat{f}(p) + k\hat{\sigma}_f(p) \\ \text{s.t.} \quad & \hat{g}_i(p) + k\hat{\sigma}_{g_i}(p) \leq 0, \quad i = 1, \dots, n_c. \end{aligned} \quad (2.45)$$

where  $\hat{f}$  and  $\hat{g}_i$  are the predictors of the objective function  $f$  and the constraints  $g_i$  respectively and  $\hat{\sigma}_f$ ,  $\hat{\sigma}_{g_i}$  are their error estimations given by the Kriging meta-model. The parameter  $k$  can be defined as the confidence level required, *e.g.*  $k = 3$ , it corresponds to a confidence level assuming that the prediction distribution is normal.

The optimization problems (2.44) and (2.45) has to be solved using appropriate optimization algorithms due to the multi-modal behaviour introduced by the  $k\hat{\sigma}$  terms. We used sequential quadratic programming from Matlab for the task and improved by a multi-start strategy with points uniformly spread on the design space, the number of starting points is much higher than the number of samples of the black-box model. Additionally, the gradient information from the meta-model is exploited to increase the efficiency of the algorithm.

The optimization problem (2.45) can present difficulties in solving it since the objective function and constraints are mostly concave. Furthermore, our experience

has shown that  $\bar{f}$  is, most of the time, equal to the best-sampled point. Thus, this optimization can be discarded, and  $\bar{f}$  is set to be equal to the sample having the smallest value of the objective function and satisfies the constraints on the sub-space. For highly constrained problems, when no feasible sample is available,  $\bar{f}$  is set to be equal a large value.

In most constrained optimization problems, the problem (2.44) may not have a solution on some sub-spaces. It means that these sub-spaces have no potentially feasible solution, and therefore the values of the bounds are set to some high value.

### 2.2.6 Elimination

The sub-spaces are added to  $L$  with their lower  $\underline{f}$  and upper  $\bar{f}$  bounds. A comparison between the newly added sub-spaces to  $L$  and the existing ones is conducted to eliminate the ones that cannot produce a better solution.

First, we set  $\bar{b}$  as the smallest upper bound of all the spaces in  $L$ , and then, we prune the spaces from  $L$  that they have a lower bound that is bigger than  $\bar{b}$ .

### 2.2.7 Selection

The remaining sub-spaces in  $L$  are all promising ones; thus, one of them has to be chosen for the next iteration to be subdivided. Many strategies could be considered; by preferring depth-first for prioritizing local search or breadth-first for prioritizing exploration of the search space. In our work, we use the best-first; it is based on the lower bound of the objective function on the sub-spaces. Indeed, the sub-space chosen from  $L$  has the smallest lower bound, and its depth level is smaller than the maximum specified depth level.

This strategy will prioritize sub-spaces that have the maximum potential for providing the best solution without any criteria on the size of the sub-space.

The sub-spaces that have attained the maximum allowed depth level are stored in a list  $L_s$ .

### 2.2.8 Termination

The algorithm will naturally iterate until the list  $L$  is empty. It is challenging to come up with a robust termination criterion, but a heuristic can be used to stop the exploration of a sub-space if the difference between its lower bound and upper bound is less than a small threshold specified by the designer. It remains possible to stop the algorithm by setting a maximum number of evaluations of the black-box model.

The list  $L_s$  contains the remaining sub-spaces that are at the maximum specified depth level, or their bounds are tight. One should investigate them, and search for the optimum in one of them without considering the prediction error. Otherwise, if no

solution is apparent, the algorithm could be restarted with the list  $L_s$  and a higher maximum depth level.

### 2.2.9 Analytic Examples

The problems (2.41, 2.42) treated using the conventional approach are solved using the new algorithm. The setup of the algorithm parameters is as follows

- The maximum depth level is two times the number of variables.
- The parameter  $k$  in (2.44) and (2.45) is taken to be equal to three
- On each sub-space, the initial sample is chosen using Hammersley sequence with two times the number of variables, the same number of samples is added using the prediction error as an ISC and two points are added using EI.

Figure 2.22 shows the evolution of the sub-space division applied to the problem treated in Example 1 ( $n_v = 2$ ). As shown, on each sub-space, we compute the lower and upper bounds, *i.e.*  $\underline{f}$  and  $\bar{f}$ , respectively, the new sub-spaces are compared with the ones in the list  $L$  to discard unpromising ones. The shaded area shows the sub-spaces discarded from the list  $L$ .

After only four iterations, the algorithm located the sub-space where the optimum lies. The green region is the only space that was left in the list  $L$ . When searching in this space without considering the prediction error, the solution found is close to the the global minimum of the problem.

The results for the remaining problems are summarized in table 2.5. Two of the metrics proposed before are not relevant here ( $cost_{1\%}$  and  $error$ ) since the algorithm itself does not aim to sample at the optimum but to have a meta-model of good quality around the optimum.

	$n_v$	$cost$	$error$ %
Example 1	2	115	0.05
	3	515	0.007
	4	2015	0.5
Example 2	2	31	0.001

Table 2.5: Performances of B2M2 on the analytic examples

The algorithm performed very well in locating the optimums for different problems; the relative error between the global solution and the solution found using the meta-model is less than 1% for all the cases.

For Example 2, where the problem is constrained, The algorithm performed only one iteration because the difference between the lower and upper bounds is

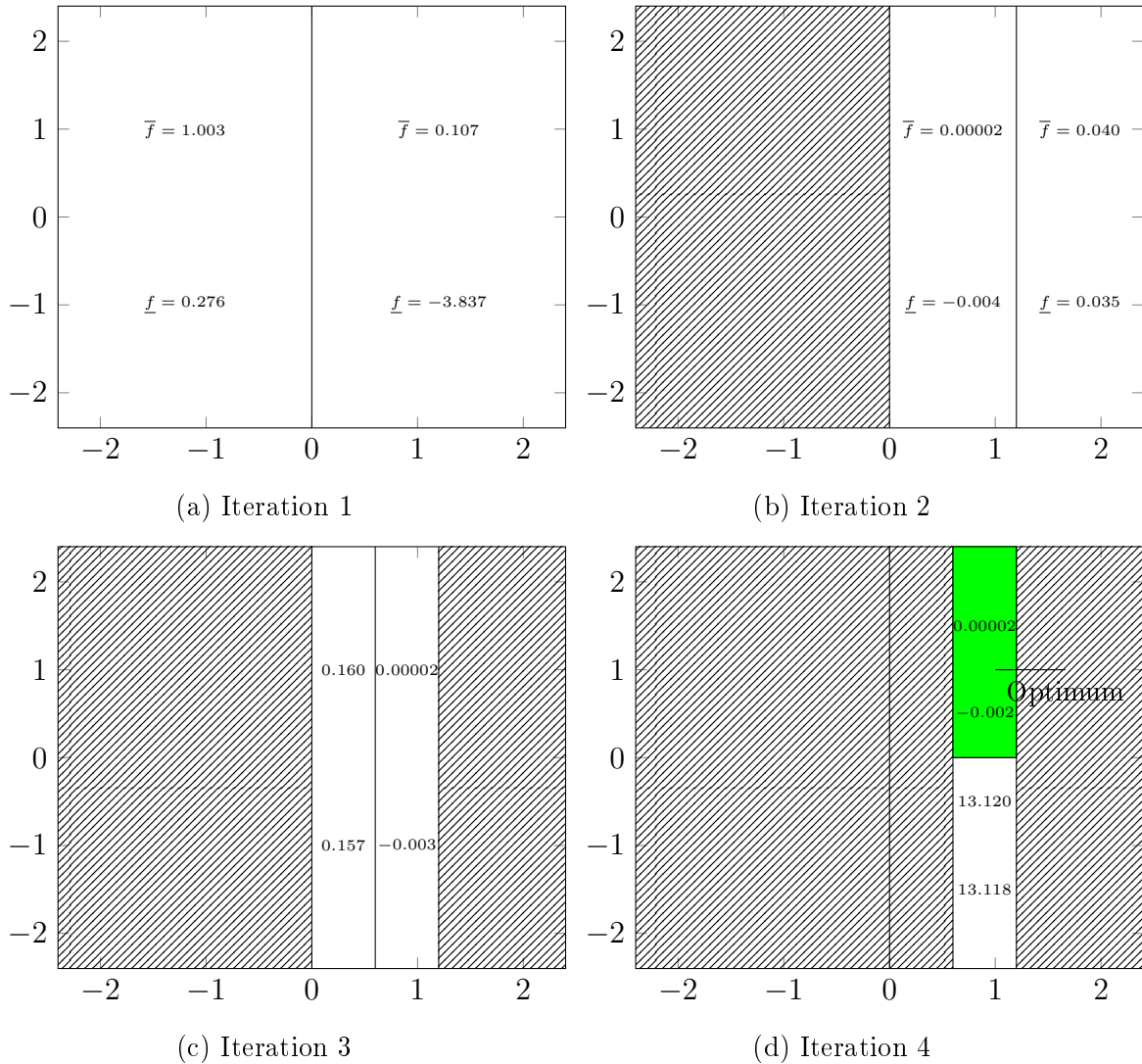


Figure 2.22: Space exploration by the algorithm

smaller than a small value. It located all feasible regions and was able to discard the unpromising region immediately. Consequently, we expect our algorithm to perform well for constrained optimization problems.

### 2.2.10 Extension to multi-objective optimization

The algorithm can treat multi-objective optimization in many ways; an  $\varepsilon$ -constraint method transforms the multi-objective problem to multiple mono-objective ones. Then, the algorithm could be used while storing the search tree in case one of the sub-spaces will be used by one of the remaining mono-objective problems.

Another strategy could be based on the concept of dominance. The bounds of each sub-space are now Pareto fronts, and a sub-space is eliminated if it cannot produce a better Pareto front than the actual best one. Nevertheless, some efficient algorithm has to be used for solving multi-objective problems to find the bounds.



### 2.2.11 Extension to optimization under uncertainty

Uncertainty could be dealt with using the same algorithm by changing the bounding problems, to make them correspond to performance needed.

For example, A worst case optimization could be integrated by using a formulation from [49] (Section 1.1.1). The problems in (2.44) and (2.45) are transformed to

$$\begin{aligned} \underline{f} = \min_p \quad & \hat{f}(p_{w,0}) - k\hat{\sigma}_f(p_{w,0}) \\ \text{s.t.} \quad & \hat{g}_i(p_{w,i}) - k\hat{\sigma}_{g_i}(p_{w,i}) \leq 0, \quad i = 1, \dots, m. \end{aligned} \quad (2.46)$$

$$\begin{aligned} \bar{f} = \min_p \quad & \hat{f}(p_w) + k\hat{\sigma}_f(p_w) \\ \text{s.t.} \quad & \hat{g}_i(p_w) + k\hat{\sigma}_{g_i}(p_w) \leq 0, \quad i = 1, \dots, m. \end{aligned} \quad (2.47)$$

$p_{w,0}$  and  $p_{w,i}$  for both problems are computed as stated in [49].

Interestingly, we have similar formulations as for the deterministic case, which enables us to treat different problems with the same algorithm and even to test multiple levels of uncertainty with little computational overhead since the explored sub-spaces could be reused.

## 2.3 Chapter Summary

In this chapter, we presented a thorough introduction to optimization using meta-models, mainly Kriging-related. We presented some of the challenges of fitting "good" meta-models and how to assess their quality. The likelihood functions can sometimes lead to meta-models of lousy quality; thus, we proposed to limit the variation of the correlation parameters in a specified interval to have a minimum and maximum allowed correlation between the samples.

Furthermore, we have highlighted the drawbacks of using the conventional approach that consists of fitting a single meta-model on the whole design space and enrich it sequentially using infill criteria. This approach becomes eventually unreliable since problems related to the conditioning of the correlation matrix, the complexity of fitting the meta-models and the constraints handling surge.

Thus, we proposed a novel strategy that consists of building many meta-models each on a specified region of the design space, each one of them is relatively easy to fit and easy to use for the purpose of optimization (less modality). Then, iteratively, prune the regions that are not promising. This process keeps a good trade-off

between exploration and exploitation of the design space; thus, it enables us to produce reliable solutions. Sometimes, this algorithm can be costly because of the B&B paradigm.

Nevertheless, the most expensive process is easily parallelizable, by either subdividing each sub-space to multiple smaller ones and explore them in a parallel manner or by taking multiple sub-spaces from the list and explore them likewise.



# Chapter 3

## Intrusive approach

As noted in the previous chapters, in this chapter, we will open the black-box to extract relevant information from the model for the optimization algorithms and uncertainty quantification methods. We have seen in the first chapter that optimization algorithms based on the derivatives are the most efficient ones when the gradient information is available. Furthermore, the gradient could also be used for uncertainty quantification using perturbation methods. This double usefulness of the gradient is of high interest. Nevertheless, getting this information is not straightforward in the case of finite element analysis.

To illustrate the methodology and to point out the main difficulties, we will treat a simple electromagnetic device shown in Figure 3.1. It consists of the model of an infinite solenoid powered by a coil of current density  $J = 0.01A/mm^2$ . The position and the width of the coil are respectively  $R = 0.7m$  and  $d = 0.3m$ . A Dirichlet boundary condition is imposed on the edge shown in black.

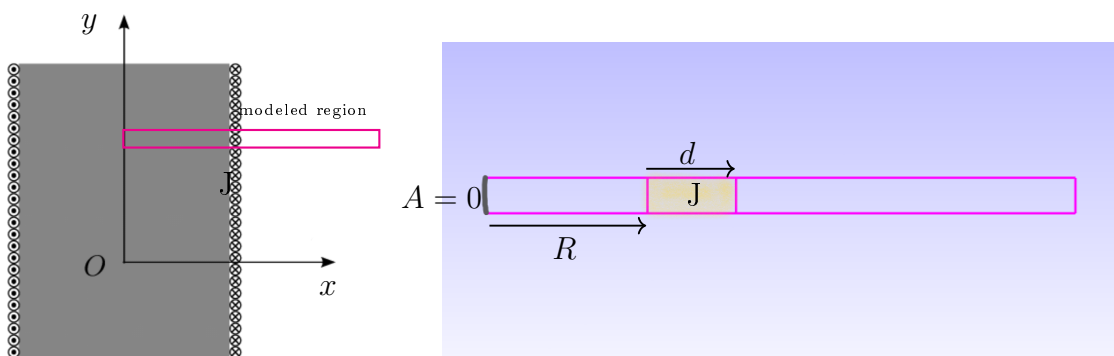


Figure 3.1: Infinite solenoid model

For this device, we aim to compute two quantities of interest and their derivatives with respect to the variables  $(R, d, J)$ :

1. the magnetic energy stored by the device :  $W$
2. the maximum magnetic flux density in the coil :  $B_c$

We chose to treat this problem thanks to its simple geometry, and also because of the explicit solutions of magnetic quantities can be easily calculated on the whole studied domain. This will enable us to validate the quantities computed using FEA and, most importantly, their derivatives.

The magnetic flux density inside the solenoid can be computed as follows.

$$\mathbf{B}(x) = \mu_0 J d, \quad x < R \quad (3.1)$$

In the coil, we have a linear decrease (in the  $x$ -direction) of the magnetic flux density,  $\mathbf{B}$  is written as

$$\mathbf{B}(x) = \mu_0 J (R + d - x), \quad R \leq x < R + d \quad (3.2)$$

Outside of the solenoid, the magnetic flux density is zero.

For computing  $B_c$ , we take a small distance inside the coil at position  $x = R + e$  ( $e = 10^{-3}$ ), then,  $B_c(R, d, J) = \mu_0 J (d + e)$ .

The magnetic energy per unit length  $W$  is computed using the following formula

$$W = \frac{1}{2} \int \frac{1}{\mu_0} \mathbf{B}^2 dx \quad (3.3)$$

Therefore, the magnetic energy is reduced to the following form

$$W(R, d, J) = \frac{1}{2} \mu_0 (Jd)^2 \left( R + \frac{1}{3}d \right) \quad (3.4)$$

Now, we have a closed-form formula for the quantities of interest  $B_c$  and  $W$ . On the other hand, an FEA of the device is conducted, and the discrete solution is shown in Figure 3.2. First order shape functions have been chosen to approximate the vector potential  $A$ . The exact solution of the problem is of the second order since the magnetic flux density  $\mathbf{B} = \text{curl}A$  is of the first order (see 3.2). Consequently, the Finite Element model can not fit perfectly the exact solution and it will lead to numerical errors.

A comparison of the quantities computed with the analytic and FEA is conducted, and relative errors are then shown in Table 3.1. We notice that we have fewer errors for energy than for  $B_c$ . Generally speaking, FEA leads to fewer errors for global quantities such as energy and higher errors for local quantities.

	Analytic	FEA	Rel. error %
$B_c$	3.757 mT	3.592 mT	4.39
$W$	4.5239 J/m	4.5225 J/m	0.03

Table 3.1: Comparison of analytic and FEA quantities

So now, we are interested in computing the gradient of the quantities of interest with respect to the variables  $R$ ,  $d$  and  $J$ . The next section shows some approaches that are often considered for conducting this task.

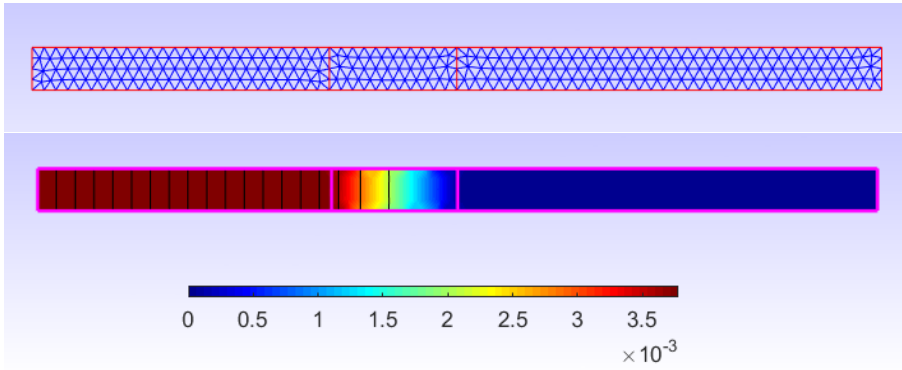


Figure 3.2: The mesh of the device (Top), and the distribution of magnetic flux density in  $T$  (Bottom)

### 3.1 How to compute the gradient?

As noted in Chapter 1 of this dissertation, one could compute the gradient using finite-difference by imposing a small perturbation on the variables.

#### 3.1.1 finite-difference

The finite-difference method enables us to approximate the gradient by imposing a small perturbation on the considered variable. For example, the derivative of the energy with respect to the variable  $R$  can be approximated as follows

$$\frac{\partial W}{\partial R} \approx \frac{W(R + \varepsilon, d, J) - W(R, d, J)}{\varepsilon} \quad (3.5)$$

where  $\varepsilon$  is a small value, this scheme implies running an additional FEA with the new value  $R + \varepsilon$ , generally speaking, we need as many simulations as there are variables. In our example, since we have three variables ( $R, d, J$ ), we will need three additional FEA for computing the gradient.

The formula in (3.5) is called forward difference; one could also use backward difference or centred difference. The latter requires twice as much number of FEA as the number of variables, but it can be more accurate than forward and backward difference.

The choice of  $\varepsilon$  can be delicate, and some tuning is always necessary. In the literature, the estimation of the optimal values of  $\varepsilon$  is quite studied [81] [82]. Some prior knowledge of the model, such as an estimation of the second derivative, is often necessary for estimating a good value for  $\varepsilon$ .

Figure 3.3 shows the relative error of the derivative computed with the finite-difference compared to the exact derivative while varying  $\varepsilon$  from  $10^{-18}$  to  $10^{-2}$ . We can see clearly that the derivatives depend on the value of  $\varepsilon$ . Small values lead to more errors because of rounding errors. Moreover, high values can lead to some truncation errors, as seen for  $\frac{\partial W}{\partial J}$  and  $\frac{\partial W}{\partial d}$ .

Rounding errors are caused by the accumulation of machine precision errors when using small values of  $\varepsilon$ , while truncation errors are related to the approximation used in 3.5; truncation errors are related to the Taylor expansion (only first order term is used for the approximation) and are proportional to  $\varepsilon$ , which explains why the error for  $\frac{\partial W}{\partial J}$  and  $\frac{\partial W}{\partial d}$  increases when  $\varepsilon$  is big.

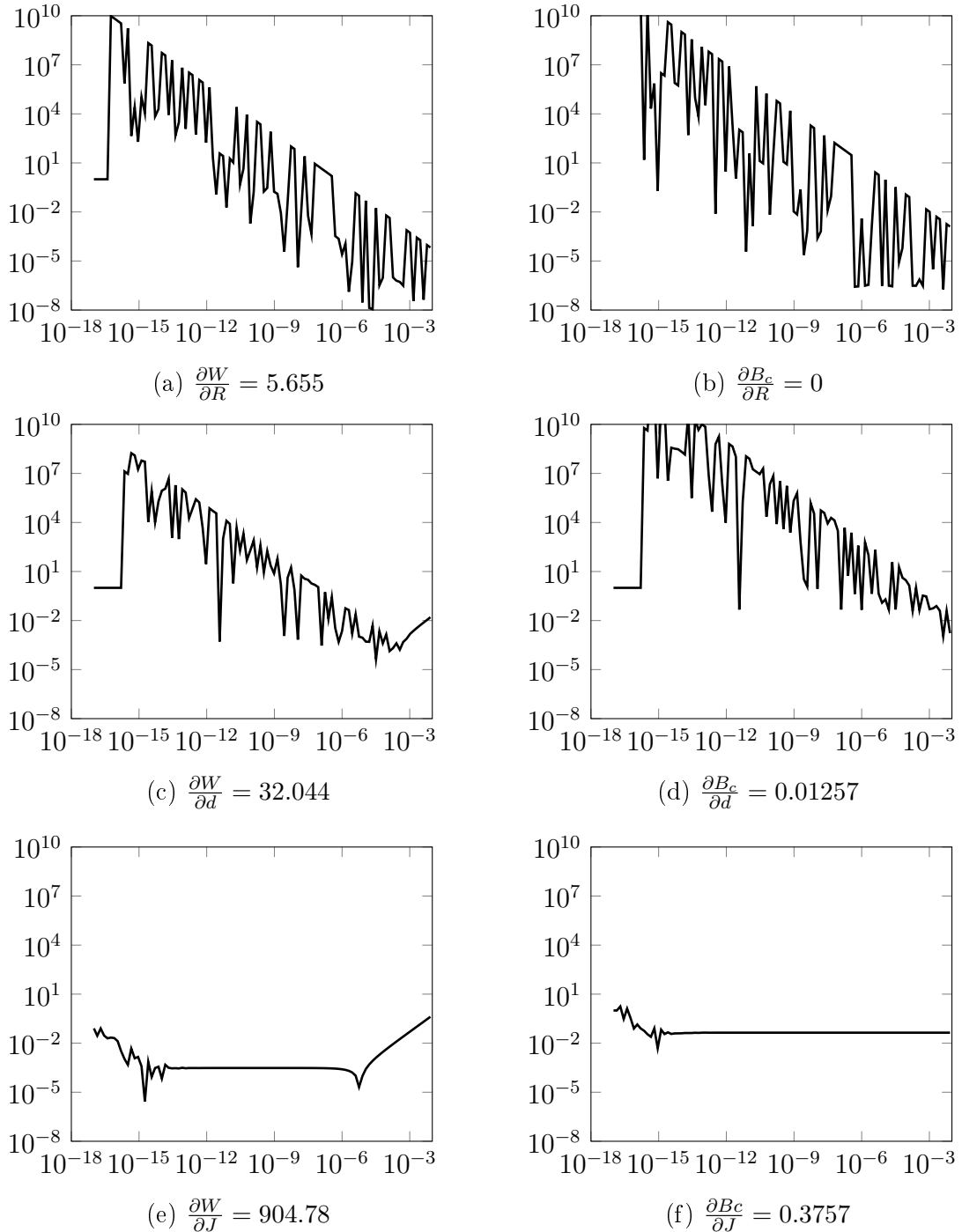


Figure 3.3: Relative error of derivative computed by finite-difference

We can see that high values of  $\varepsilon$  ( $> 10^{-5}$ ) have led to reasonably acceptable results for all the derivatives in our simple example. Nevertheless, it remains risky

to use such a scheme since the optimal value of  $\varepsilon$  depends on the quantity computed (global or local) and the variables for which the derivatives are calculated.

Another behaviour that is seen in Figure 3.3 is the erratic variation when dealing with geometric variables  $R$  and  $d$ ; the chaotic discontinuities are related to the change of the mesh when varying  $\varepsilon$ . In fact, when we apply the finite-difference scheme, we have to calculate two values of the quantity of interest for two geometries (for example, geometry 1 with  $R$  and geometry 2 with  $R + \epsilon$ ). The brute approach to obtaining the two FE solutions involves re-meshing the two geometries and then solving the FE problem. This approach often leads to a "discontinuity" between the two FE solutions, which are not then approximated in the same space. In Figure 3.4, we see that the same geometry could be meshed differently depending meshing algorithm configuration. Even small variations in the geometry can lead to a change in the resulted mesh (re-meshing), which, as mentioned previously, impacts the FEA solution.

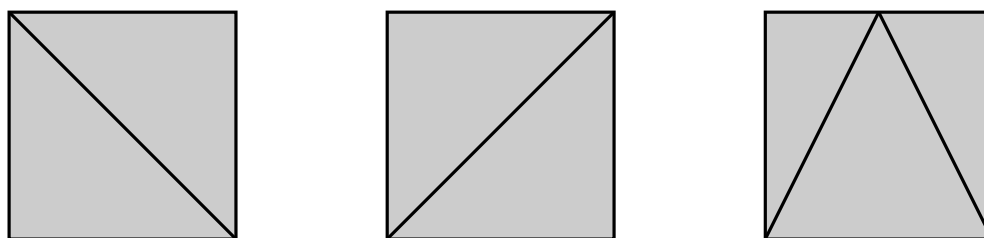


Figure 3.4: Same geometry, different meshes

To eliminate the impact of re-meshing, one would like to keep the same mesh topology while changing the geometry, as shown in Figure 3.5. This approach is commonly referred to as mesh morphing.

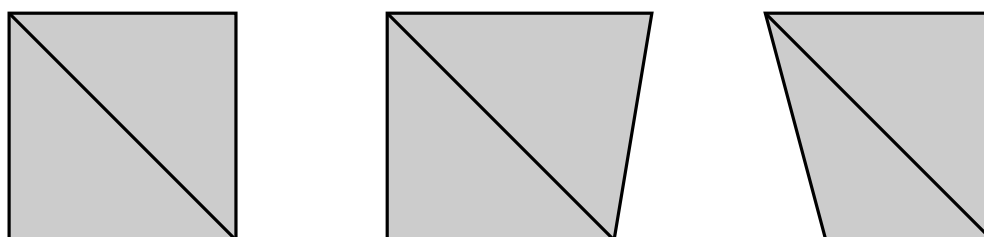


Figure 3.5: Same mesh topology, different geometries

### 3.1.2 Mesh-morphing

Different approaches for mesh morphing have been developed in order to deform an initial mesh to take into account shape modification without changing the mesh topology. The concept consists in imposing a displacement for a set of mesh nodes and determining the new coordinates for all other ones by an interpolation approach. In this context, the spring analogy [83] [84], Laplacian smoothing [85], linear elasticity [86] and the Radial Basis Function [87] [88] interpolation method are largely



treated in the literature. Explaining the ideas behind each method is out of the scope on this dissertation, we will limit ourselves to the application of the concept to our simple electromagnetic device.

Changing the geometry of an electromagnetic device by mesh morphing stands for moving the mesh nodes according to the change in the geometry. In our example when changing the parameter  $R$ , we proceed by moving all the mesh nodes on the boundary of the coil by  $\varepsilon$  in the  $x$ -direction while all the remaining nodes stand still.

In Figure 3.6, we show the resulted relative error in the derivative computation when using the mesh morphing. We can see that the erratic behaviour seen in Figure 3.3 has disappeared. Mesh morphing gives a piece of more reliable information about the derivatives when compared to the one given with re-meshing since its less sensitive to the value of  $\varepsilon$ . Still, it requires further attention to the choice of the best values of  $\varepsilon$ . Moreover, carrying out a mesh morphing is not necessarily evident for 3D geometries, especially in areas where the mesh is fine such as the airgap of an electrical machine.

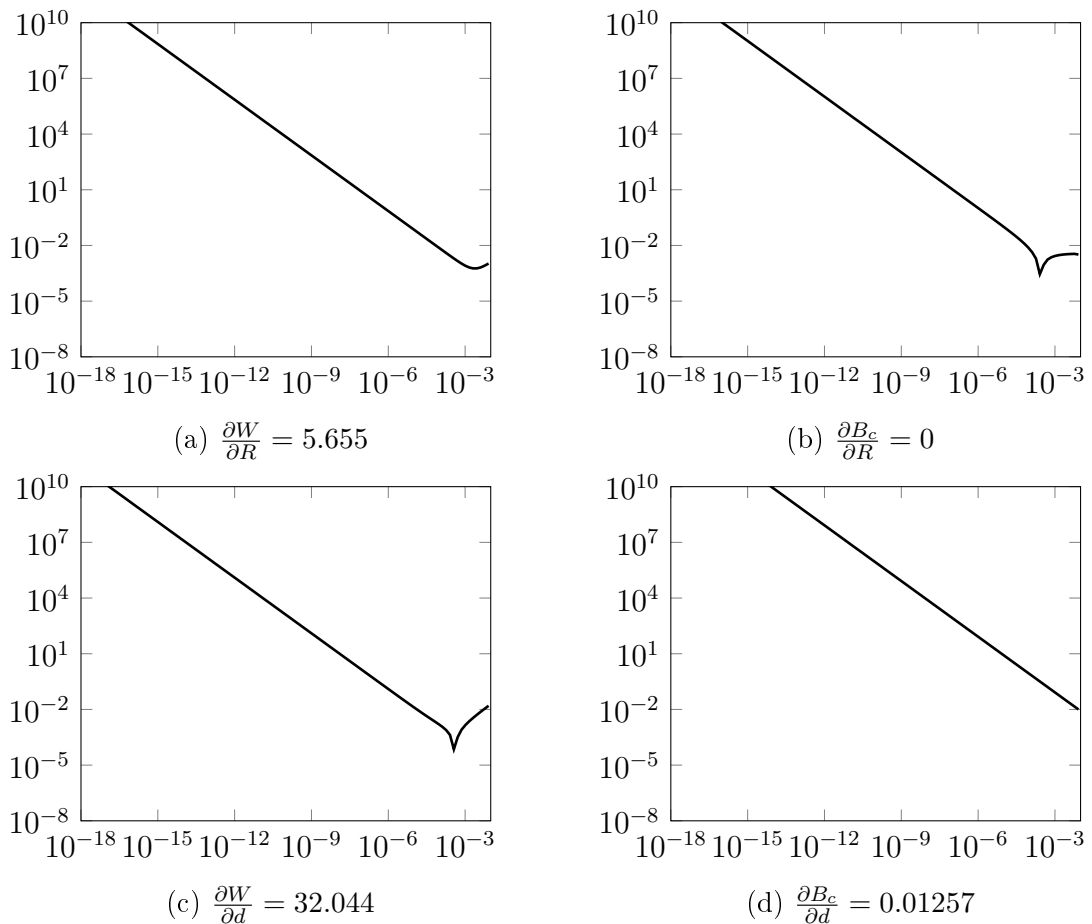


Figure 3.6: Relative error of derivative computed by finite-difference and mesh-morphing

### 3.1.3 Discussion

We have shown in this section how to compute the gradient of a quantity of interest using finite-difference method. The finite-difference has three shortcomings

1. The choice of the step size  $\varepsilon$  is not straightforward [89].
2. Re-meshing error can highly deteriorate the information about the derivatives and mesh-morphing is not straightforward for most of geometries.
3. The computational cost is proportional to the number of variables which can be disadvantageous when dealing with computer expensive simulation such as FEA.

To tackle all these issues, some researchers have proposed to use the adjoint variable method for computing the gradient from the mathematical model of the electromagnetic phenomena [90] [91] [92]. This approach requires an intrusive manipulation of the FE code to calculate the gradient efficiently and accurately.

The adjoint variable method is used in many fields that involve solving a large system of algebraic equations. The solution to such system is computationally expensive; thus, computing quantities of interest involving this system is also computationally expensive. The adjoint variable method can compute the gradient with a cost less or equal to the cost of solving the initial system. This last property makes the approach very attractive when dealing with many variables in the context of an FEA.

However, the implementation of such a method is not trivial and requires intrusive manipulation in the code of the simulation tool. For this reason, in this chapter, we will start by explaining the main ideas behind that adjoint variable method for a simple problem (an electrical circuit) before extending the concept to an FEA code.

## 3.2 Explanatory example

Let us consider a basic electrical circuit to explain the fundamental ideas of the adjoint variable method. Figure 3.7 shows an RL circuit composed of a resistor of resistance  $R$  and an inductor of reactance  $X$  driven by the sinewave voltage source  $\underline{V}_{in}$ .

For this circuit, we want to compute the Joule losses in the resistor. However, to be able to get to this end, the current flowing through the circuit has to be computed since the Joule losses are equal to the resistance times the RMS current squared. To show this, we will adopt the complex notation for the calculation.

The equation of the circuit is written as

$$(R + jX)\underline{I} = \underline{V}_{in} \quad (3.6)$$

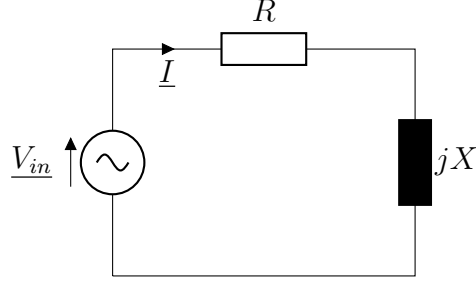


Figure 3.7: Electrical circuit

where  $j$  represent the imaginary unit ( $j^2 = -1$ ).

We suppose that the voltage source has only a real component ( $\underline{V}_{in} = v_r$ ) and the current is written as  $\underline{I} = i_r + ji_i$ .

Therefore, equation (3.6) becomes

$$Ri_r - Xi_i = v_r \quad (3.7)$$

$$Ri_i + Xi_r = 0 \quad (3.8)$$

In a more compact matrix form, we write

$$Ku = b \quad (3.9)$$

where  $K = \begin{bmatrix} R & -X \\ X & R \end{bmatrix}$ ,  $u = \begin{bmatrix} i_r \\ i_i \end{bmatrix}$  and  $b = \begin{bmatrix} v_r \\ 0 \end{bmatrix}$

This system of equation has to be solved to identify  $u$ , *i.e.* the values of the current components, and then the Joule losses can computed by

$$P_j = R(i_r^2 + i_i^2) = Ru^2 \quad (3.10)$$

Thus, we could consider that  $R$  and  $X$  as the input variables of the model and an output  $P_j$ , as shown in Figure 3.8. We denote this model as

$$f(p) = \tilde{f}(p, u) = P_j$$

where  $p = (R, X)$ .  $f$  and  $\tilde{f}$  are different since the first one includes  $u$  implicitly in its definition, while  $\tilde{f}$  considers  $u$  as an independent variable from  $p$ . Practically,  $f$  is the quantity of interest when looking from outside of the box, while  $\tilde{f}$  is only seen inside the box.

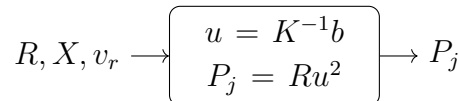


Figure 3.8: Model of the circuit

For the sake of simplification, we will show the derivation of the gradient of the quantity of interest using this example, and interestingly, this derivation extends to the FE model.

### 3.2.1 Derivation of gradient

We denote the partial derivative of a function  $h$  with respect to a variable  $x$  as  $\partial_x h$ . In the same manner, the total derivative is written as  $d_x h$ .

We want to compute the gradient of the function  $f$ , it means, the partial derivatives of the function  $\nabla f = (\partial_R f, \partial_X f)$ .

The equation (3.11) expresses the relation between  $f$  and  $\tilde{f}$

$$\partial_p f = d_p \tilde{f} = \partial_p \tilde{f} + \partial_u \tilde{f} d_p u \quad (3.11)$$

$\partial_p \tilde{f}$  shows the explicit dependence of the function  $\tilde{f}$  on the variable  $p$  while  $\partial_u \tilde{f} d_p u$  shows the implicit dependence of  $p$  through the variable  $u$ . We can note that if  $u$  is not dependent on  $p$  then  $d_p u = 0$ . We retrieve then the fact that the total derivative is equal to the partial derivative.

To compute the gradient of the function  $f$ , we need to compute the total derivatives of  $\tilde{f}$ , *i.e.*  $d_p \tilde{f}$ . In the following derivation, we will only consider  $\tilde{f}$  rather than  $f$ .

We rewrite the equation (3.9) as

$$g(u) = \tilde{g}(p, u) = Ku - b = 0 \quad (3.12)$$

The dependence of  $u$  on the variable  $p$  is expressed in the equation  $\tilde{g}(p, u) = 0$ , Thus, the last term  $d_p u$  of (3.11) is computed by deriving this equation

$$d_p \tilde{g} = \partial_p \tilde{g} + \partial_u \tilde{g} d_p u = 0 \quad (3.13)$$

Then,  $d_p u$  is the solution of the equation

$$d_p u = -(\partial_u \tilde{g})^{-1} \partial_p \tilde{g} \quad (3.14)$$

By replacing  $d_p u$  in (3.11), we get

$$d_p \tilde{f} = \partial_p \tilde{f} - \partial_u \tilde{f} [(\partial_u \tilde{g})^{-1} \partial_p \tilde{g}] \quad (3.15)$$

This formula enables to fully determine the gradient of the function  $f$  versus partial derivatives of the function  $\tilde{f}$  and the equation  $\tilde{g}$ .

For the example above  $\tilde{f}(p = (R, X), u) = Ru^2$  and  $\tilde{g}(p = (R, X), u) = Ku - b$ ,

the quantities forming the gradient are written as

$$\partial_u \tilde{f} = 2Ru^\top \quad (3.16)$$

$$\partial_R \tilde{f} = u^2 \quad (3.17)$$

$$\partial_X \tilde{f} = 0 \quad (3.18)$$

$$\partial_u \tilde{g} = K = \begin{bmatrix} R & -X \\ X & R \end{bmatrix} \quad (3.19)$$

$$\partial_R \tilde{g} = \partial_R Ku = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} u \quad (3.20)$$

$$\partial_X \tilde{g} = \partial_X Ku = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} u \quad (3.21)$$

Thus, the derivative can be written as

$$d_R \tilde{f} = u^2 - 2Ru^\top \begin{bmatrix} K^{-1} \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} u \end{bmatrix} \quad (3.22)$$

$$d_X \tilde{f} = -2Ru^\top \begin{bmatrix} K^{-1} \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} u \end{bmatrix} \quad (3.23)$$

We can notice that we have a general formula to compute the derivative for any value of  $R$  and  $X$ . The derivatives for each variable require solving a system of linear equations of the initial problem as we can see in (3.22) and (3.23) where we have to solve for  $X_R$  and  $X_X$

$$KX_R = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} u \quad (3.24)$$

$$KX_X = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} u \quad (3.25)$$

Solving the equations is the most expensive operation; thus, the cost for computing the gradient is proportional to the time needed multiplied by the number of variables  $p$ . The adjoint variable method enables to tackle this drawback by solving the equations only once.

### 3.2.2 The adjoint variable method

As noted in the last section, the gradient of the function  $f$  is written as

$$\partial_p f = \partial_p \tilde{f} - \partial_u \tilde{f} [(\partial_u \tilde{g})^{-1} \partial_p \tilde{g}] \quad (3.26)$$

By inverting the order of multiplication of the terms above, we get

$$\partial_p f = \partial_p \tilde{f} - \left[ \partial_u \tilde{f} (\partial_u \tilde{g})^{-1} \right] \partial_p \tilde{g} \quad (3.27)$$

The expression between the brackets does not depend on the parameters  $p$ ; it depends only on the solution  $u$  of the equation  $\tilde{g}(p, u) = 0$ . Thus, this operation can be done once and be applied to all the variables  $p$ . We introduce

$$\lambda = -\partial_u \tilde{f} (\partial_u \tilde{g})^{-1} \quad (3.28)$$

where  $\lambda$  is called the **adjoint variable**. Then, the previous expression becomes

$$\partial_p f = \partial_p \tilde{f} + \lambda \partial_p \tilde{g} \quad (3.29)$$

To summarize, for computing the gradient of a function  $f$ , the following steps has to be conducted :

1. Compute the partial derivatives for  $u$  :  $\partial_u \tilde{f}$  and  $\partial_u \tilde{g}$
2. Solve the linear system:  $\lambda \partial_u \tilde{g} = -\partial_u \tilde{f}$  for  $\lambda$
3. Compute the partial derivatives for  $p$  :  $\partial_p \tilde{f}$  and  $\partial_p \tilde{g}$
4. Calculate the adjoint using 3.29

Step 1 is relatively easy to conduct. In fact, the function  $f$  is calculated from the solution  $u$  of the equation  $g(u) = 0$ ; thus, its derivative should be no harder than calculating the function itself. On the other hand, the partial derivative of  $\tilde{g}$  with respect to  $u$  is sometimes already provided by the original model. If the equation  $g(u) = 0$  is a linear system of equations, then its derivative is equal to the matrix of this system. Otherwise, if the equations  $g(u) = 0$  are nonlinear,  $\partial_u \tilde{g} = \partial_u g$  is usually computed to solve the equation using an iterative process, *e.g.* Newton-Raphson, then the derivative is the Jacobian which are often available or at least easy to determine.

Step 2 is the most computationally expensive since it requires solving a system of linear equations of the same size of the system of equation defined by  $g(u) = 0$ . It is worth noting that even if  $g(u) = 0$  is a nonlinear system of equations, the problem to solve in step 2 is linear. Therefore, solving this system will always be less expensive than solving  $g(u) = 0$ , That is to say, that the cost of computing the gradient will be equivalent or less than solving  $g(u) = 0$  for any number of variables. This property makes the adjoint variable method very attractive when dealing with many variables.

Step 3 requires the calculation of the partial derivatives of the  $\tilde{f}$  and  $\tilde{g}$  with respect to  $p$ .  $\partial_p \tilde{f}$  calculation is done as for  $\partial_u \tilde{f}$ . Nevertheless,  $\partial_p \tilde{g}$  may not be straightforward since it involves the derivatives of the system of equations. Sometimes this link between the variables is not as obvious as it was in our example.

Step 4 is straightforward since the total derivatives is given by 3.29 and calculated from the product of terms determined in the previous steps.

### 3.3 Adjoint variable method for FEA

The development of the adjoint variable method for FEA has started in the field of structural engineering, mainly in topology optimization for reducing the volume of devices [93]. Afterwards, some researchers extend its usage for shape and topology optimization of electromagnetic devices [91] [92] [94].

Despite the attractiveness of the method, to the author's knowledge, no commercial software dedicated to electromagnetic design has developed the method. At the same time, it has been present in almost all CFD and structural mechanics software for almost two decades.

In the following section, we will explain how the adjoint variable method can be implemented in a magnetostatic FEA code. The implementation can be extended to solve any other electromagnetic formulation derived from the Maxwell equations.

#### 3.3.1 Finite element model

As detailed in Chapter 1, Maxwell equations are solved using FEM, and the derivation ends by solving a discrete system that reads as follows:

$$\text{Find } u : Ku = b \quad (3.30)$$

where  $K$  is a sparse matrix,  $u$  is the state variable and  $b$  is the source term defined as follows (see 1.80 and 1.81)

$$K_{ij} = \sum_{T \in \mathcal{T}_h} \sum_g \frac{1}{\mu} \nabla \alpha_i^g J_T^{-1} \cdot \nabla \alpha_j^g J_T^{-1} |J_T| w_g \quad (3.31)$$

$$b_j = \sum_{T \in \mathcal{T}_h} \sum_g [J_s \alpha_j^g + \mathbf{M} \cdot \nabla \alpha_j^g J_T^{-1}] |J_T| w_g \quad (3.32)$$

$\alpha_j^g$  and  $w_g$  are, respectively, the basis functions evaluated in the Gauss quadrature points and their weights defined in the reference element,  $\mu$  is the permeability of material,  $J_s$  source current density,  $\mathbf{M}$  is the magnetization of magnets and  $J_T$  is the Jacobian of the mapping from the reference element to an element  $T$  ( $|J_T|$  is determinant of  $J_T$ ).

Additionally, we consider a quantity of interest  $f(p)$  for which we want to compute the gradient.

As for the electrical circuit, we define two functions  $\tilde{f}$  and  $\tilde{g}$  for FEA

$$\tilde{g}(p, u) = K(p, u)u - b(p) = 0 \quad (3.33)$$

$$f(p) = \tilde{f}(p, u) \quad (3.34)$$

This representation of the problem will simplify the derivation of the gradient using the adjoint variable method as for the electrical circuit; thus, we write

$$\partial_p f = \partial_p \tilde{f} + \lambda \partial_p \tilde{g} \quad (3.35)$$

where  $\lambda$  is the adjoint variable and is solution of the linear system of equations

$$\lambda \partial_u \tilde{g} = -\partial_u \tilde{f} \quad (3.36)$$

The solution of the linear system 3.36 is the most computationally expensive operation, and it is independent of the variables  $p$ . Thus, the gradient computation cost is independent of the number of variables.

Computing the gradient using the adjoint method is reduced to compute the partial derivatives with respect to state variable  $u$  ( $\partial_u \tilde{f}$ ,  $\partial_u \tilde{g}$ ) and those with respect to design variables  $p$  ( $\partial_p \tilde{f}$ ,  $\partial_p \tilde{g}$ ).

### 3.3.2 Derivatives for state variables

Derivatives with respect to the state variables are relatively simple to express.  $\partial_u \tilde{g}$  is generally always computed in the initial FEA code.

If there is no nonlinear ferromagnetic material ( $\partial_u \mu = 0$ )

$$\partial_u \tilde{g} = K \quad (3.37)$$

Otherwise ( $\partial_u \mu \neq 0$ )

$$\partial_u \tilde{g} = \partial_u (Ku) \quad (3.38)$$

For solving  $\tilde{g}(p, u) = 0$ , the Jacobian matrix  $\partial_u \tilde{g}$  is necessary for a nonlinear solver such as Newton-Raphson. Thus, this quantity could be retrieved without any additional effort.

On the other hand,  $\partial_u \tilde{f}$  can be computed efficiently because the function  $f$  is given by an explicit expression of the state variable  $u$ . For example, if we consider that the quantity of interest is the magnetic energy that is written, as shown in Chapter 1

$$\tilde{f}(p, u) = W = \frac{1}{2} u^\top b, \quad (3.39)$$

then, the derivative with respect to  $u$  is

$$\partial_u \tilde{f} = \frac{1}{2} b^\top, \quad (3.40)$$

For other quantities of interest  $f$ ,  $\partial_u \tilde{f}$  could be derived and implemented with ease.



### 3.3.3 Derivatives for design variables

The derivatives of the quantities  $\tilde{f}$  and  $\tilde{g}$  with respect to the design variables could be somewhat challenging due to the variety of parameters that could be considered. Often, in the literature, researchers separate between geometric variables and physical variables where the first are often referred to as shape sensitivity analysis [91] and the latter are mainly used in the context of topology optimization [95]. In this dissertation, we will treat both types of variables.

#### A Geometric variables

Geometric variables are the ones that involve changing the geometry of the device studied. A geometric definition of the problem must be made before starting optimization and uncertainty quantification processes. The choice of variables is of paramount importance since it is equivalent to define the mathematical model of the problem studied. Clearly, it defines the nature, and the dimensions of the research space and possible solutions largely depend on it. The computation of the derivatives highly depends on the way the geometry is parameterized.

#### Parameterization

In the most straightforward configuration, the shape parameters can be considered as merely a collection of points in 3D space. The analysis, in this case, involves moving each of the points in the desired direction by some small amount and determining the effect on the function. This process is both complicated and computationally expensive. Shape parameterization exists as an effort to overcome these complexities and inefficiencies.

Shape parameterization is the method of determining a set of parameters that control the size and shape of the device to be designed. This is conducted by determining some driving parameters and reducing the number of design variables from the number of 3D space points to the number of parameters.

There exist many methods for geometry parameterization, such as free-form deformation [96], polynomial and spline (NURBS) [97], iso-geometric analysis [98], and computer-aided design (CAD). Among these categories, CAD appears to have some capability that makes it very attractive for design engineers such as efficiency, compactness and suitability for complex configurations. Besides, CAD-based parameterization methods can achieve significant geometry changes. Another advantage of CAD-based is the availability of a comprehensive set of geometric functionalities provided by commercial CAD systems. Parameterizing a complex model is still a challenging task with today's CAD systems.

Furthermore, they are not capable of calculating derivatives analytically. Computer codes for commercial CAD systems are huge; to differentiate an entire system with automatic differentiation tools may be a non-trivial task. Therefore, the calcu-

lation of derivatives of geometry with respect to design variables could be challenging within commercial CAD software.

The derivative of the quantities  $\tilde{f}$  and  $\tilde{g}$  with respect to a geometric design variable  $p$  can be written as follows

$$\partial_p \tilde{f} = \sum_{i=1}^N \left[ \partial_{x_i} \tilde{f} d_p x_i + \partial_{y_i} \tilde{f} d_p y_i \right] \quad (3.41)$$

$$\partial_p \tilde{g} = \sum_{i=1}^N \left[ \partial_{x_i} \tilde{g} d_p x_i + \partial_{y_i} \tilde{g} d_p y_i \right] \quad (3.42)$$

where  $(x_i, y_i)$  for  $i = 1, \dots, N$  are the node coordinates used for the FEA, by writing the quantities in this form, we decouple the dependence the quantities related to the FEA from the geometry definition. Furthermore, we decouple the computation cost from the number of design variables since  $\partial_{x_i} \tilde{f}$ ,  $\partial_{y_i} \tilde{f}$ ,  $\partial_{x_i} \tilde{g}$  and  $\partial_{y_i} \tilde{g}$  are independent of  $p$ .

These quantities can be computed analytically.  $\partial_{x_i} \tilde{g}$  and  $\partial_{y_i} \tilde{g}$  can be calculated based on (3.31), (3.32) and (3.33). In fact, the only quantity that depends on the mesh coordinates is the mapping from the reference element to an arbitrary element. Thus, only the derivative of the mapping is needed for the computation of the quantities.

We have

$$\partial_{x_i} \tilde{g} = \partial_{x_i} [Ku - b] \quad (3.43)$$

$$\partial_{y_i} \tilde{g} = \partial_{y_i} [Ku - b] \quad (3.44)$$

Moreover, as detailed in Chapter 1,  $K$  and  $b$  are computed using the reference element concept, as shown in Figure 3.9. Equation (3.45) shows how the reference element coordinates  $(x_r, y_r)$  are transformed into the coordinates  $(x, y)$  of an arbitrary element using the mesh nodes  $(x_0, x_1, x_2, y_0, y_1, y_2)$  defining this element.

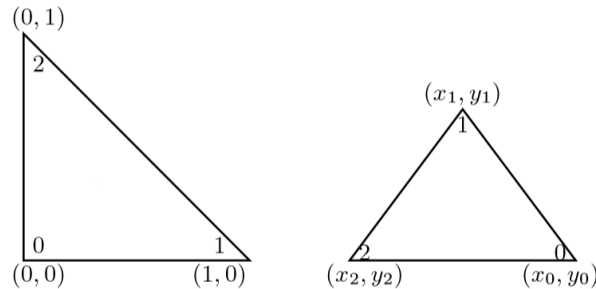


Figure 3.9: (left) The reference element, (right) an arbitrary element.

$$(x, y) = (x_0, y_0) + (x_r, y_r) J_T \quad (3.45)$$

with

$$J_T = \begin{pmatrix} x_1 - x_0 & y_1 - y_0 \\ x_2 - x_0 & y_2 - y_0 \end{pmatrix}$$

The derivative of the mapping with respect to  $x_0, x_1, x_2, y_0, y_1$  and  $y_2$  is easy to compute; hence,  $\partial_{x_i} \tilde{g}$  and  $\partial_{y_i} \tilde{g}$  can be assembled as done for  $\tilde{g}$ .

Now, let us go up to the other part of equations (3.41) and (3.42):  $d_p x_i$  and  $d_p y_i$ . These quantities somewhat link the variation in the node coordinates versus the change in geometry. Hence, they represent the real link to the CAD model after the latter has been meshed. In this dissertation, we will show how to compute these quantities without intrusive manipulation, neither in the CAD software nor in the mesh tool.

Geometric parameters are related to one or many edges that separate different regions (faces), as shown in Figure 3.10. This numbering is very useful for FEA since it enables to impose external conditions. For example, we used  $E2$  to impose a Dirichlet boundary condition, and the face  $F3$  is used for imposing a current density  $J$ .

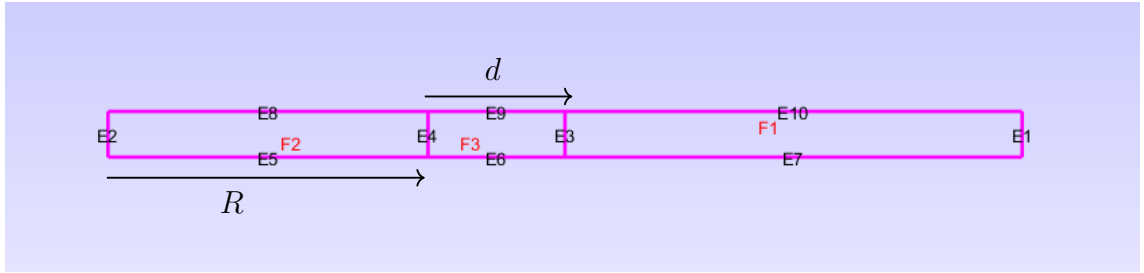


Figure 3.10: Edges and faces numbering

We use this numbering for calculating the desired quantities. In this example, we have two geometric design variables  $R$  and  $d$ , as shown in Figure 3.10. These variables enable the positioning and sizing of the coil modelled by the face  $F3$ ; thus, they mainly act on edges surrounding the coil, namely  $E3, E4, E6$  and  $E9$ . Hence,  $d_p x_i$  and  $d_p y_i$  are computed only for the mesh nodes that lie on these edges and not on all the other nodes. Therefore, we write

$$d_p x_i = \begin{cases} d_p x_i, & \text{if node } i \text{ is on } E3, E4, E6 \text{ or } E9. \\ 0, & \text{otherwise.} \end{cases} \quad (3.46)$$

$$d_p y_i = \begin{cases} d_p y_i, & \text{if node } i \text{ is on } E3, E4, E6 \text{ or } E9. \\ 0, & \text{otherwise.} \end{cases} \quad (3.47)$$

The edges  $E3, E4, E6$  and  $E9$  define a rectangle. The objective is to define the rectangle contour using a parametric equation as follows

$$P(t) = (x(t), y(t)) \tag{3.48}$$

In general, one could propose many ways to parameterize a shape. For the rectangle, we could parameterize each edge independently and compute the derivative or use a single parametrization for the whole rectangle as below

$$\begin{aligned} x(t) &= R + \frac{d}{2} [1 + \text{sign}(\cos(t))] \\ y(t) &= \frac{h}{2} [1 + \text{sign}(\sin(t))] \end{aligned}$$

and  $t \in [0, 2\pi[$ .

The derivatives of  $x(t)$  and  $y(t)$  with respect to the design variables are computed with ease,

$$d_R x = 1 \tag{3.49}$$

$$d_R y = 0 \tag{3.50}$$

$$d_d x = \frac{1}{2} [1 + \text{sign}(\cos(t))] = \frac{x - R}{d} \tag{3.51}$$

$$d_d y = 0 \tag{3.52}$$

Equations (3.49)-(3.52) are evaluated using the node coordinates previously identified, then, used in equations (3.41) and (3.42) to compute  $\partial_p \tilde{f}$  and  $\partial_p \tilde{g}$ .

Now, we have everything for the computation of the gradient of a quantity of interest with respect to geometry variables using the adjoint variable method.

In this paragraph, we computed the shape sensitivities  $d_p x_i$  and  $d_p y_i$  for a rectangular shape using a parametric equation. This framework is extendable to different types of geometry and design variables. In appendix A, we derive the shape sensitivities for some recurrent types of geometries.

## B Physical variable

The physical variables that can be considered in electromagnetic modelling are the permeability of ferromagnetic materials, the coercive field of magnets, and the imposed current density. This kind of modelling is commonly used in the context of topology optimization, where the objective, in general, is to reduce the volume of material present in a device. Topology optimization was first introduced in the context of structural optimization before being extended to other physical fields such as heat transfer, fluid dynamics and electromagnetism.

In this dissertation, the objective is to derive the formula that enables to compute the gradient for parameters that act on the physical properties of electromagnetic devices.

### Permeability

If a parameter  $p$  controls how  $\mu$  varies, we can compute the derivatives  $\partial_p \tilde{f}$ ,  $\partial_p \tilde{g}$  can be computed :

$$\partial_p \tilde{g} = \partial_p [Ku - b] = [\partial_p K] u \quad (3.53)$$

with  $\partial_p K_{ij} = K_{ij} \frac{-\partial_p \mu}{\mu}$ .

The quantity  $\partial_p \mu$  highlights the sensitivity of  $\mu$  with respect to the design variable  $p$ . In the literature, we can find multiple ways of modelling. For example, the density method uses a polynomial mapping for the parameterization [94].

$$\mu = \mu_{air} + (\mu_{iron} - \mu_{air}) p^n \quad (3.54)$$

with  $\mu_{air}$  and  $\mu_{iron}$  are respectively air and iron permeabilities, A3. This method enables us to chose where in the studied domain to put iron or air. In this case,  $\partial_p \mu$  is written as

$$\partial_p \mu = n(\mu_{iron} - \mu_{air}) p^{n-1} \quad (3.55)$$

### Coercive field

As for the permeability, if  $p$  controls the magnetization of a permanent magnet, one could, in a similar manner, deduce the derivatives

$$\partial_p \tilde{g} = \partial_p [Ku - b] = -\partial_p b \quad (3.56)$$

with  $\partial_p b_j = \sum_{T \in \mathcal{T}_h} \sum_g [\partial_p \mathbf{M} \cdot \nabla \alpha_j^g J_T^{-1}] |J_T| w_g$ .

The quantity  $\partial_p \mathbf{M}$  is computed based on the design variable;  $p$  can be the magnitude, the direction of the coercive field, *etc.*

### Imposed current density

Following the same principle, the derivative with respect to a variable that controls the current density is

$$\partial_p \tilde{g} = \partial_p [Ku - b] = -\partial_p b \quad (3.57)$$

with  $\partial_p b_j = \sum_{T \in \mathcal{T}_h} \sum_g [\partial_p J_s \alpha_j^g] |J_T| w_g$ .

### 3.3.4 Discussion

In the precedent section, we showed the derivation of the gradient of a quantity of interest from a finite element code using the adjoint variable method. It is worth noting that two approaches exist to calculate the adjoint: the continuous approach and the discrete approach. The continuous approach aims to compute the gradient from the governing equations (Maxwell equations) before discretization, while the discrete approach calculates the gradient from the discrete equations. We focused and developed the discrete approach for many important reasons :

1. Relatively simple implementation (but tedious)
2. Implementation of the derivatives of each subroutine/process individually in analysis code
3. Build up larger components by chain rule differentiation of analysis code
4. Ability to check the derivatives by finite-difference while debugging

The discrete adjoint approach can be implemented in a modular fashion using the same data-structures/solution strategy as analysis as shown in Figure 3.11.

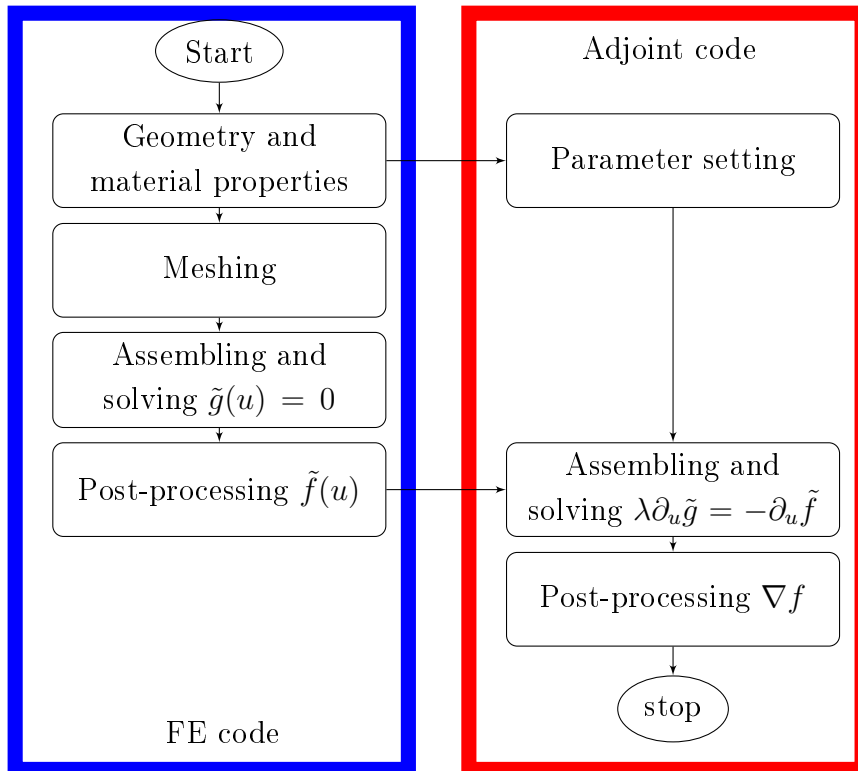


Figure 3.11: Flowchart of a modular adjoint implementation

The adjoint is composed of three main steps, at parameter setting, one should compute the sensitivities with respect to the design variables (geometric and physical), then, the assembling of the quantities  $\partial_u \tilde{f}$ ,  $\partial_u \tilde{g}$ ,  $\partial_{x_i} \tilde{f}$ ,  $\partial_{x_i} \tilde{g}$ ,  $\partial_{y_i} \tilde{f}$ ,  $\partial_{y_i} \tilde{g}$  and the

ones related to the physical variables is conducted, afterwards the adjoint problem is solved, and finally, the gradient of the quantity of interest is computed as

$$\begin{aligned}\nabla f = & \left[ \partial_{x_i} \tilde{f} + \lambda \partial_{x_i} \tilde{g} \right] d_p x_i \\ & + \left[ \partial_{y_i} \tilde{f} + \lambda \partial_{y_i} \tilde{g} \right] d_p y_i \\ & + \left[ \partial_{\mu} \tilde{f} + \lambda \partial_{\mu} \tilde{g} \right] d_p \mu \\ & + \left[ \partial_{\mathbf{M}} \tilde{f} + \lambda \partial_{\mathbf{M}} \tilde{g} \right] d_p \mathbf{M} \\ & + \left[ \partial_{J_s} \tilde{f} + \lambda \partial_{J_s} \tilde{g} \right] d_p J_s\end{aligned}$$

In this equation, one can see that all the dependencies on the design variables  $p$  are outside of the most computationally expensive operations, namely the solution of the adjoint problem and the assembling of the different quantities. The independence of the number of design variables makes the adjoint variable method very attractive for treating problems with large numbers of design variables and few numbers of objectives.

To show the utility and effectiveness of the adjoint variable method, in the next section, we treat some examples to highlight the advantages of the adjoint variable approach over the finite-difference in computational cost and precision.

## 3.4 Examples

### 3.4.1 Solenoid model

We consider the device treated at the beginning of the chapter, where we have made a comparison between analytic and FEA computation of the quantities of interest. We conduct the same comparison on the gradient of the quantities of interest as shown in Table 3.2.

	Analytic	Adjoint	Rel. error %
$\partial_R B_c$	0	0.006	0.60
$\partial_d B_c$	0.0126	0.0125	0.38
$\partial_J B_c$	0.376	0.359	4.39
$\partial_R W$	5.655	5.655	0.00
$\partial_d W$	32.044	32.032	0.04
$\partial_J W$	904.78	904.5	0.03

Table 3.2: Comparison of analytic and and adjoint variable gradients

One can notice that the gradient of  $B_c$  and  $W$  computed by the adjoint variable method are corresponding to the ones computed analytically with a relative error of the same magnitude as for the quantities  $B_c$  and  $W$ .

### 3.4.2 TEAM workshop 22

This example is considered as one of the benchmarks for optimization problems of electromagnetic devices. The objective here is compare the gradient computed finite-difference to the one computed using the adjoint variable method (further details in Chapter 4). To validate the effectiveness of the adjoint variable method, the approach is tested against the gradient computed using centred difference (CD)<sup>1</sup> using mesh morphing technique and with remeshing technique. The mesh morphing (MM) technique enables "small" changes of the shape while maintaining the topology (node connectivity) of the mesh. The aim of using such a technique is to reduce the errors due to re-meshing in the comparison (see example treated at section 1).

The results of the comparison are summarized in Table 3.3. It shows the gradient of the problem's objective function in the optimum. For more details, the reader may refer to reference [1].

Method	$\varepsilon$	$p$								Rel. Error	Time (s)
		$R_1$	$R_2$	$h_1/2$	$h_2/2$	$d_1$	$d_2$	$J_1$	$J_2$		
Adjoint	-	24.827	-19.222	13.725	-10.619	25.852	-81.824	8.418e-07	8.444e-07	-	1.27
CD w. MM	$10^{-5}$	24.803	-19.212	13.726	-10.616	25.855	-81.817	8.418e-07	8.444e-07	0.1%	4.06
CD w/ MM	$10^{-3}$	36.497	-23.592	15.756	-19.098	35.757	-93.534	8.418e-07	8.444e-07	155%	7.02

Table 3.3: Gradient Comparison for TEAM workshop 22

For finite-difference schemes, we tried different values of  $\varepsilon$ , ranging from  $10^{-10}$  to  $10^{-2}$ . In the table, only the values that correspond to the best results in terms of relative error are shown.

When using mesh morphing (CD w. MM), the gradients are coherent in less than 1% error relative to the gradient computed with the adjoint with a speedup of 3.2. This speedup is attained thanks to the main property of the adjoint variable method that solves fewer equations. However, when not using mesh morphing (CD w/ MM), which means generating a new mesh for each geometry change, re-meshing errors appear to perturb the gradient information highly (relative error higher than 150%).

It is worth noting that the difference in time for CD w. MM and CD w/ MM is related to the overhead of regenerating a new mesh for all geometry variations.

In general, mesh morphing is not always a simple task; in this example, it was possible to morph to mesh nodes to correspond to the geometry change thanks to the simple shapes involved in the modelled device (rectangular regions). Thus, without using mesh-morphing, the finite-difference method may lead to significant errors when compared to the adjoint variable method. Furthermore, a speedup from 5.5 was attained for the computation of the gradient, which confirms the efficiency of the approach.

---

<sup>1</sup>  $\frac{dOF}{dp_i}(p) \approx \frac{OF(p_1, \dots, p_i + \varepsilon, \dots) - OF(p_1, \dots, p_i - \varepsilon, \dots)}{2\varepsilon}$



## 3.5 Chapter Summary

In this chapter, we have presented the adjoint state variable method and how it can be applied to a FEM code. We have developed an efficient way to compute the derivatives of the shape sensitivities, which are vital for the computation of the gradient. Sometimes, finite-difference could be used with a mesh-morphing strategy for computing the gradient but it can have redhibitory effect since the mesh displacement is not always straightforward and can lead to non-conforming mesh. The adjoint variable method was compared to the finite-difference one to validate and highlight its effectiveness in terms of precision and computational time.

Geometric parametrization of shape variables is still one of the shortcomings of the method. We presented an approach based on the parametric equations of the geometric shapes, however, for very complex shapes, this can be very cumbersome. In the following Chapter, the gradient computed using the adjoint variable method is provided to a derivative-based optimization algorithm for the analysis of two benchmarks from the literature and comparison of the performances to the non-intrusive approach detailed in Chapter 2.

# Chapter 4

## Applications and benchmarking

This chapter is dedicated to numerical tests of the developed approaches in the context of optimization. For this task, we address two well-known benchmarks treated by electromagnetic community researchers. They are the ones of the TEAM (Testing Electromagnetic Analysis Methods) workshops [1] [2]. Since the comparison of results to other researchers is somewhat tricky because of the differences related to the mesh and the numerical methods that can be used such as the formulation, the integration technique, ..., we solve these benchmarks with the approaches we have developed and some of the algorithms that are conventionally found in the literature.

The test cases are the TEAM workshop problems 22 and 25, there exist two publications describing both of them [1] [2]. We solve both problems using four methods; the non-intrusive approach developed in Chapter 2, an SQP algorithm assisted by the adjoint variable method for computing the gradient, the DIRECT algorithm [99] and the genetic algorithm (GA of [100]).

In the following of this chapter, we detail the test cases and their related optimization problems, afterwards, we present the settings of each optimization algorithm for both test cases, and lastly, a comparison of the results is conducted.

### 4.1 Test cases

#### 4.1.1 TEAM Workshop Problem 22

The Superconducting Magnetic Energy Storage (SMES) device in Figure 4.1 consists in two concentric superconducting coils fed by currents that flow in opposite directions [1]. The inner coil is used for storing magnetic energy  $E$ , while the outer one has the role of diminishing the magnetic stray field  $B_{stray}$  computed on *line a* and *line b*.

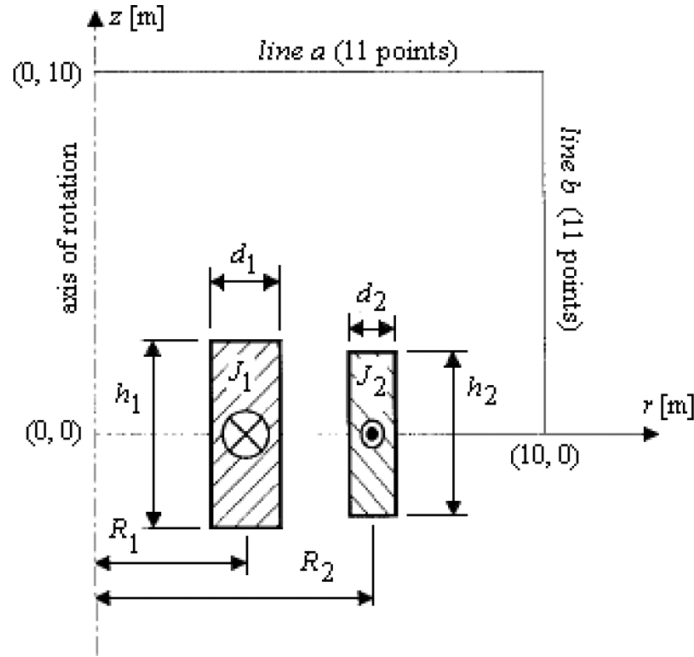


Figure 4.1: SMES Device [1]

## A FEM model

We have created a model for the benchmark with our FEA tool (MagFEM); as the model is symmetric with respect to the  $r$ -axis (the antisymmetric case is somewhat different than the 2D case. In appendix B, we detail the corresponding formulation). In this part, we discuss some aspect of the model mainly related to the parameterization and the FEA.

### Parameterization

The geometry of the device is simple; it is composed of three rectangles, as shown in Figure 4.2; two for the coils and one limiting the studied domain  $([0, 15] \times [0, 15])$  which holds the two coils.

The variables considered in the optimization problem are the ones related to the coils (position, size, current density imposed). The variables  $(R_1, R_2, h_1/2, h_2/2, d_1, d_2)$  are used for the parameterization of the geometry, while the current densities  $(J_1, J_2)$  are physical variables that are parametrized in the FE model.

### Simulation

In the FEA, the geometry is meshed as shown in Figure 4.3 on the left. A fine mesh is imposed around the coils and coarse one far from them. On the right of Figure 4.3, the distribution of the flux density around the coils is shown.

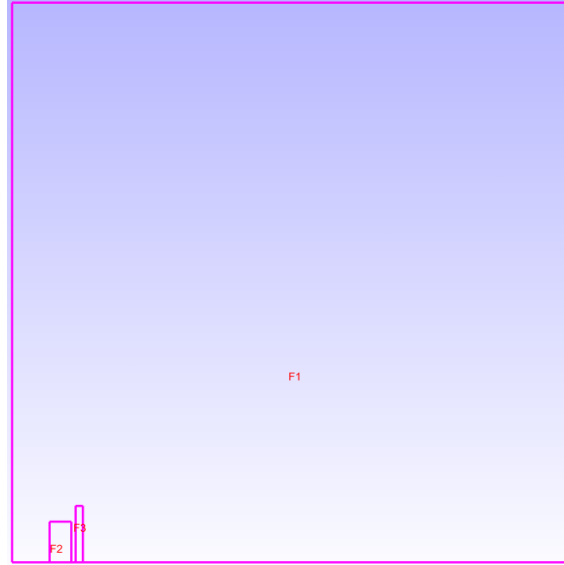


Figure 4.2: Modelled geometry of the SMES

## B Optimization problem

The goal of the optimization problem is to find the design configurations (8 design variables) that give a specified value of stored magnetic energy and a minimal magnetic stray field while satisfying some constraints. Mathematically, this is formulated as

$$\min_x OF(x) = B_{stray}^2(p)/B_{norm}^2 + |E(p) - E_{ref}|/E_{ref} \quad (4.1)$$

$$s.t. |\mathbf{J}| + 6.4|\mathbf{B}| - 54 \leq 0 \quad (4.2)$$

$$R_1 - R_2 + \frac{1}{2}(d_1 + d_2) < 0 \quad (4.3)$$

where  $E_{ref} = 180MJ$ ,  $B_{norm} = 200\mu T$  and  $p$  are the design variables and their bounds are shown in Table 4.1.

$p$	$R_1$	$R_2$	$h_1/2$	$h_2/2$	$d_1$	$d_2$	$J_1$	$J_2$
min	1.0	1.8	0.1	0.1	0.1	0.1	10	-30
max	4.0	5.0	1.8	1.8	0.8	0.8	30	-10

Table 4.1: Bounds of design variables for TEAM problem 22

The constraint (4.2) aims to limit the maximal flux density ( $|\mathbf{B}|$ ) in the coils to ensure the quench condition [1]. As shown in Figure 4.3, the maximal values of  $|\mathbf{B}|$  are located on the boundaries of the coils and specifically on coordinates  $P_1 = (R_1 - d_1/2, 0)$ ,  $P_2 = (R_1 + d_1/2, 0)$  and  $P_3 = (R_2 - d_2/2, 0)$  (shown by cross sign in Figure 4.3 on the right). Thus, we replace this constraint by three ones (4.5),(4.6) and (4.7).

The second constraint aims to prevent both coils from overlapping. Unfortunately, optimization algorithms can sometimes during iterations violate some con-

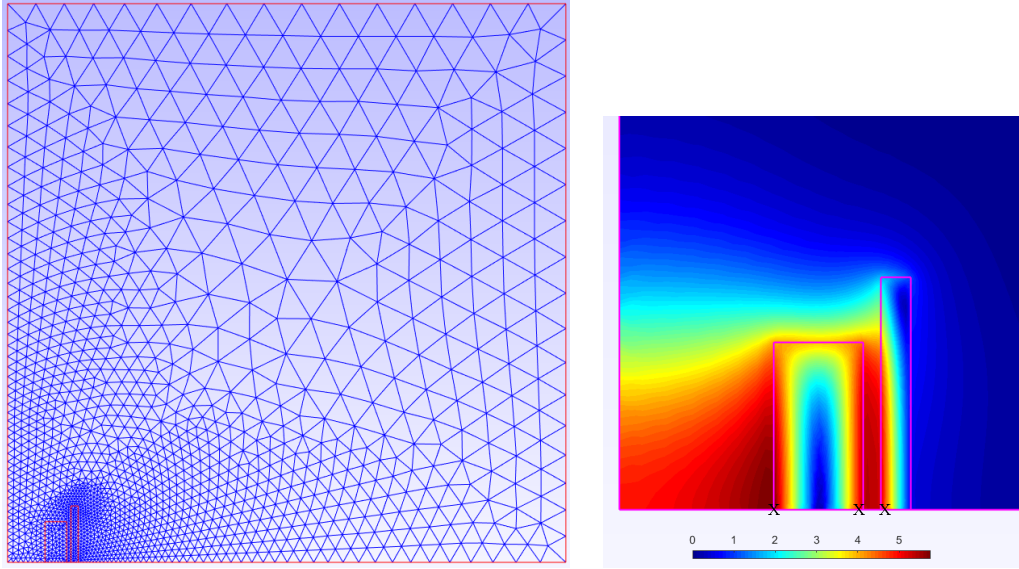


Figure 4.3: Mesh of studied domain (left), enlarged view of the flux density distribution around the coils (right)

straints, and this implies taking some special care of this issue that depends on the type of algorithm used. If this constraint is violated, the model is not physically valid; thus, the other quantities such as the objective function cannot be computed. Some optimization algorithms can handle this issue if the model returns a high value of the objective function, but others are not capable of dealing with this kind of constraint.

As we are comparing different approaches, we chose to define another optimization problem equivalent to the initial one that avoids the issue mentioned above. We define a variable  $A_2$  as

$$A_2 = R_2 - R_1 - (d_1 + d_2)/2$$

that replaces  $R_2$ . Using interval arithmetic, we compute the bounds of this new variable  $[-3, 3.9]$ . By using the constraint (4.3) that is preventing the two coils from overlapping, we impose  $A_2$  to be higher than 0.001. Then, the interval of variation of  $A_2$  is  $[0.001, 3.9]$ .

Introducing this new variable is somewhat relaxing the initial optimization problem by allowing the variable  $R_2 = R_1 + A_2 + (d_1 + d_2)/2$  to vary in a more significant interval  $[1.101, 8.7]$  than the one defined in the initial problem  $[1.8, 5]$ . Therefore, two constraints (4.8) and (4.9) are added. Finally, the optimization problem

becomes

$$\min_p OF(p) = B_{stray}^2(p)/B_{norm}^2 + |E(p) - E_{ref}|/E_{ref} \quad (4.4)$$

$$s.t. \quad (J_1 - 54)/6.4 + |\mathbf{B}(p, P_1)| \leq 0 \quad (4.5)$$

$$(J_1 - 54)/6.4 + |\mathbf{B}(p, P_2)| \leq 0 \quad (4.6)$$

$$(-J_2 - 54)/6.4 + |\mathbf{B}(p, P_3)| \leq 0 \quad (4.7)$$

$$R_1 + A_2 + (d_1 + d_2)/2 \leq 5 \quad (4.8)$$

$$-R_1 - A_2 - (d_1 + d_2)/2 \leq -1.8 \quad (4.9)$$

where  $p$  are the design variables  $p = (R_1, A_2, h_1/2, h_2/2, d_1, d_2, J_1, J_2)$ .

### 4.1.2 TEAM Workshop Problem 25

In this test case, the device is used to orient the magnetic powder and produce anisotropic permanent magnets [2]. The magnetic powder is inserted in the cavity. The orientation and strength of the magnetic field should be controlled in order to obtain the required magnetization. A coil creates the magnetic field. The current density is fixed to  $1.239219A/mm^2$ . The die press and electromagnet are made of steel with a non-linear permeability. The geometry of the whole device is shown in Figure 4.4.

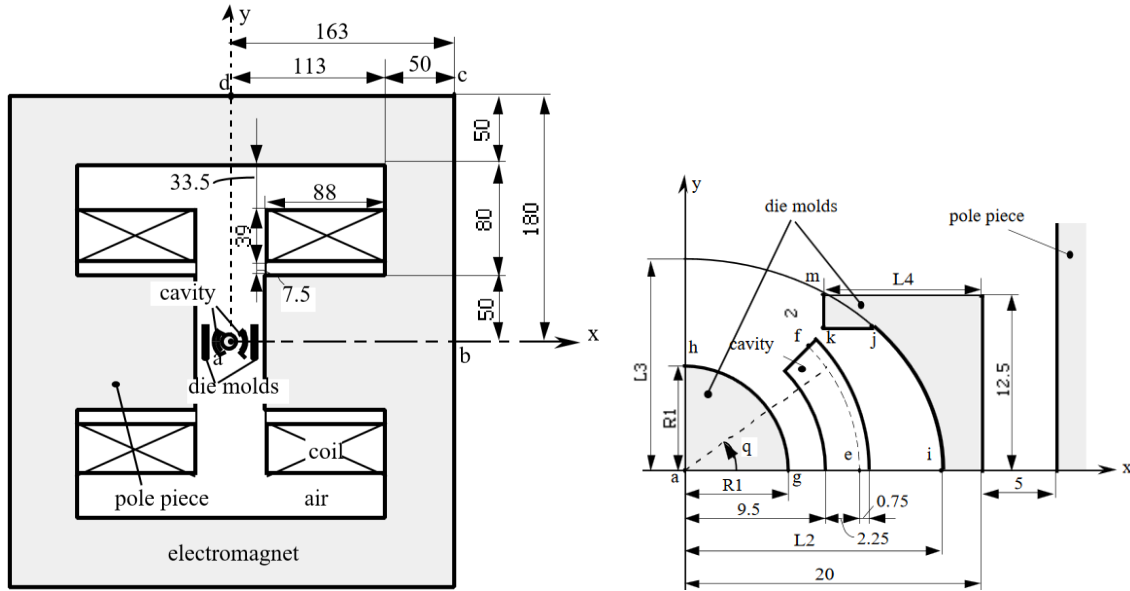


Figure 4.4: Model of die press with electromagnet, Whole view (left), Enlarged view (right) [2]

#### A FEM model

A model of the device is created. By exploiting symmetries, only a quarter of the device is sufficient for FEA.

### Parameterization

The geometry of the device is created and parametrized on MagFEM, as shown in Figure 4.5. The geometry is composed of six regions ( $F1-F6$ ). The region  $F2$  is used for imposing the current density while regions  $F1$ ,  $F4$  and  $F5$  are used for the ferromagnetic material. Other regions are considered as air.

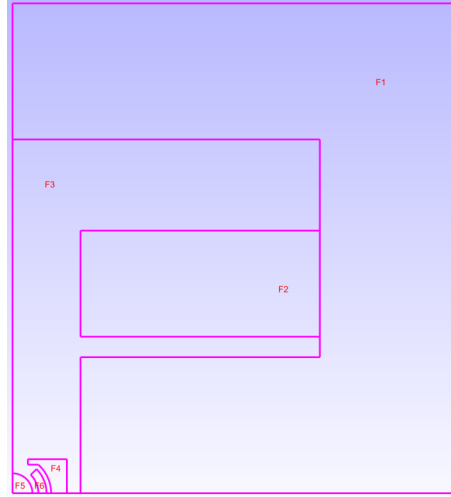


Figure 4.5: Modelled geometry of the die press

The variables considered in the optimization are related to the mould's shape of the die press. The inner mould is controlled by the variable  $R1$  while the outer mould is controlled by the variables ( $L2, L3, L4$ ). The parameterization is somewhat more challenging than the first test case because of the elliptical shape of the outer mould.

### Simulation

In the FEA, the geometry is meshed as shown in Figure 4.6 on the left. A fine mesh is imposed around the moulds and coarse mesh far from them. At right, in the same figure, the distribution of the flux density is shown.

## B Optimization problem

The objective of the shape optimization is to obtain a flux density that is radial in the cavity space and with a constant magnitude of 0.35T. The objective function  $W$  is the squared error between the  $B_x$  and  $B_y$  values sampled in 10 positions along the arc e-f shown in Figure 4.4.

$$\min_p W(p) = \sum_{i=1}^{10} (B_{xip} - B_{xio})^2 + (B_{yip} - B_{yio})^2 \quad (4.10)$$

where  $p = (R1, L2, L3, L4)$  are the design variables (their bounds are shown in Table 4.2) and  $B_{xio} = 0.35 \cos(\frac{\pi}{40}i)$ ,  $B_{yio} = 0.35 \sin(\frac{\pi}{40}i)$  and  $B_{xip}$  and  $B_{yip}$  are computed by the FEA of the device.

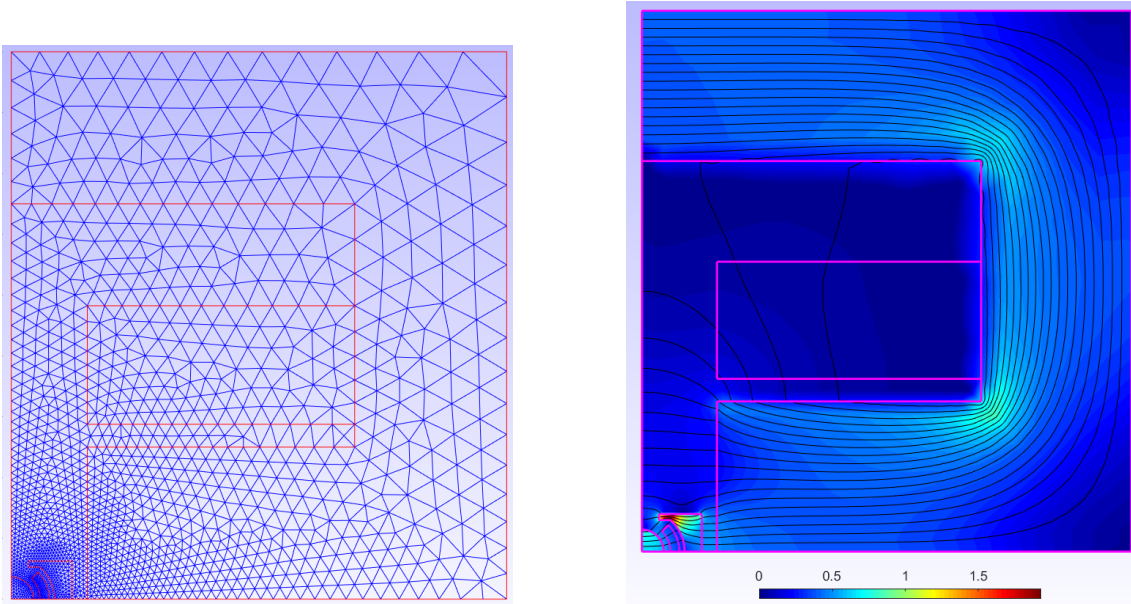


Figure 4.6: The mesh of studied domain (left), The flux density distribution in the studied domain (right)

$p$	$R1$	$L2$	$L3$	$L4$
min	5.0	12.6	14	4
max	9.4	18	45	19

Table 4.2: Bounds of design variables for TEAM problem 25

## 4.2 Algorithms settings

In this section, we present the configuration of the optimization algorithms. A set of parameters are defined based on our experience from using each algorithm.

### 4.2.1 B2M2 algorithm

The branch and bound based meta-model (B2M2) algorithm is detailed in Chapter 2, Section 2. Some options need to be configured, mainly the depth level and the number of FEA evaluations needed for fitting the meta-models.

**Initial design :** The initial design of each sub-space is created using a space-filling LHS of size two times the number of variables. This may seem few, but as the algorithm progresses, the sample points of a parent space will be added to the initial design of its sub-spaces. Thus, the sample will grow as the sub-division of the space continues.

**Infill points :** Additional sample points are added in two different manners; the first one aims to further explore the design space by adding points where the



prediction error is high, while the second one enables the exploitation of the best solution by adding new sample points that maximize the expected improvement ISC.

The number of sample points to be added by each ISC (infill sampling criterion) is set to twice the number of variables. However, for expected improvement, it may be stopped before this limit if there is no improvement in sampling additional points.

Globally, in each sub-space, there will be sampled at most six times the number of variables.

**Stopping criteria :** The algorithm is set to stop if it attains a predefined depth level. The maximum depth level is set to 10.

### 4.2.2 SQP algorithm assisted by adjoint variable method

We use SQP algorithm from the Matlab Optimization toolbox with the gradient of the quantities of interest computed using the adjoint variable method.

**Multi-start :** To cope with the local search of the SQP method, we use a multi-start strategy. We perform 100 runs with different initial points; These points are sampled using an LHS design to obtain a uniform distribution on the whole design space.

**Stopping criteria :** The stopping criterion is based on the *StepTolerance* option, meaning iterations end when the relative change in the solution is less than  $10^{-8}$ .

### 4.2.3 DIRECT

We use DIRECT implementation of [99]. The default options are maintained except for the number of iterations.

**Constraints :** The constraints are handled by penalty method by minimizing the penalized objective function

$$OF(p) + \sum_i \lambda \max(0, g_i(p))$$

where  $\lambda$  is the penalty factor and has been chosen to be equal to 10.

**Stopping criteria :** As the algorithm does not have an appropriate stopping criterion except the number of iterations and the number of evaluations. We stop the algorithm manually when there is no improvement in the objective function in 50 successive iterations.

## 4.2.4 Genetic algorithm

We use GA from Matlab Global Optimization toolbox [100].

**Multi-run :** As the randomness inherent in genetic algorithm, meaning solving twice the same problem may not lead to the same solution, we adopted a multi-run strategy by performing 50 runs of the algorithm.

**Population :** The population options are set as follows

1. Size : 100 individual for unconstrained case and 200 for constrained one.
2. Selection function is uniform
3. Crossover function is "scattered"
4. Crossover fraction is 80% of the population
5. Mutation function is Gaussian
6. Elite fraction is 5% of the population

**Constraints :** The constraints are handled by the penalty method as defined in [27].

**Stopping criteria :** The algorithm is stopped if the value of the fitness function did not change during 50 generations.

## 4.3 Results and comparison

### 4.3.1 Comparison protocol

The comparison of the performance of different algorithms will be based on two criteria: the cost and the quality of the solution.

The quality of the solution stands for the value of the objective function found. The smaller, the better.

On the other hand, the cost of the optimization stands for the number of evaluations needed to attain the solution. As we compare different approaches, we propose a probabilistic metric to measure the cost.

The convergence rate (CR) is the percentage of the number of runs that converged to the best solution. The convergence rate concerns mainly SQP and GA since we perform multiple runs and the number of runs needed is unknown beforehand. To compute CR, we look, among the solutions of each run, for the best one (smallest

value), then we count the number of runs that converged to that solution within a tolerance of 1% of the design range.

Once the convergence rate is determined, we write the probability that at least one optimization lead to the best solution is

$$P = 1 - (1 - C.R.)^n$$

where  $n$  is the number of optimization runs to be considered.

If we want to consider a high probability ( $P = 99.73\%$ ) of finding the best solution, we can compute the expected number of runs needed  $n$

$$n = \frac{\ln(1 - P)}{\ln(1 - CR)} \quad (4.11)$$

Then, the expected total number of evaluations (Expected # FEM evals) is  $n$  times the total average number of evaluations performed.

$$\text{Expected \# FEM evals} = n \text{ average}(\# \text{ FEM evals}) \quad (4.12)$$

A particular property related to SQP is the computation of the gradient using the adjoint variable method. The cost of computing the gradient is equivalent to evaluating the FEA once; thus, the # FEM evals are doubled for the SQP method.

### 4.3.2 TEAM Workshop Problem 22

The TABLE 4.3 summerizes the optimization results

Table 4.3: TEAM Workshop problem 22 optimization results

Approach	SQP	GA	DIRECT	B2M2
$R_1$	1.336	1.457	1.543	1.369
$A_2$	0.027	0.481	0.229	0.054
$h_1/2$	1.011	1.209	0.951	0.888
$h_2/2$	1.452	1.800	1.526	1.394
$d_1$	0.677	0.347	0.374	0.791
$d_2$	0.269	0.121	0.217	0.203
$J_1$	15.579	19.834	22.346	14.099
$J_2$	-15.069	-17.305	-13.441	-18.273
$OF$	<b>0.00197</b>	0.03502	0.04881	0.00510
# FEM evals	1902	160201	421995	106308
CR	1 %	2 %	100 %	100 %
Expected # FEM evals	529640	94276389	421995	<b>106308</b>

We notice that SQP is largely outperforming the other approaches in term solution quality, actually, 17% of the solutions found by this approach has an objective

function value less than 0.05 while being distinct in term of the variables, this explains the low CR of the method and, thus, the high expected number of evaluations.

The genetic algorithm performed better than DIRECT in terms of quality of the solution, but the former somewhat suffers from its inherent randomness and thus the low convergence rate.

B2M2 approach was able to get a solution of better quality than GA and DIRECT but less than SQP.

### Discussion

TEAM Workshop Problem 22 is one of the most treated benchmarks for the optimization of electromagnetic devices. In the literature, researchers have reported their results in many papers. However, due to the differences in the FE solver used by each researcher, the comparison to their results is not possible.

It is worth noting that some results from literature were taken into account and compared to ours [25]. Nevertheless, those from literature, when evaluated in our FE model, have bigger values of the objective function .

Another characteristic that can be deduced from this problem is the multi-modality. The low convergence rate of SQP highlights this fact, indeed, the solutions found tend to have similar values of the objective function while having design variables that are entirely different.

The SQP assisted the adjoint variable method was able to get the best solution. However, due to multi-modality, the cost of obtaining this solution is high when compared to the one found by B2M2, which is of inferior quality and with a lower cost.

Two facts can explain GA and DIRECT failure; first, the multi-modal behaviour of the problem treated, which rendered locating the best solution very delicate, and secondly, the constraint handling strategy. Indeed, the penalty approach used may not be best-suited, and the penalty factors may not be very adequate. Thus, the tuning of these factors will induce additional cost.

### 4.3.3 TEAM Workshop Problem 25

The results are summarized in TABLE 4.4.

GA could not get reliable results; only one optimization converged to the solution shown in the table. Furthermore, this one is not competitive compared to other solutions. SQP and DIRECT algorithms gave the best results, while B2M2 outperforms all the others in terms of computational cost.

### Discussion

TEAM Workshop Problem 25 is having a convergence rate (CR) of is 34% when

Table 4.4: TEAM Workshop problem 25 optimization results

Approach	SQP	GA	DIRECT	B2M2
R1	7.31	7.51	7.31	7.28
L2	14.21	14.64	14.20	14.17
L3	14.11	14.39	14.08	14.06
L4	14.37	14.44	14.39	14.34
W	<b>7.62e-5</b>	12.44e-5	<b>7.61e-5</b>	9.73e-5
# FEM evals	280	10100	24255	2575
CR	34 %	2 %	100 %	100 %
Expected # FEM evals	3585	2146501	24255	<b>2575</b>

using the SQP method. This suggests that the optimization problem is less multi-modal than the TEAM Workshop Problem 22.

In terms of solution quality, SQP and DIRECT methods lead to the best solutions (smaller value of  $W$ ), while SQP has a cost of almost seven times less than DIRECT.

B2M2 approach led to a competitive solution, even though the value of the objective function is a little bit higher. However, the design variables are very close to the ones found by SQP (difference of less than 1% of the design range). Most importantly, B2M2 cost is the most effective.

## 4.4 Chapter Summary

This Chapter presents a comparison between intrusive and non-intrusive approaches for the optimization of electromagnetic devices using FEM. We treated two well-known benchmarks from the literature [1] [2]. Then we used two metrics for the comparison, the first one is the quality of the solution, and the second is the computational cost.

The choice of the algorithm from each category is based on what we can generally find in the literature, the choice of a particular algorithm or implementation was based on what we are working on, *i.e.* SQP and meta-model approaches, and the availability and simplicity of usage. These are usually the challenges that designers are facing when doing optimization.

The genetic algorithm (GA) is one of the most used algorithms to deal with noisy data. For both test cases, GA did not perform well. These performances might be slightly improved by doing some parameter tuning or using other implementations.

DIRECT performed very well for the second test case, but it was ineffective for the first one; this can be explained by the constraints handling strategy, the penalty method implemented in the algorithm may not be the best, and additional tuning is required.

In terms of performances, the SQP approach outperforms other strategies for both test cases; this was possible due to the computation of the gradient using the adjoint variable method. This improves the quality of the solutions drastically, but this comes with the expense of intrusive manipulation of the FEM code.

The B2M2 approach remains a good alternative in terms of implementation and solution quality. The developed approach was able to overcome some of the very well-known issues when using meta-models for optimization.



# Chapter 5

## Claw-pole machine

This last Chapter first presents the main characteristics of the claw machine, the subject of this study. This machine differs from conventional electrical machines, in particular by the geometry of its rotor, which generates a three-dimensional magnetic flux and by its mechanical structure as the stator is enclosed in two brackets.

In this Chapter, we will focus on another aspect of the thesis related to the reliability of an electrical machine but closely related to the other aspects. The manufacturing processes, due to their imperfections, lead to dispersion in the material properties and the dimensions of the products. We focus on the impact of imperfections on the performance of the machine. The study presented here enables to characterize the imperfections and to conduct, for example, a robust optimization using the tools proposed in the previous Chapters.

The dispersion related to the manufacturing process induce "variability" on quantities of interest, which must be chosen among the preponderant performance factors for the product. For example, for a claw-pole machine (automotive alternator), the influence of the dispersion on the output current or acoustic response must be studied. The aim is obviously to ensure that the product complies with the specifications that are contractual between the supplier and the customer. Claw-pole machines are commonly used as automotive generators due to their simplicity and low manufacturing cost. This machine is produced in mass. Deviations on the dimensions of the machines parts and also on the parts relative position versus the nominal values can occur due to the imperfections of the manufacturing and assembling processes. Moreover, these deviations are also subject to dispersions in mass production due to the variability of the processes with time.

The classical approach based on a deterministic model (the model inputs are entirely known) to address this problem is no longer sufficient. It is necessary to consider the input quantities as uncertain quantities and to place oneself within the conceptual framework of uncertainties [46]. The approach to uncertainties introduced by Taguchi and extended to various uncertainty propagation methods (as shown in Chapter 1) is today commonly used in different engineering domains.



In this Chapter, we present the claw-pole machine and its operating mode then introduce the metrology procedure adopted for the measurement of the electrical machine. These measurements are analyzed, and the variability of the manufacturing process is characterized. Afterwards, a parametrized finite element model of the machine is presented. Next, the uncertainty propagation using the variability characterized and the model of the machine is conducted employing Monte Carlo simulation assisted by a Kriging meta-model.

## 5.1 Electrical machine

The claw-pole alternator is a synchronous electrical machine with a wound rotor which aims to supply the on-board network and charge the vehicle battery. It is placed on the combustion engine's accessory side and is driven by a belt connected to the crankshaft pulley, as shown in Figure 5.1. The ratio between the rotation speed of the alternator and that of the combustion engine is generally between 2.5 and 3.



Figure 5.1: Position of claw-pole alternator

### 5.1.1 Structure of the Claw-pole machine

Figure 5.2 shows the structure and the main parts of a claw alternator. As with all rotating electrical machines, there are a rotor and a stator enclosed in two brackets. The diode bridge and its heatsink, as well as the brushes and the regulator, are attached to the rear bearing. A plastic cover is fitted to protect these components.

Therefore, the alternator is a complicated assembly comprising parts whose geometries are also complex, such as the rotor. The role and features of each part are

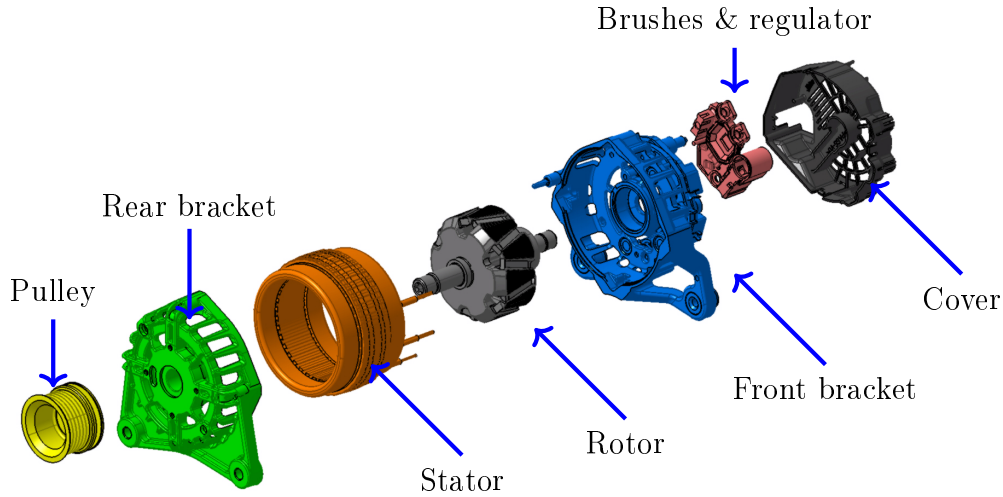


Figure 5.2: Detailed view of the claw-pole alternator

detailed in the following Sections.

### A The stator

The stator of the claw-pole machine is a helical assembly of steel sheet lamination obtained by a manufacturing process called "Slinky" (left of Figure 5.3). In our case, the sheet lamination used for manufacturing the stator is of type M800-50A. The first number in this denomination means that the iron losses are  $8 \text{ W/kg}$  at  $1.5 \text{ T}$  and  $50 \text{ Hz}$ . The second number corresponds to the thickness of the sheet in hundredths of a millimetre, *i.e.*  $0.50 \text{ mm}$ .

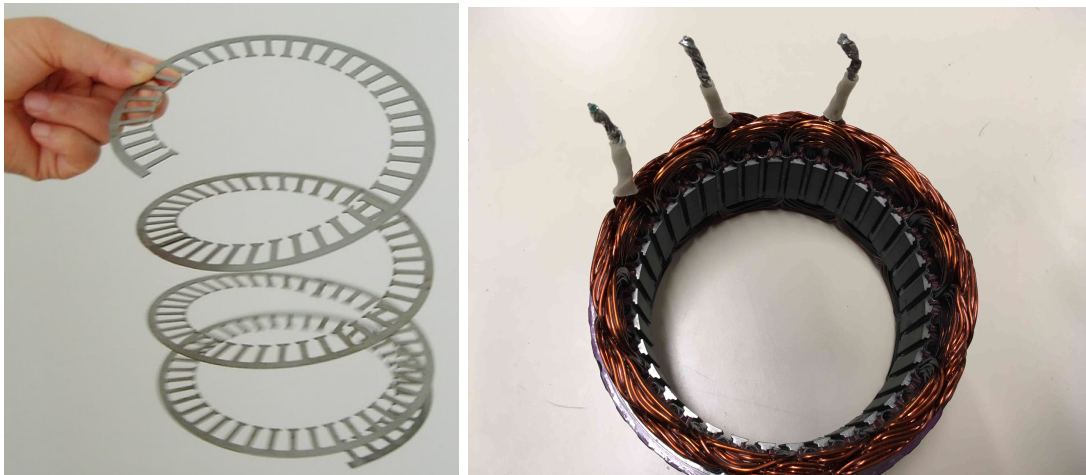


Figure 5.3: Stator of claw-pole machine

The stator winding can be single, with one slot per pole and per phase, or double with two slots per pole and per phase. The stator phases can be star or delta coupled. Delta coupling is sometimes preferred since it does not require the connection of a neutral point. However, there is a third harmonic current flowing in

the delta in relation to the electrical frequency in contrast to star connection. This current generates additional Joule losses.

After the stator winding operation, the stator is impregnated with a varnish in order to improve the mechanical strength of the winding and to improve the thermal conduction between the stator and the winding (Figure 5.3).

## B The rotor

The rotor consists in two main parts; the claws that form the machine's magnetic poles and an excitation coil. The machine contains 12 claws (6 in the front and 6 in the rear), each claw corresponds to a pole. These steel parts, of type SAE 1005, are obtained by forging. Surrounded by the claws, a coil of copper wire creates a magnetic field in the axial direction (i.e. the direction of the axis of rotation). This magnetic field is directed by ferromagnetic steel to the stator core.

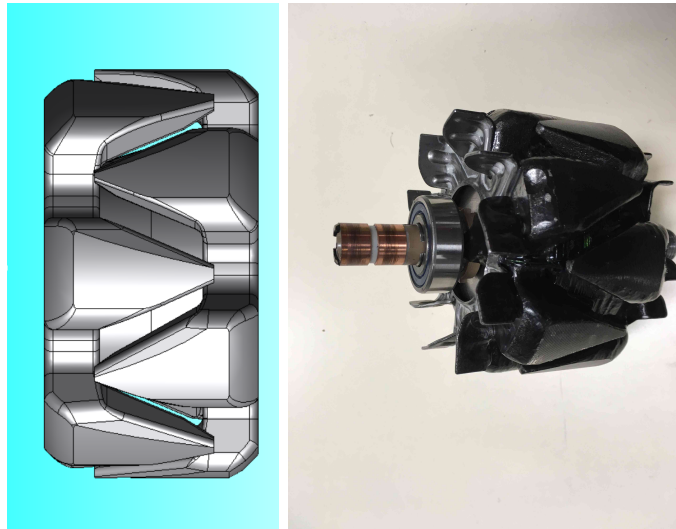


Figure 5.4: Rotor of claw-pole machine

## C The brackets

The brackets refer to the parts on each side of the stator. The geometries of the two bearings ("front" side pulley and "back" on the heatsink side) are generally different (Figure 5.5). They ensure the containment of the stator by enclosing it in the axial direction (so-called "sandwich" configuration) with the help of four screws. The brackets also enable the alternator to be mounted on the combustion engine using mounting brackets. The brackets are die-cast aluminum alloy (EN AC-46200).

## D The brushes and regulator

The excitation current of the rotor coil is supplied via two brushes (Figure 1.9), sliding on two rings placed on the shaft (see Figure 1.6). The brush holder is

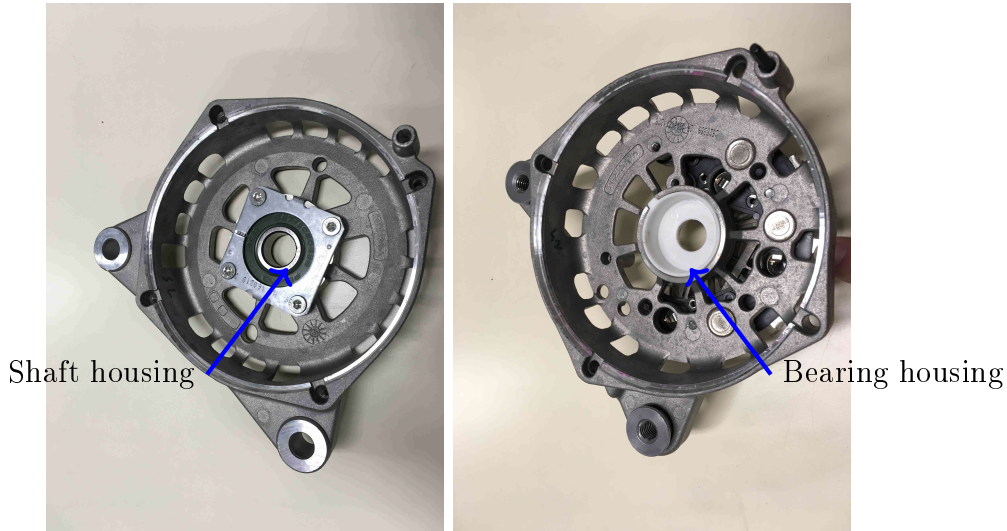


Figure 5.5: The brackets of claw-pole machine

usually located in the regulator whose role is to adapt the excitation current of the alternator in order to maintain the voltage at the battery terminals at a given level.

### E Bridge rectifier

The bridge rectifier enables to transform, using a diode bridge (Figure 5.6), the three phase AC currents ( $i_A$ ,  $i_B$ ,  $i_C$ ) to a DC current supplied to the battery.

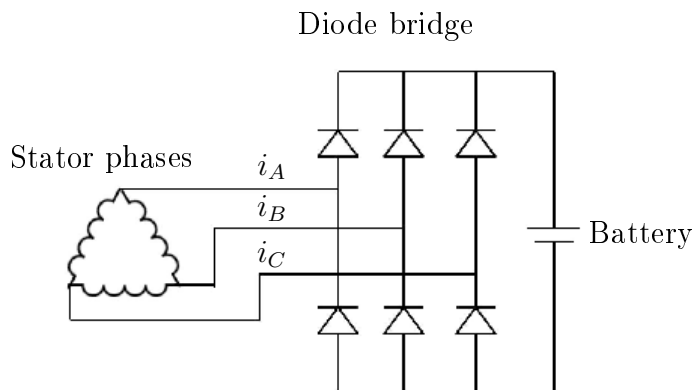


Figure 5.6: Electrical diagram of claw-pole machine

### 5.1.2 Operating mode and electrical characteristic

The rotor, to which the pulley is attached, is driven by the belt. The magnetic field is created at the rotor by the current flowing through the excitation coil. The magnetic flux then follows a three-dimensional path. When the rotor is in rotation, it induces a time-varying magnetic flux in the winding of the three phases leading to the creation of alternative three-phase ElectroMotive Force (EMF). The diode

bridge rectifies these EMF to create a DC, *i.e.* connected to the battery (see Figure 5.6)). The variation of the rectified output current (the main characteristic of the alternator) as a function of rotation speed is shown in Figure 5.7.

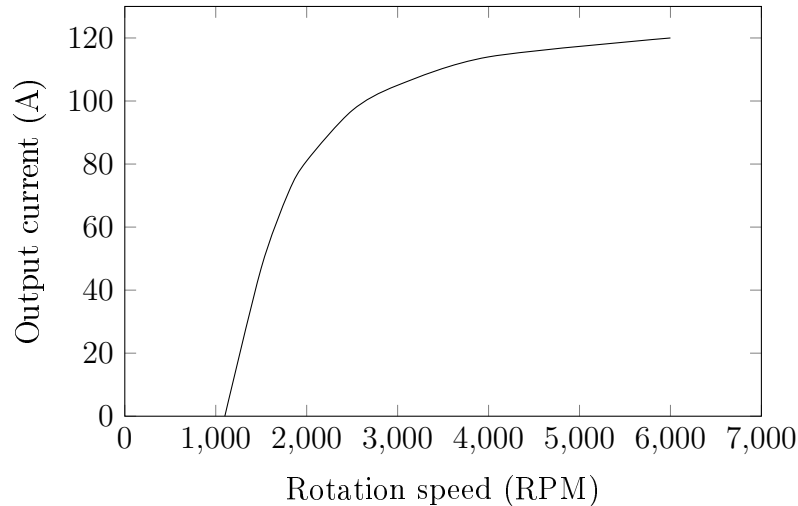


Figure 5.7: Measured output current as a function of rotation speed

It can be seen on this curve that the alternator only provides current from a certain speed ( $>1000$ RPM), which corresponds to the battery voltage exceeding the voltage at the terminals of the diode bridge.

In their specifications, car manufacturers generally require a minimum output current and efficiency for different RPM (typically 1800, 3000, 6000 RPM). Various electrical tests for alternators are defined in ISO 8854 and internal Valeo specifications.

### 5.1.3 Variability of output current in claw-pole machine

The curve shown in Figure 5.7 is obtained by measuring a machine issued from the production line. For our study, we collected three batches; each batch containing ten machines, and then we measured the outputs current. Figure 5.8 shows the bounds of variation (minimum and maximum) of the measured output current of 30 alternators. The gray area represents the possible values that the output current takes for different rotation speeds.

The variability in the measured current can attain 4 A; this leads to more than 5% of deviation at low speed. This variability is explained by the imperfection of the manufacturing process. In the next Section, we discuss the impact of the manufacturing process on the electrical machines' performances.

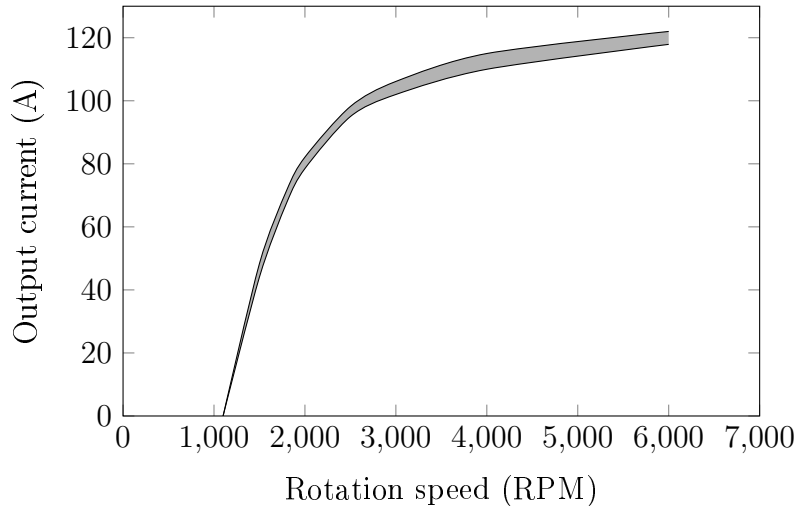


Figure 5.8: Variation of measured output current on 30 machines

#### 5.1.4 Impact of manufacturing process

There is a great deal of research into the design and optimization of electrical machines. New machine structures are emerging as well as new procedures to optimize their performance. Thus, the claw-pole machine's performance is analyzed and optimized by analytical and/or finite element methods. The influence of magnetic materials on the behaviour of the claw machine has also been studied.

However, it is noted that, in general, the manufacturability and imperfections of the electrical machine are not, or only to a limited extent, considered in most of these studies. Nevertheless, in the literature, there is some research on the impact of process imperfections and their effects, which we will summarize in the following Section.

##### A Impact on material properties

Different studies were conducted on the impact of the manufacturing process on the claw-pole machine's material properties. Some of the latest are studying the magnetic properties of the stator and the rotor [101] [102]. In [101], El Youssef et al. classify the impact of different operations in the manufacturing process (straightening, punching, rolling, ...) on the magnetic properties of the stator. They characterize qualitatively how each operation impacts the permeability and the iron losses of the stator. On the other hand, Borsenberger et al. highlighted how to measure the magnetic properties of a massive material (such as the rotor of the claw-pole machine) [102]. They revealed that the forging operation leads to disparities within the magnetic properties within the rotor.

Most manufacturing processes deteriorate the magnetic properties of materials. However, it may not always be the case; some can be beneficial, El Youssef et al. showed that compacting can improve the magnetic properties.

## B Impact on geometry

The impact of the manufacturing process on the geometry of the machine is less addressed in the literature except on some limited studies dedicated to eccentricities. The tolerances induced by mass production can lead to variabilities on the performances of the electrical machine. In [45], S. Liu has applied a statistical approach for quantifying the impact of variability on the electromotive force and the torque of the claw-pole machine. He used a parametric finite element model with an approximation method (sparse approximation) to highlight the impact of geometry variability on the performances of the machine.

### 5.1.5 Summary

Our work can be seen as a succession of S. Liu Ph.D. thesis; We are interested in the geometry and its impact on the electrical machine's performances. We aim at studying how the manufacturing process induces variability on the electrical machine by performing metrological measurement series on machines withdrawn from the production line, then, using the metrology data, to model the variability using random variables. Afterwards, these variables are used in collaboration with a parametric finite element model of the machine to perform an uncertainty propagation using an approximation method to quantify the influence of the variability of the geometry on the quantities of interest.

In the following Sections, we start with the metrology protocol to highlight how the electrical machine geometry is measured. Afterwards, we proceed by the variability modelling presentation. Then we present the parametric finite element model before explaining how we use the Kriging approximation for the uncertainty propagation to quantify the influence of the variability of the geometry on the quantities of interest.

## 5.2 Metrology

In this Section, the aim is to measure the geometry of the claw-pole machine manufactured by VALEO. Metrology is performed using a Coordinate Measuring Machine (CMM). This device measurements the geometry of physical objects by detecting discrete points on the surface of the object using a probe. Different types of sensors are used in CMMs, including mechanical, optical, laser, ... In our case, we use a ZEISS UPMC CARAT machine, which has a mechanical probe (as shown in 5.9), The uncertainties on our machines due to this CMM are less than a few micrometers.

The main objective is to metrologically determine the geometry of air-gap zone between the stator and the rotor. The air-gap plays a key role in electromagnetic behaviour; the nominal size of the air-gap is equal to  $325\mu\text{m}$ , which is inaccessible



Figure 5.9: The Coordinate Measuring Machine

to a probe. The measurements can, therefore, only be carried out on parts after the machine has been disassembled. Once measured, the parts are virtually reassembled to evaluate the air gap variation. The machines are measured at the Metrology Laboratory of the MSMP (Mechanics Surfaces and Materials Processing) at Lille, France.

The position of the probe can be controlled manually or using a computer. CMMs typically specify a probe's position as a function of its movement relative to a reference position in a three-dimensional Cartesian coordinate system (XYZ axes). In addition to move the probe along the X, Y and Z axes, many machines also allow the angle control to measure surfaces that would otherwise be inaccessible. Figure 5.10 shows how the measurements are performed; indeed, the contact point is computed mathematically from the measured point in the CMM scale and the direction of the contact force. A surface can be reconstructed from limited measured points.

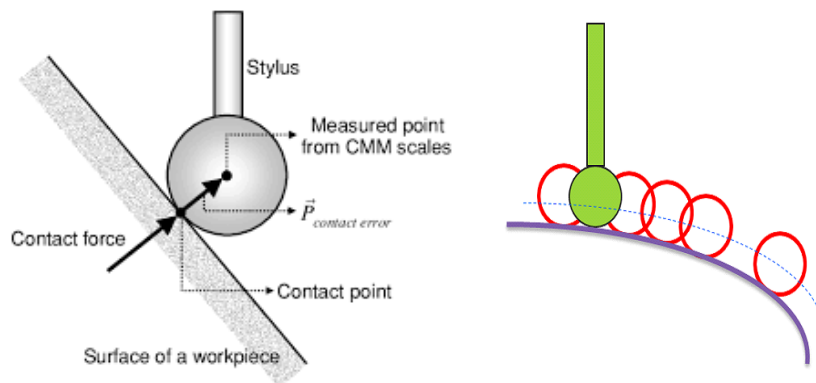


Figure 5.10: The measurement procedure using the CMM



### 5.2.1 Methodology

As noted before, the machines are disassembled to measure the air-gap. However, some referencing elements are kept to virtually reassemble the measured data and then fully characterize the air-gap. This methodology was initially developed during S. Liu thesis in collaboration T. Coorevits of the MSMP lab [103] [46] [45]. We adopted the a similar procedure for our thesis.

#### A Machine disassembling

The claw-pole machine is called a sandwich machine since the stator is maintained between the two brackets. To reduce the number of parts to be reassembled virtually, one of the brackets is glued to the stator (Figure 5.11 at left), then the pulley is removed using a tool as shown in Figure 5.11 at right. Furthermore, since the stator stiffness is low, gluing it into a bracket makes it possible (as far as possible) to preserve its shape.



Figure 5.11: Claw-pole machine disassembling

Once the pulley and the clamping screws are removed, the machine is disassembled, as shown in Figure 5.12. This leads us to three parts to be measured

- The bracket on the right denoted front bracket in the rest of this dissertation.
- The rotor and the shaft
- The bracket and the stator on the left

For measuring a workpiece using a CMM, a reference frame needs to be defined on that workpiece. In the following, we detail how these references are defined for each part and how the measurements are performed.

#### B Stator

In the stator, we are interested in measuring its internal surface to determine its deformation according to its nominal shape that is cylindrical. The stator teeth define the internal surface. However, before proceeding with this measure, some preliminary ones are done to define the workpiece's reference frame.

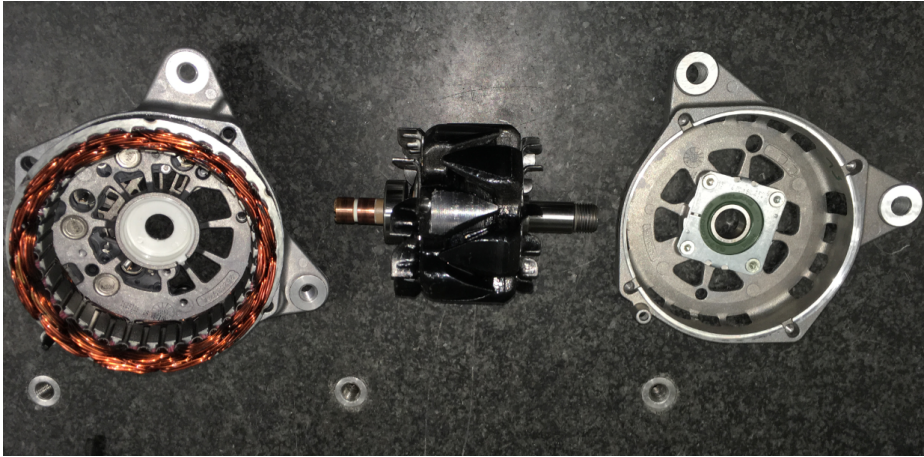


Figure 5.12: Claw-pole machine disassembled

### Reference frame

Figure 5.13 shows the reference frame adopted for the stator measurements.

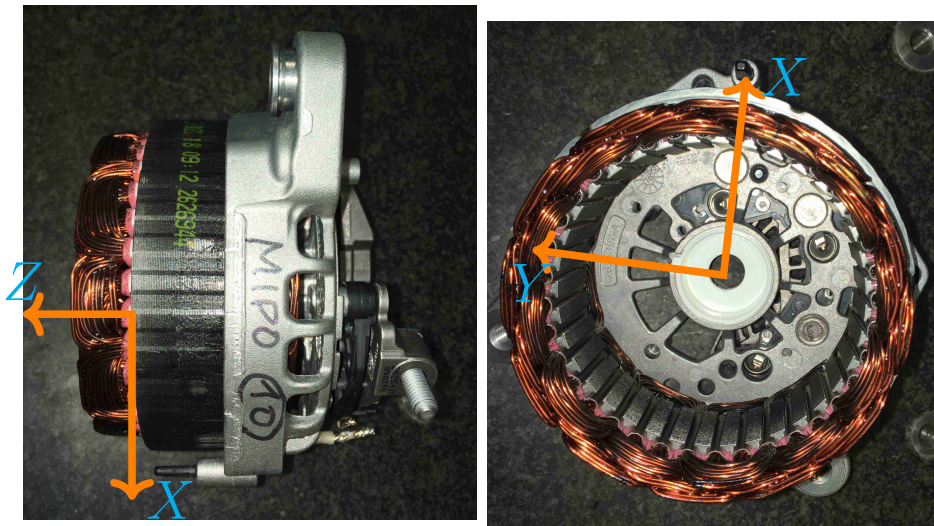


Figure 5.13: Reference frame of the stator

The reference frame is defined in the following manner

- The  $XY$ -plane is defined by using the plane on the top of the stator. Then the orthogonal vector  $Z$  is defined.
- The center of the frame is defined using some measurement in the teeth of the stator by a least square method. We use the least square method because the aim is to simulate the assembly, not to verify a conformity in accordance with a standard.
- The orientation of  $XY$  is defined by pointing the  $X$ -axis in the direction of the plastic positioning pin shown in Figure 5.13.

### Measurement process

The measurement of the inner surface of the stator is carried out on the 36 stator teeth (Figure 5.14). On each tooth, 15 points are measured, which leads to 540 points that characterize the stator's inner surface.

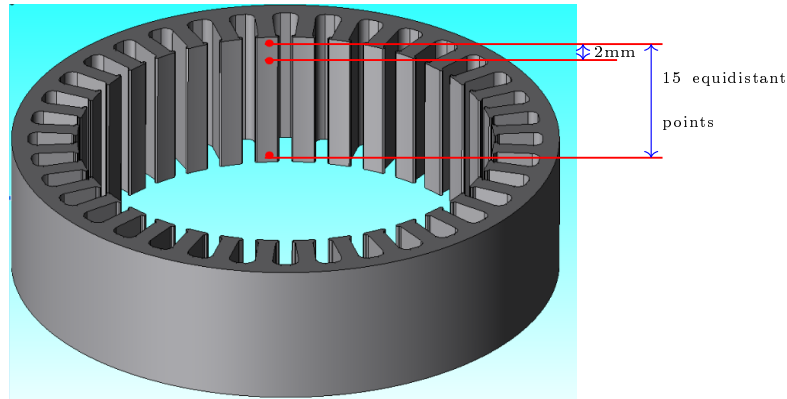


Figure 5.14: Measurement points of the inner surface of the stator

These measurement points are defined by their coordinate in the reference frame of the stator defined before.

### C Rotor

For the rotor, we want to measure its external surface to define the second surface defining the air-gap (the first one is defined by the previous measurement of the stator). The surface of the claws defines the desired surface.

### Reference frame

Figure 5.15 shows the reference frame adopted for the rotor measurement.

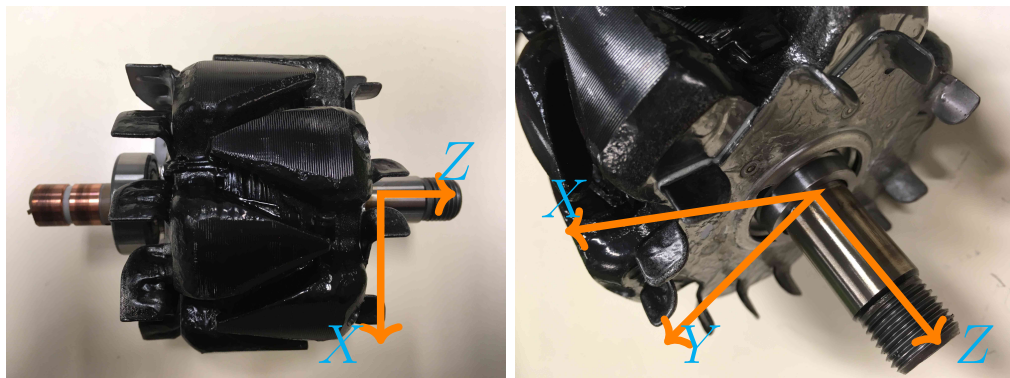


Figure 5.15: Reference frame of the rotor

The reference frame is defined in the following manner

- The  $Z$  axis is defined as the axis of the shaft and the ball bearing

- The  $XY$ -plane is defined by the surface (shoulder) retaining the axial movement of the rotor (as shown in Figure 5.15 at right).
- The orientation of  $XY$  is defined by pointing the  $X$ -axis in a direction between two claws.

### Measurement process

The measurement of the outer surface of the rotor is carried out on the 12 claws (Figure 5.16). As the claws from each side of the rotor are not identical, the number of measurement points is not the same; indeed, twelve points are performed on each claw of the upper ones and eight on the lower ones.

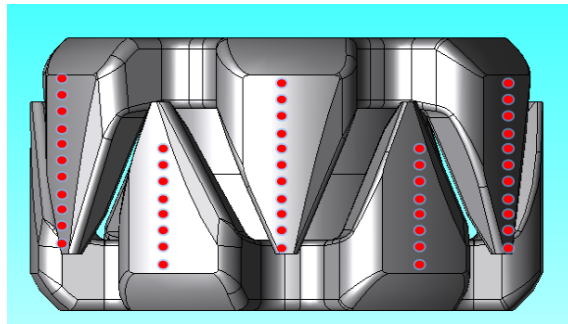


Figure 5.16: Measurement points of the outer surface of the rotor

## D Virtual assembly

We now have dealt with the measurements of the inner surface of the stator and the outer surface of the rotor. In order to be able to reconstruct the air-gap, we have to locate the stator versus the rotor. The air-gap is theoretically a zone of two coaxial cylinders, can be represented in a plane perpendicular to the axis as a curve in polar coordinates, which depends on the angular position of the rotor with respect to the stator.

First, we locate the front bracket versus the stator, then the frames of these two parts are confounded. Afterwards, the rotor is located in the frame defined on the stator.

### Between stator and brackets

The bracket on which the stator is glued is positioned on the other bracket using the interface between the two parts. The center of the pin and the bore is then used for locating (Figure 5.17).

The assembling is done virtually by confounding the reference frame defined on the bracket with the one defined on the stator.

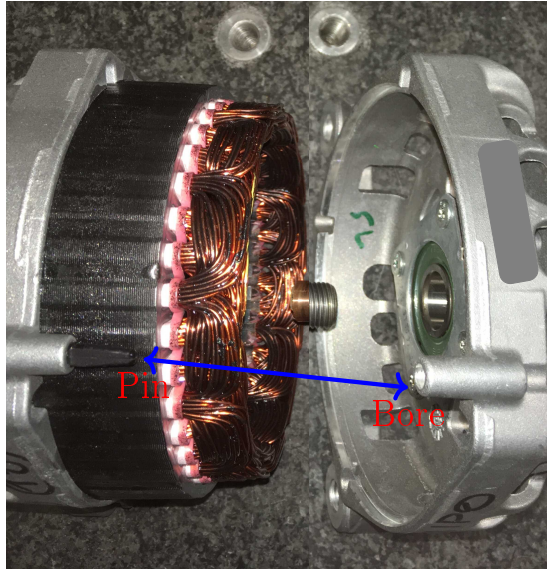


Figure 5.17: Virtual assembly of the stator and the bracket

### Between rotor and brackets

To locate the position of the rotor with respect to the stator, we measure contact surfaces of the bearings and the shaft. By these measurements, we characterize the centre of the cylindrical surfaces. Then we confound the centers to obtain a virtual assembling of the rotor and the stator.

### Assembling

When all the virtual assembling is done, we get the coordinate of the rotor axis in the stator reference frame. Figure 5.18 shows these axes, the  $z$ -direction of the rotor reference frame is confounded with the rotation axis and defined by the coordinates at the extremities of the red segment.

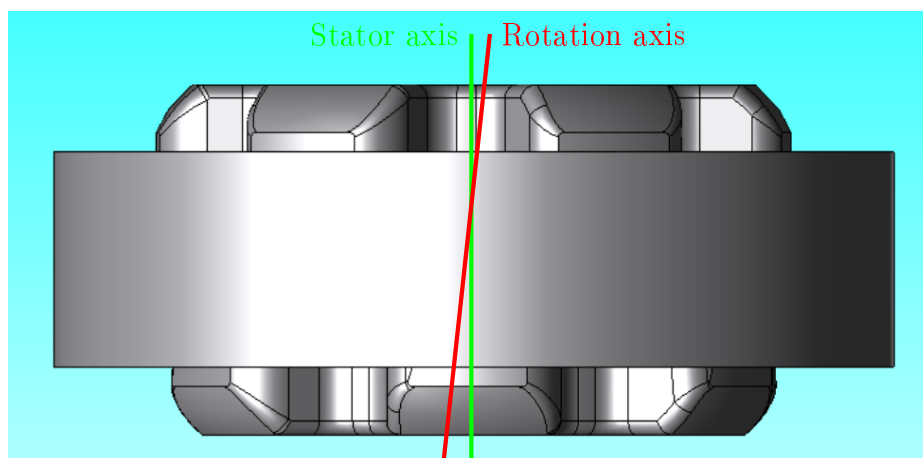


Figure 5.18: Virtual assembly of the stator and the rotor

### 5.2.2 Summary

In this Section, we presented how the geometry of the claw-pole machine is measured. Because of a lack of accessibility to the air-gap, we disassembled the electrical machines and proceeded to measure the part individually. We showed how the reference frames are defined in each part. Reference frames are essential for the virtual assembly of the disassembled machine. It allows us to assemble the machine using measurement and calculation; this enables us to characterize the air-gap size and shape, although its small size.

## 5.3 Raw data and variability modelling

In this Section, we show the measurement data of each part of the claw-pole machine. We propose a methodology to construct a stochastic model of the geometry from the measured points to exploit it in the context of uncertainty quantification.

Stochastic modelling aims to reduce the number of degrees of freedom; for example, on the internal surface of the stator, we measure 540 points. These points represent the degrees of freedom that describe the internal surface; however, all these cannot be taken into account for practical reasons (very chaotic surface) and computation (the curse of dimension). Thus, we deduce some variables representing the maximum information about the surface while being of small size.

### 5.3.1 Stator

#### A measurements

As noted before, we measure the internal surface of the stator; In Figure 5.19, we show the measured data of one of the stators. The colours highlight the deviation from the nominal radius of the stator to the measured one.

$$R_{measured} = R_{nominal} + D \quad (5.1)$$

where  $D$  is the deviation.

One can see the following

1. The deviation is not homogeneous on the surface.
2. The deviation varies between  $-50 \mu\text{m}$  and  $20 \mu\text{m}$ .
3. The mean measured radius is smaller than the nominal one.
4. The surface tends to have an elliptic shape.

These observations are relatively the same for all the measured machines but with orders of magnitude that are slightly different.



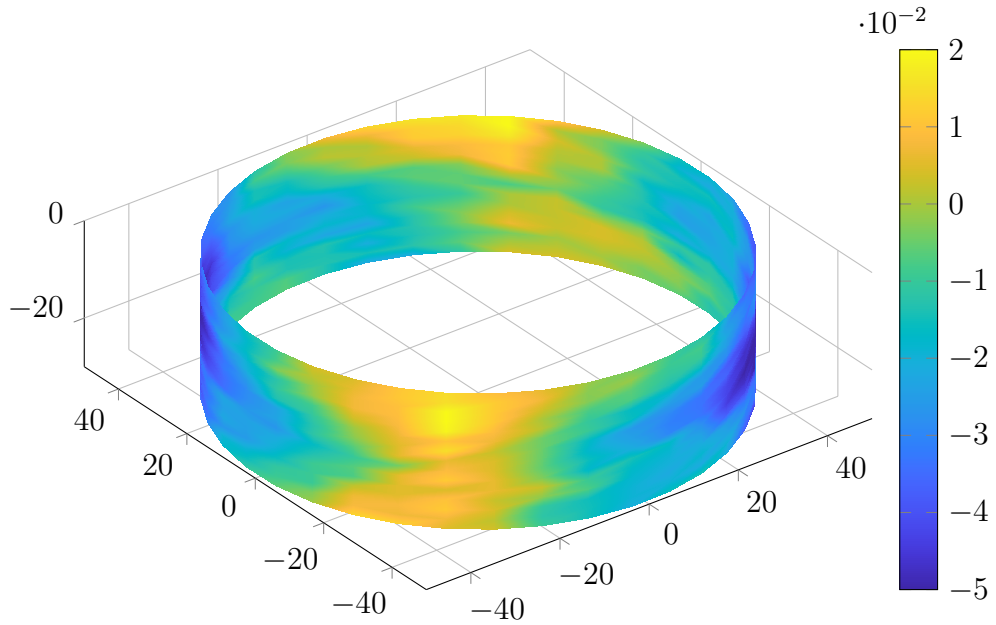


Figure 5.19: Deviation to nominal of the internal surface of the stator in  $mm$

## B Variable modelling

The purpose here is to propose a modelling technique for the shape of the stator's internal surface. As we have seen in (5.1), the measured radius of the stator is the difference between the nominal radius and the deviation  $D$ , then, to model the internal surface of the stator we need to find an appropriate model for the deviation  $D$ .

From the observations mentioned above, we can deduce a simple model for the deviation. The first part of this model is dedicated to the variation of the radius and the second part dedicated to the elliptic shape. This model is written as follows

$$D \approx a_0 + a_2 \cos(2\theta) \quad (5.2)$$

where  $a_0$  model the variability in the radius,  $\theta$  is the angular coordinate,  $a_2$  is the amplitude of the elliptic deformation.

The parameters  $a_0$  and  $a_2$  can be calculated using the least square method. The resulted shape is shown in Figure 5.20. One can visually see that this model reasonably approximates the deformation.

We adopt this approach since it reduces drastically the number of degrees of freedom to be taken into account in the numerical evaluation. These two variables ( $a_0, a_2$ ) model the shape of the stator's internal surface.

The same procedure is applied to all the machines to determine these variables for each one. Since we have made the measurement on 30 machines, we get at the end 30 values of  $a_0$  and  $a_2$ . Table 5.1 show the values of the variable for some machines and Figure 5.21 shows the statistical distribution of the values of the parameters.

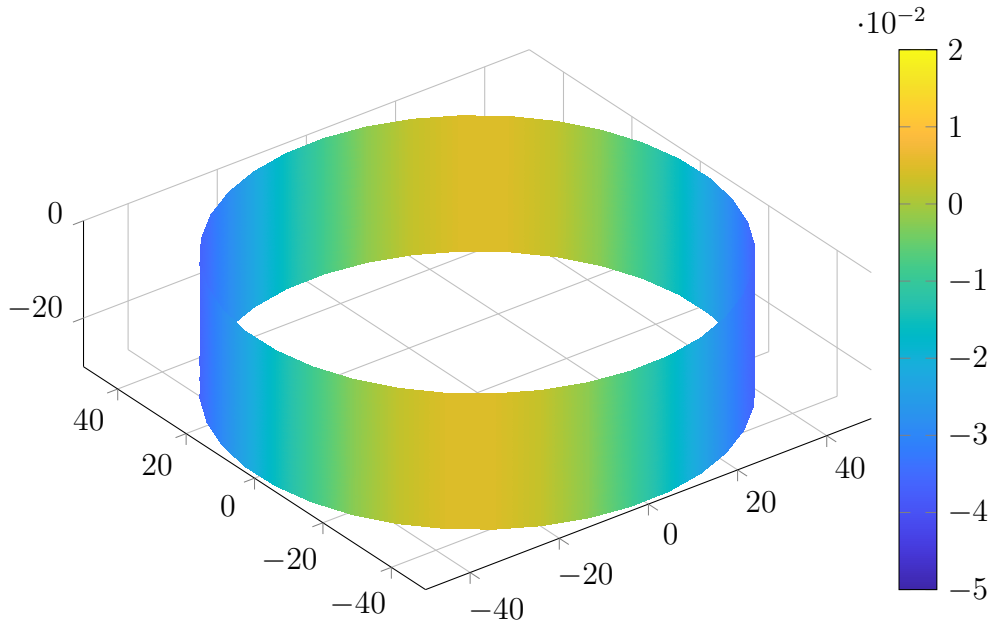


Figure 5.20: Model of deviation to nominal of the internal surface of the stator

Machine	$a_0$	$a_2$
1	-0.016	-0.021
2	-0.019	-0.013
3	-0.016	-0.021
4	-0.019	-0.012
$\vdots$	$\vdots$	$\vdots$

Table 5.1: Variables modeling the stator shape for some machines

Figure 5.21-a and 5.21-b show the histograms of the values of parameter  $a_0$  and  $a_2$  respectively while 5.21-c shows the distribution of variable  $a_2$  with respect to variable  $a_0$ . The variables  $a_0$  and  $a_2$  seem to be independent.

## 5.3.2 Rotor

### A measurements

Rotor measurements are somewhat tricky to represent graphically since they are performed on the twelve claws, six from the front of the rotor and six of its rear. Figure 5.22 shows the deviation of the radius of the external surface of the rotor to the nominal one.

In Figure 5.22-a, we highlight the measure points (triangles) by colour with the scale shown on the colour bar. In contrast, in Figure 5.22-b, we show the deviations and compared to the nominal radius and the manufacturing tolerance allowed on the external surface of the rotor.



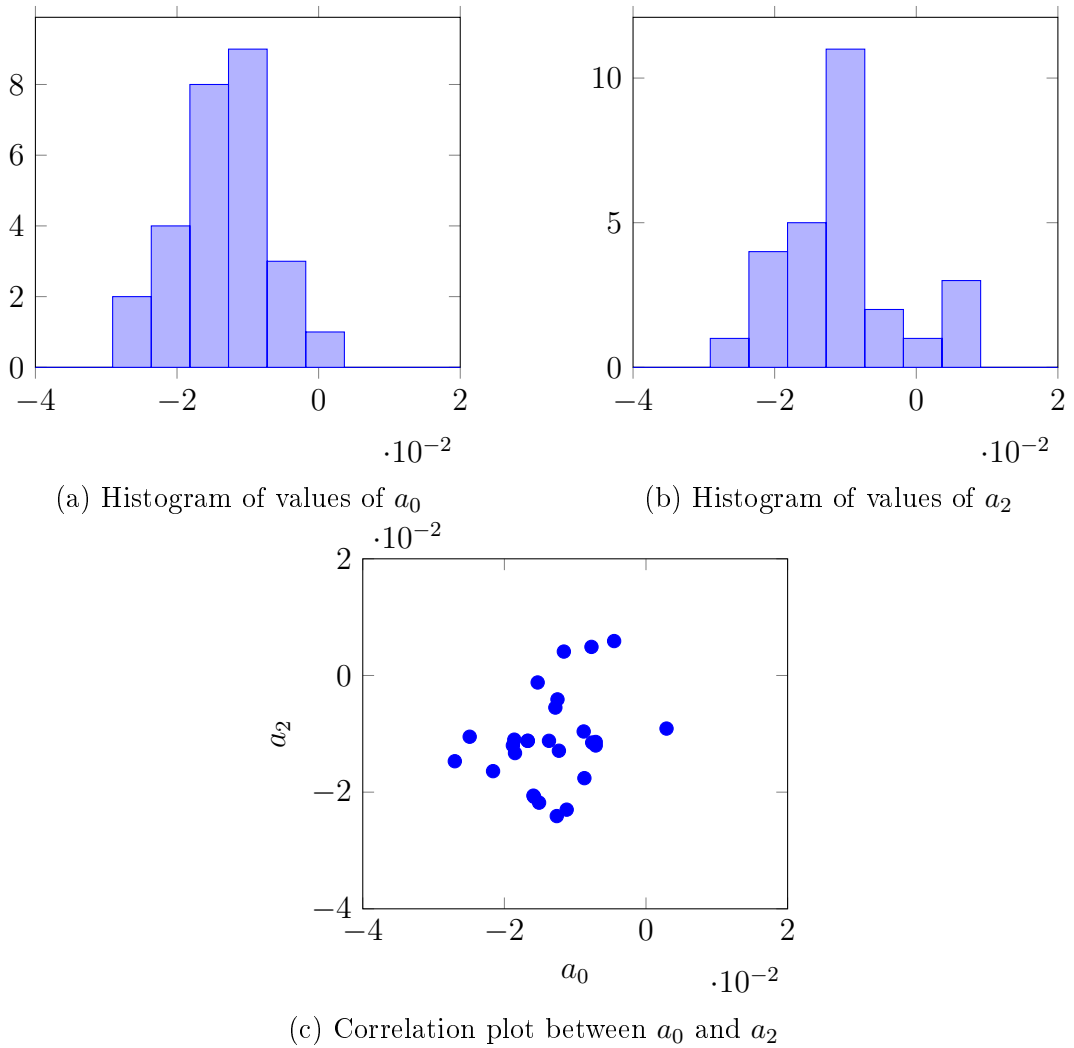


Figure 5.21: Synthesis of the measurements of the internal surface of the stator

## B Variable modelling

In Figure 5.22, one can notice two phenomena; there is variability in measurement points on the same claw; another variability is when going through claws on the same machine. Moreover, another kind of variability is related to the rotor as a whole when measuring rotors of different machines. These three sources of variability contribute to the global one.

To quantify the impact of each type on the global variability, we carry out an analysis of variance (ANOVA) study. ANOVA enables us to characterize the most significant sources of variances and, most importantly, to eliminate the sources that do not contribute much in the total variability. The presentation of ANOVA is out of the scope of this dissertation; we refer the reader to the literature for further information [104].

The factors of the ANOVA are :

1. Measurement point (the positions along the claw. If this factor is important,

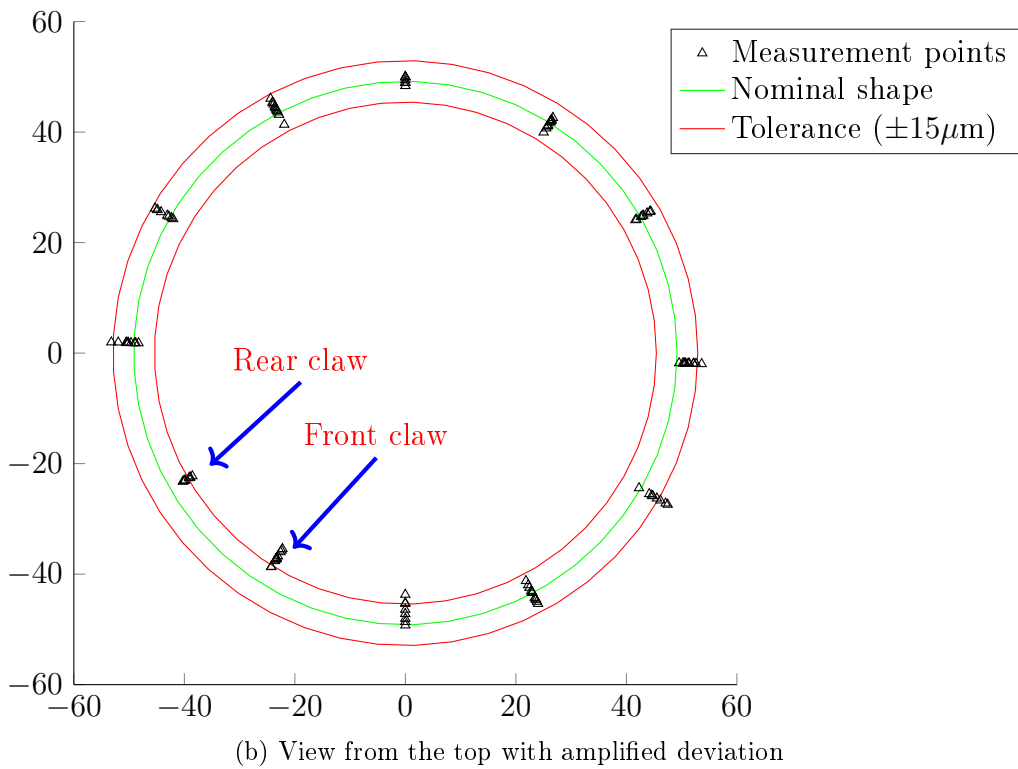
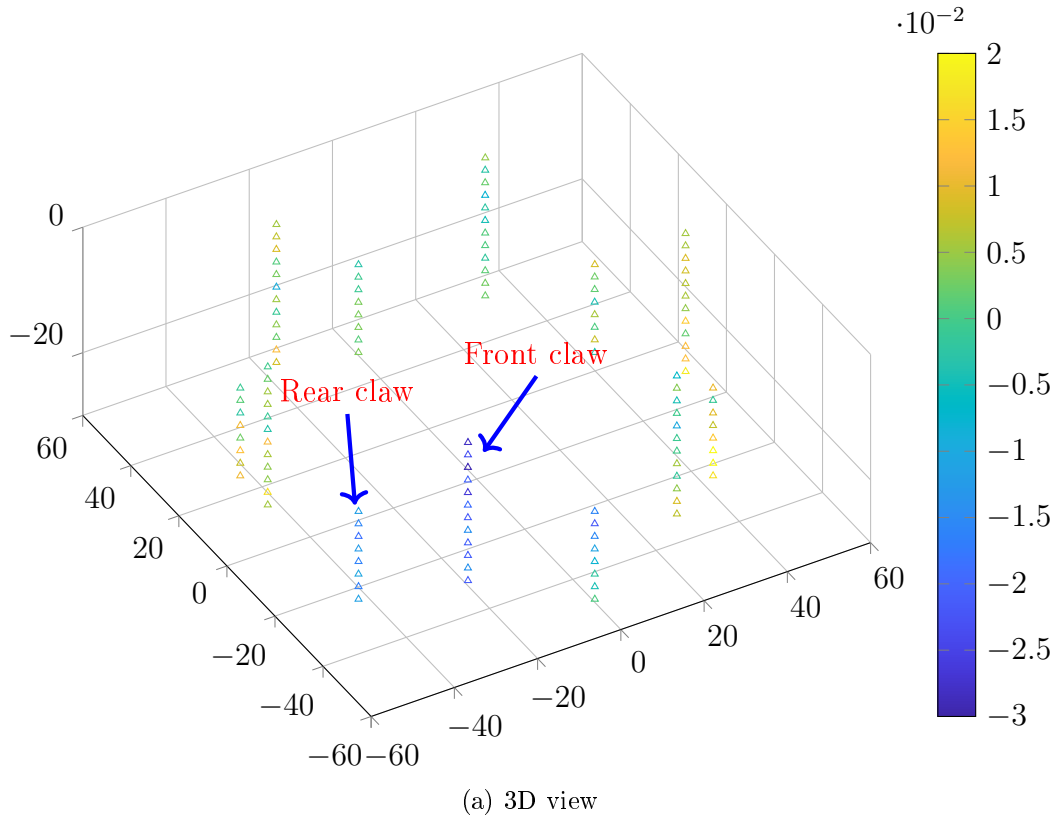


Figure 5.22: Deviation from nominal of the external surface of the rotor

it means that the variation on a claw along the axis direction is significant)

2. Claw (if this factor is important, it means that the variation of dimension

along the angle position is significant)

3. Rotor (if the factor rotor is significant, it means that the mean of rotor radius changes from a machine to another)

The results of the analysis are summarized as follows:

- Around 33% of variability is related to the factor "rotor"
- Around 27% of variability is related to the factor "claw"
- Less than 3% of variability is related to the factor "measurement point"
- The remaining is related to the interactions between factors.

From these results, we notice that the variability related to the factor "measurement point" does not contribute much in the global variability; thus, it can be eliminated without a significant impact on the analysis. It means that we do not need to develop a model accounting for position along the claw, i.e. along the  $z$ -direction.

The factor "measurement point" is responsible for the variability of measurement points on each claw independently. By eliminating this factor, we can consider that there is no variability on each claw. Using this argument, we can say that the radius of the rotor is constant for each claw but varies when going from a claw to another and from a rotor to another. From the ANOVA, we can see that the factor "rotor" and "claw" have an influence, meaning that the mean radius will change significantly from one machine to another and that the angular position also has an influence.

In other words, based on the visual observation in Figure 5.22, One can notice that the measured cylindrical surface presents some eccentricity; in fact, the axis computed using a least square method is shifted from the axis that defines the nominal cylindrical surface (see the definition of the reference frame of the rotor). Furthermore, the measured surface radius is smaller than the nominal one. Simply, we want to find the deviation, which is the difference between the measured radius and the nominal radius.

$$R_{measured} = R_{nominal} + D \quad (5.3)$$

where  $R_{measured}$  and  $R_{nominal}$  are the measured and the rotor nominal radius respectively and  $D$  is the deviation.

Based on the observations above, we can write  $D$  as follows.

$$D \approx b_0 + b_x \cos(\theta) + b_y \sin(\theta) \quad (5.4)$$

where  $b_0$  models the offset to nominal radius while  $b_x$  and  $b_y$  model the dynamic eccentricity.

We characterize these variables for the whole set of machines, and their distribution is shown in Figure 5.23.

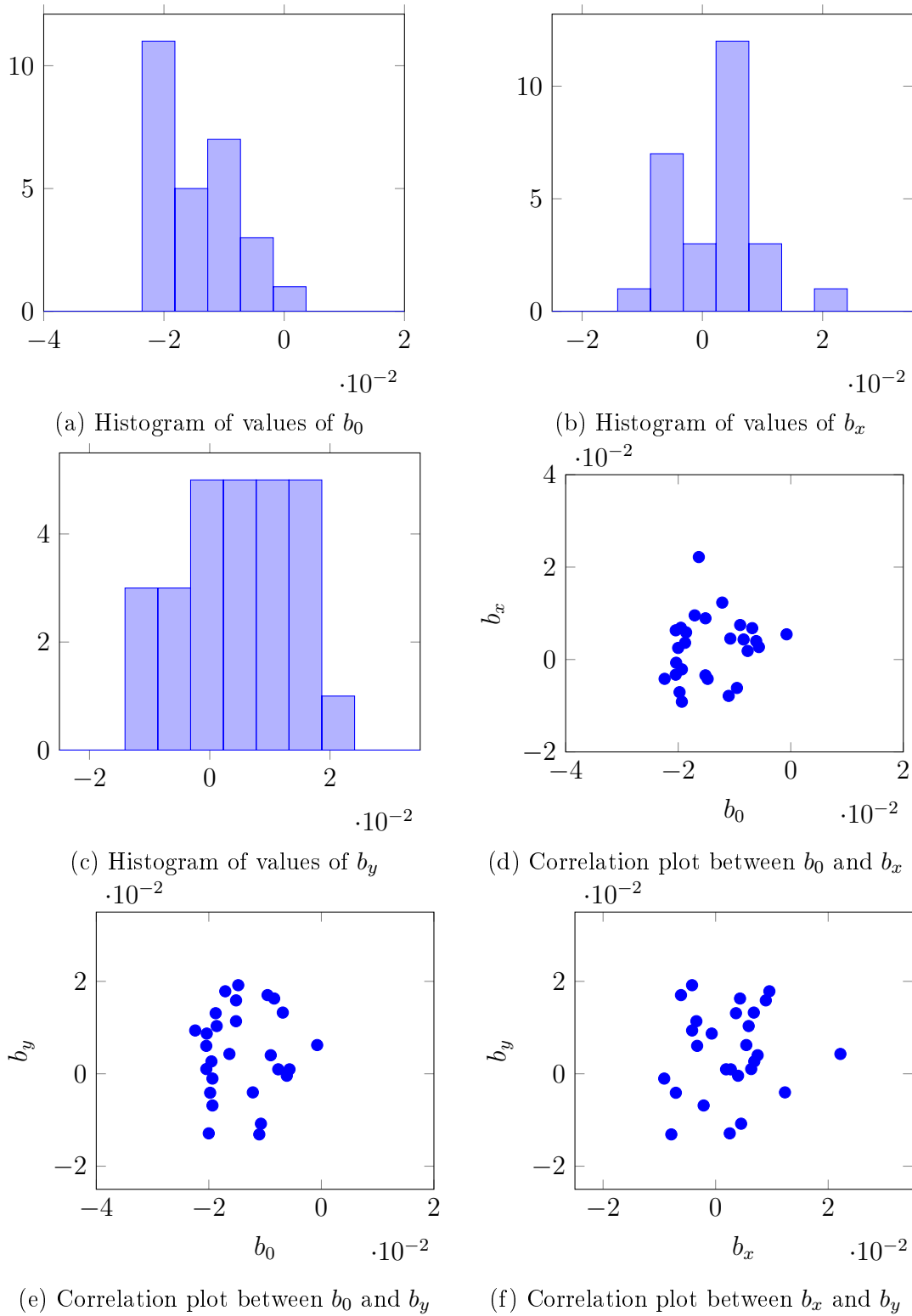


Figure 5.23: Synthesis of the measurements of the outer surface of the rotor

### 5.3.3 Virtual assembly

#### A measurements

As detailed in the previous Section, the virtual assembly leads to define the rotation axis in the reference frame of the stator, as shown in Figure 5.24.

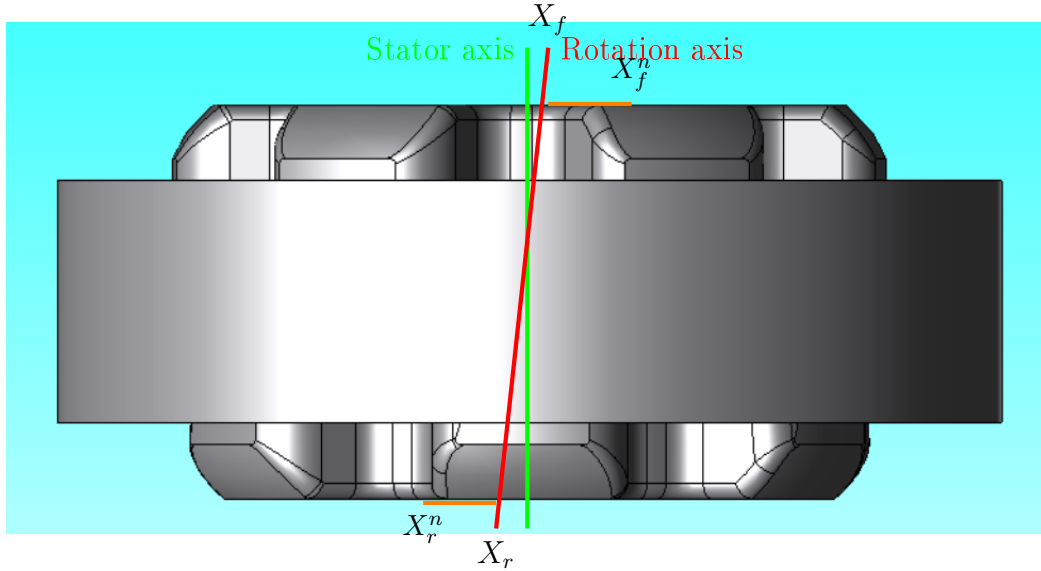


Figure 5.24: Virtual assembly of the stator and the rotor

The rotation axis is defined by the coordinate of the extremities  $X_f$  and  $X_r$ . Each extremity is defined by three coordinate  $(x, y, z)$ . Thus, we have six degrees of freedom for defining this axis. We can eliminate the  $z$  coordinate if we consider the  $x$  and  $y$  coordinates at the limit of the rotor at  $X_f^n$  and  $X_r^n$ .

#### B Variable modelling

We consider the  $x$  and  $y$  coordinates of the points addressed before as the variables to be controlled; they are defined as follows :

- $x$  and  $y$  coordinates at  $X_f^n$  are denoted  $c_x$  and  $c_y$  respectively
- $x$  and  $y$  coordinates at  $X_r^n$  are denoted  $d_x$  and  $d_y$  respectively

In Figure 5.25 and 5.26, we show the distribution of these variables.

### 5.3.4 Variability modelling

The measured data enables us to model the parameters that represent the variability of the electrical machine's shape. These parameters are calculated for each machine, and then we need to identify the probability distribution of each parameter. We

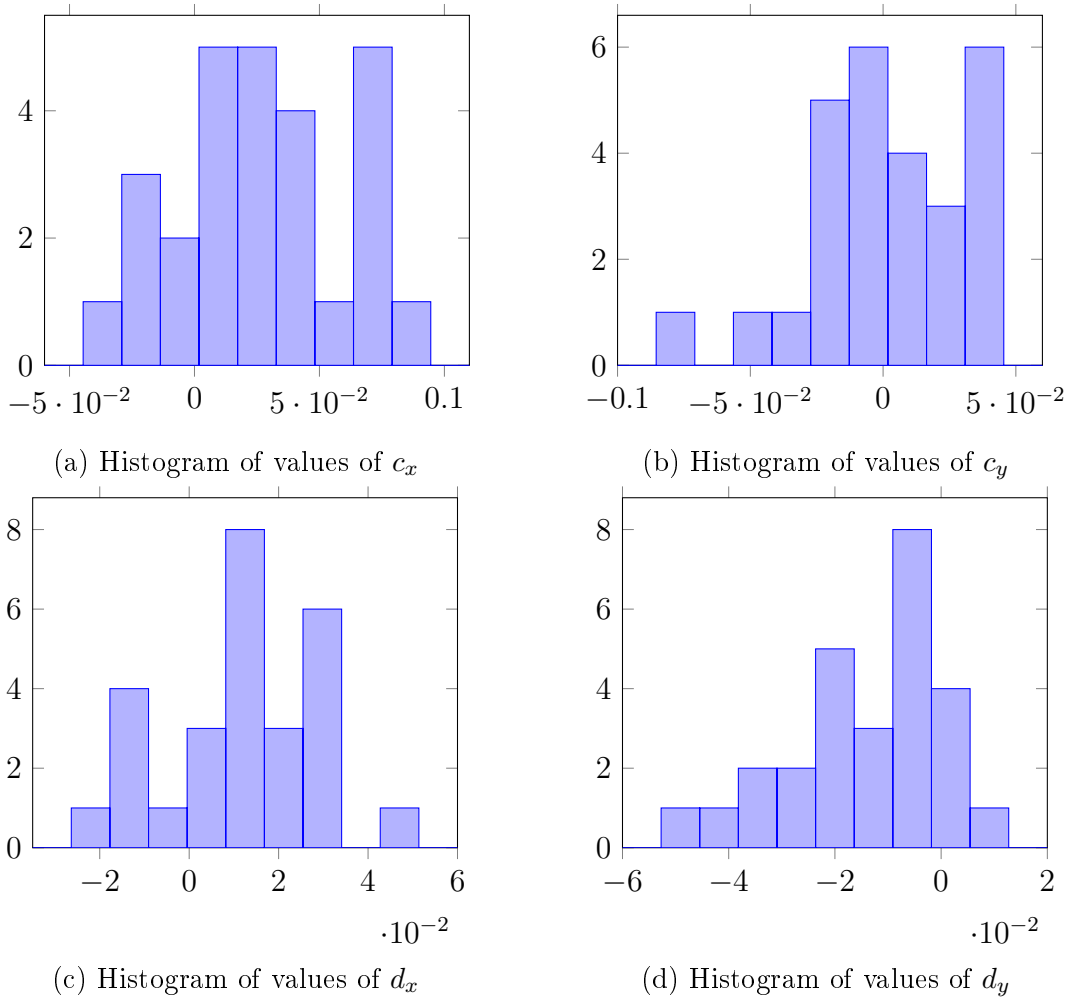


Figure 5.25: Synthesis of the measurements of of virtual assembling (Part 1)

consider the parameters as random variables with normal distribution. We have identified the best parameters (mean and standard deviation) that are the best fitted to represent the sample of each parameter. We have also carried some Normality tests in order to validate our assumption. We used the toolbox developed by M. Öner et al. to conduct ten of the most used Normality tests. As a result, at least five out of the ten tests have shown that the normality hypothesis cannot be rejected (for a  $p$ -value of 0.05) for all the parameters. Figure 5.27 shows all the variable histograms with their relative normal distributions.

### 5.3.5 Summary

In this Section, we have shown the protocol developed for measuring the electrical machines; the machines were first disassembled into three parts.

1. The stator glued to one of the brackets
2. The remaining bracket

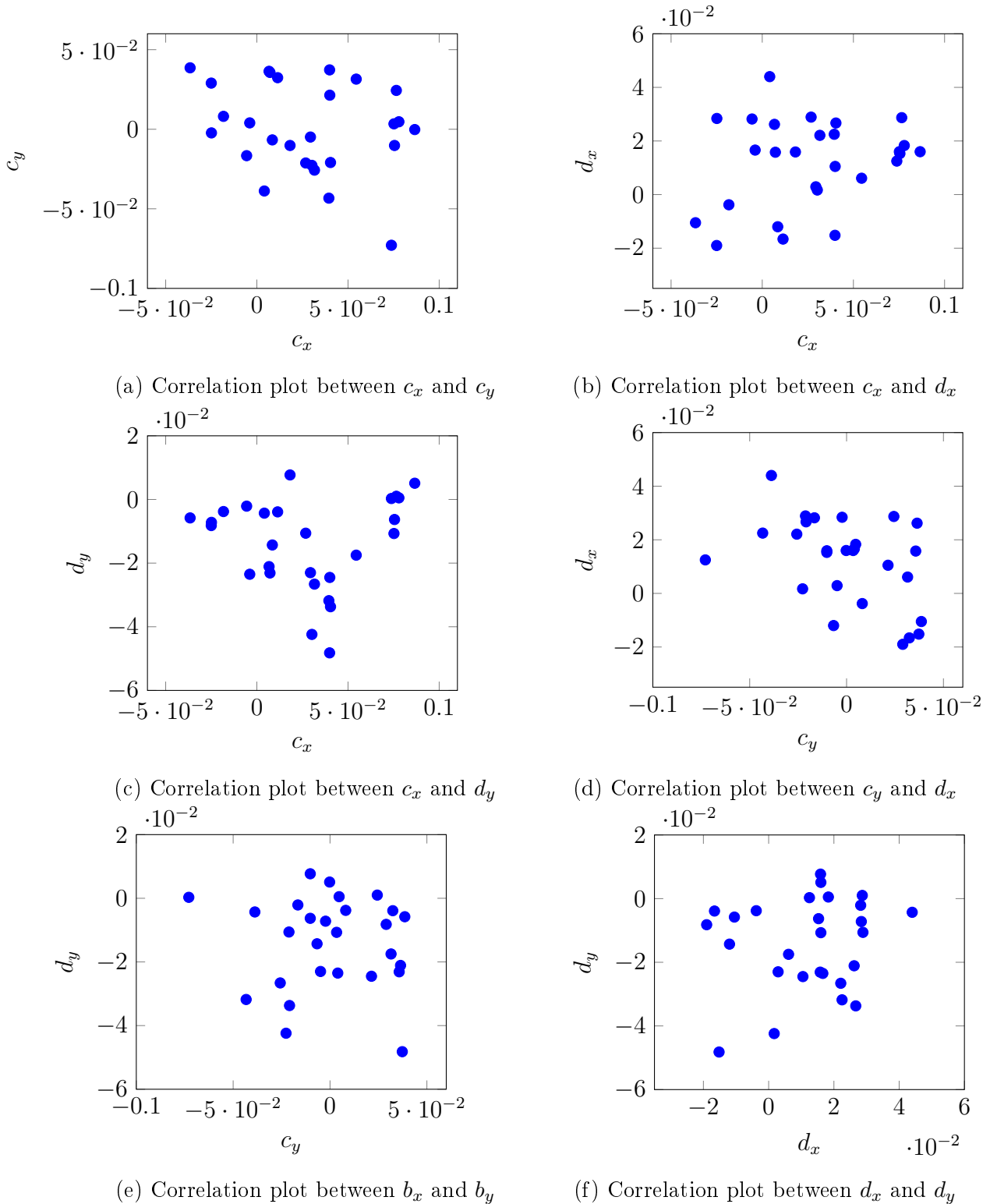


Figure 5.26: Synthesis of the measurements of of virtual assembling (Part 2)

### 3. The rotor and the shaft

Each part was measured individually to characterize the variability on the geometry of the machine. Afterward, the machine was assembled virtually using the reference frames defined on each part. The virtual assembling enabled us to locate

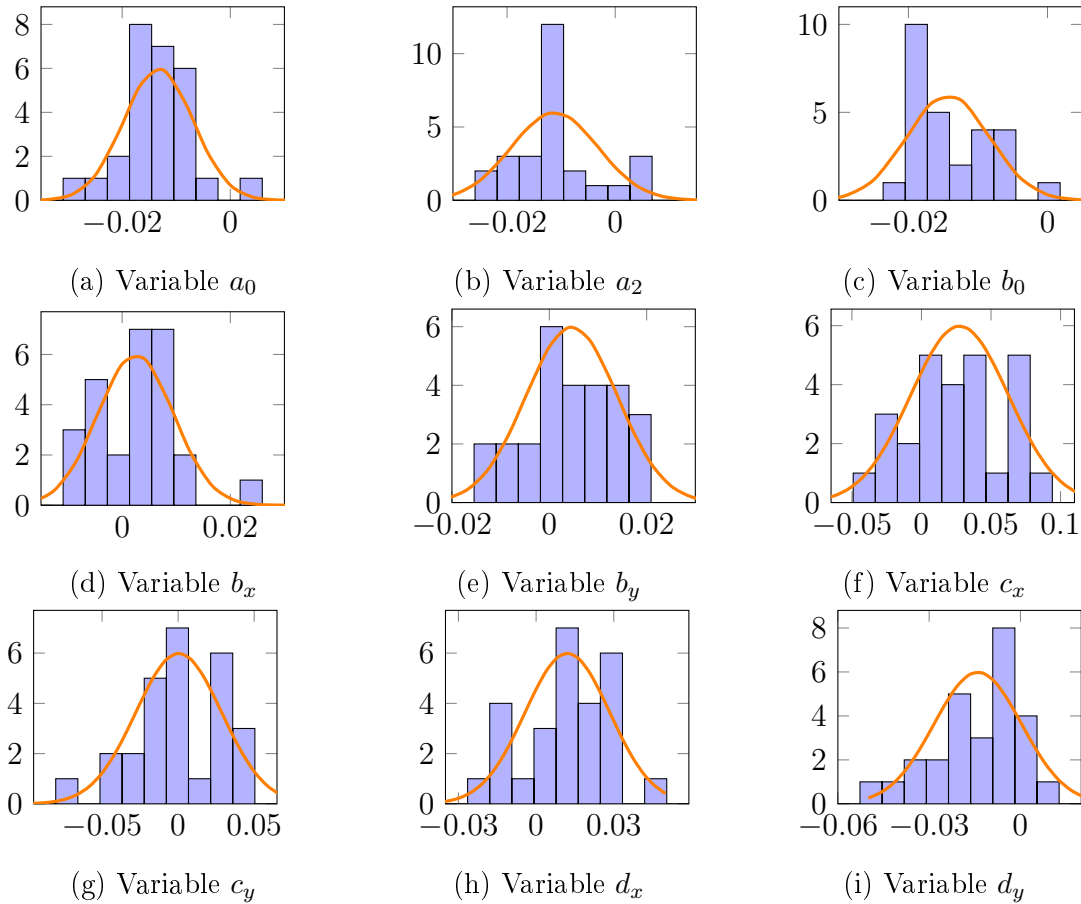


Figure 5.27: Synthesis of the all measurements on claw-pole machines

each part with respect to the others and therefore have a sophisticated protocol for measuring the air-gap although its small size ( $325 \mu\text{m}$ ).

The measurements were exploited to get intuitions of the parameters that model the variability. Each measurement point represents a degree of freedom of the geometry of the machine. Treating all the degrees of freedom is impractical and intractable due to their high number (540 in the stator, 120 in the rotor, ...). Therefore, we proposed some key parameters ( $a_0, a_2, b_0, \dots$ ) to capture the maximum information about the shape of the measured surfaces. These parameters were mainly deduced using graphical representation and some statistical methods such as ANOVA.

Once the variables were defined for all the claw-pole machines and the variability were modelled as random variables by determining their marginal probability density function (these variables are considered to be independent), the probability distributions are considered as input for the FEA model that is used for the variability (uncertainty) propagation.



## 5.4 FEA model

The modelling of the claw pole machine using finite element analysis is cumbersome, primarily because of the tridimensional geometry. The difficulty related to the creation of the model is dealt with gradually to be able to run the uncertainty propagation. The robustness of the model is a key-ingredient to the success of this task. We start by creating a parameterized model of the electrical machine with its nominal geometry, material property, electrical circuit and functioning conditions (temperature, excitation current, ...). Then, the variables introduced in the previous Section are also added to extend the capabilities of the model.

For this analysis, we use JMAG as a finite element tool for the machine's electromagnetic modelling. Thanks to its numerous advantages in terms of geometry modelling, mesh morphing, eccentricity handling, *etc*

### 5.4.1 Geometry

The geometry of the electrical machine is built based on the 2D technical drawings (as shown in Figure 5.28) describing its different parts. The idea is to have a geometry the closest to the nominal one.

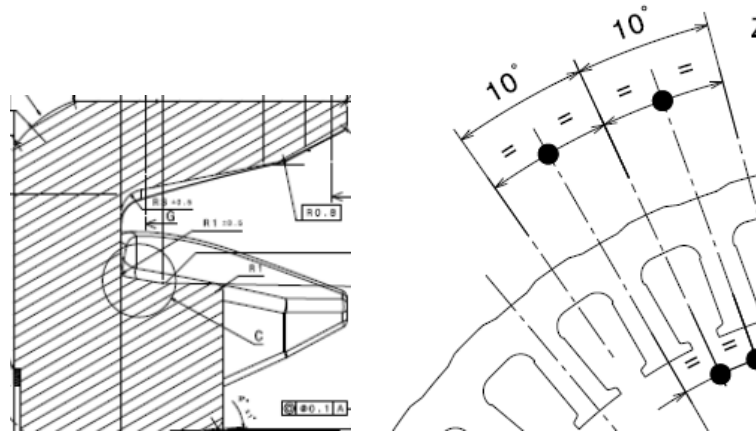


Figure 5.28: Technical drawing of the electrical machines

The geometry is then parametrized; more than 50 parameters are used to control the shape of the geometry (some parameters are shown in Figure 5.29). The parameters enable to change the shape of the machine by changing the values of these parameters.

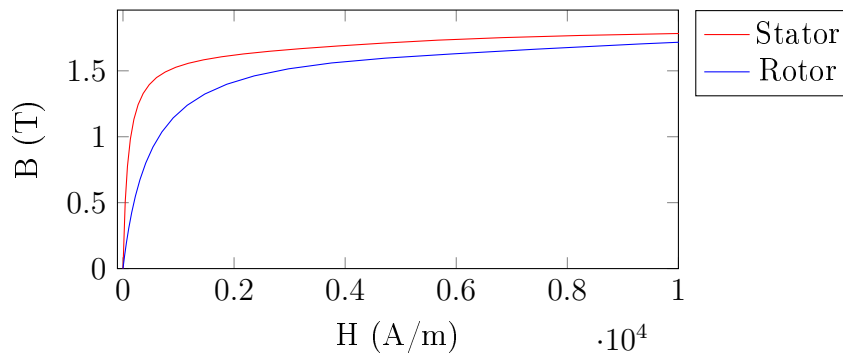
The parameterization is not a straightforward task due to the complexity of the geometry and the invalid shapes generated when varying the parameters. Thus, the robustness of the model is critical when parameterizing a geometry.

	Variable	Type	Expression
1	r_claw_inner_radius	Expression	7
2	r_claw_length	Expression	2
3	r_plate_length	Expression	2
4	r_core_length	Expression	1
5	r_core_radius	Expression	5
6	r_shaft_radius	Expression	1
7	r_claw_width	Expression	2
8	r_plate_filet_radius	Expression	1
9	s_outer_radius	Expression	1
10	s_inner_radius	Expression	9

Figure 5.29: Some parameters of the geometry

### 5.4.2 Material properties

The materials used in electrical machines are specified by their  $BH$ -curves. The stator and the rotor are of different materials, thus the curves are different, as shown in Figure 5.30. The stacked lamination in the stator leads to a better performance than the massive rotor.

Figure 5.30:  $B(H)$  curves of the machine materials

### 5.4.3 Electrical circuit

The terminals of the stator winding are connected to a rectifier, which is itself connected to a battery. In Figure 5.31, we show the component of the circuit; the stator winding, the diode bridge, and the battery. We add a current probe to measure the electrical current transferred to the battery.

### 5.4.4 Model validity

Before using our model for uncertainty quantification, we assess a model validity step to compare the simulation result with respect to the experimental measurement on

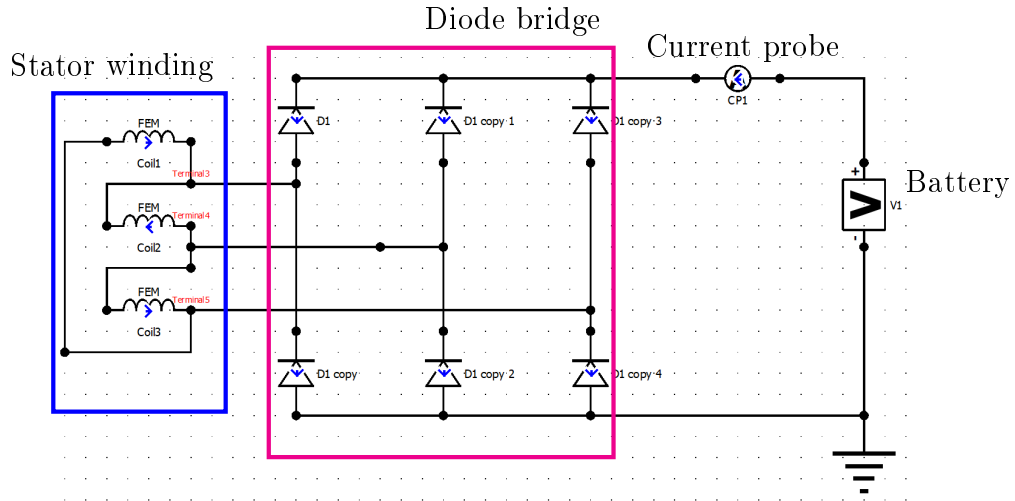


Figure 5.31: Electrical circuit of the claw-pole machine.

the output current. For this purpose, we choose three functioning points at 1800 RPM, 3000 RPM and 6000 RPM.

For experimental results ( $I_{\text{exper.}}$  : measured output current), the values taken are the means of the measured output currents for all machines. These current are compared to the ones computed using simulation ( $I_{\text{simul.}}$  : simulated output current). The comparison is summarized in Table 5.2.

RPM	$I_{\text{exper.}}$ (A)	$I_{\text{simul.}}$ (A)
1800	69.5	70.1
3000	104.7	106.0
6000	120.4	120.1

Table 5.2: Comparison of simulation result with respect to experimental one

We can notice that the simulation results are in good agreement with the mean of the experimental measurements with a relative error of less than 1.3 %. It worth noting that the variability on the measured machines is around 4A for all the functioning points.

### 5.4.5 Variability parameters

In the following we aim at quantifying the effect of the random input parameter defined in the previous Section on the current. We proceed to take into account the variability that was identified using the metrology. We present how each of the parameters is taken into account using the developed model of the claw pole machine.

**Variables  $a_0$  and  $a_2$** 

The parameters  $a_0$  and  $a_2$  are used to model the interior surface of the stator; thus, these variables used to deform the mesh of the teeth of the stator. On each tooth, we apply a radial deformation (as shown in red for one tooth in Figure 5.32) of amplitude defined by

$$D_i = a_0 + a_2 \cos(2\theta_i) , \quad i = 1, \dots, 36 \quad (5.5)$$

with  $i$  is the number of the tooth and  $\theta_i = 10i$

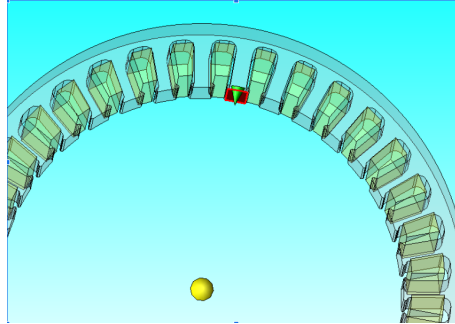


Figure 5.32: Radial deformation of stator teeth

**Variable  $b_0$** 

The variable  $b_0$  models the variation of the radius of the rotor. Thus, as the rotor radius is already defined in the initial parameterization of the machine geometry, no mesh deformation is needed in this case (as for the stator teeth).

**Variables  $b_x, b_y, c_x, c_y, d_x$  and  $d_y$** 

The variables  $b_x$  and  $b_y$  are used to represent the rotation axis eccentricity (dynamic eccentricity) while the variables  $c_x, c_y, d_x$  and  $d_y$  model the static eccentricity. JMAG is equipped with an eccentricity module (shown in Figure 5.33) that enables us to take into account these variables at once. The parameterization of the eccentricity on JMAG is somewhat different from the variables we use. Nevertheless, the transformation that links both of them is not very difficult (geometrical transformations such as translations and rotations).

**5.4.6 Summary**

In this Section, we presented the developed finite element model of the electrical machine with all the aspects of the geometry, the material properties and the electrical circuit. We verified the model validity with respect to experimental measurements. We have shown how the parameters determined from the analysis of the metrology were taken into account in the numerical model. Due to the complexity of the deformations involved, such as eccentricity, the symmetries of the machine cannot

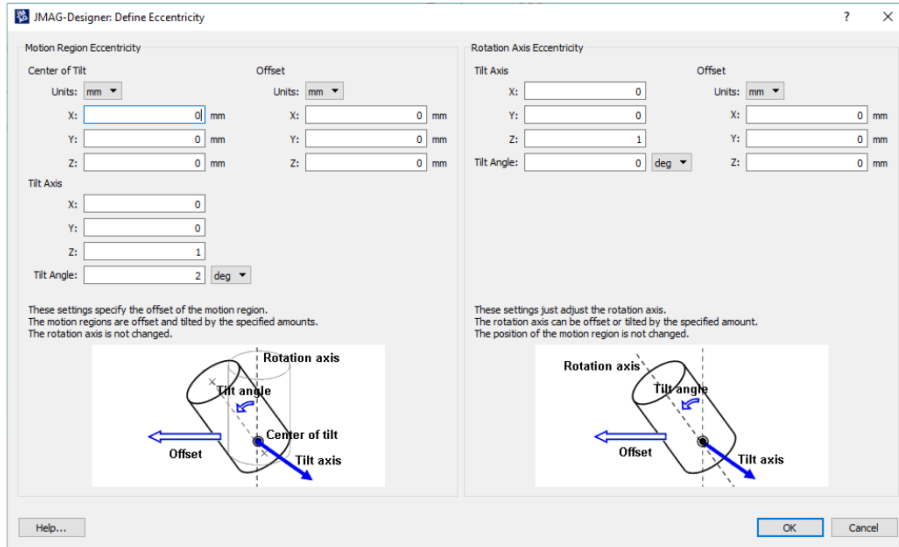


Figure 5.33: Eccentricity module of JMAG software

be exploited; therefore, the whole machine needs to be modelled. Therefore, this increases the evaluation time of the FEA, which is close to 10 hours. For this dissertation, we choose only a operating point at 4000 RPMs. The point is chosen because it presents the largest variability in the experimental data.

## 5.5 Uncertainty propagation

The uncertainty propagation aims to propagate the variability that we have characterized on the shape of the electrical machine and then compute the variability on the output current. For this purpose, we use a meta-model approach. The meta-model is constructed based on a small sample from the computer expensive FEA then the meta-model is used for statistical inferencing.

### 5.5.1 Sampling

The sampling aims at evaluating the model of the electrical machine at some sample points that represent the distribution of the variables that have been characterized. As we considered the variables to be normally distributed, we use an LHS method adapted to normal distributions [105], also implemented in MATLAB as *lhsnormal*. We sample 30 points and ran the simulations for all the points. As each simulation take around 10 hours to complete, all the simulations were performed in around 13 days on a Windows core i7, 4 cores of 2.71Ghz.

### 5.5.2 Meta-modelling

After the evaluation of the FEA, we proceed to fit a meta-model on the computed data; the Kriging model is used for this purpose.

### 5.5.3 Statistical inference

As the Kriging model is cheap to evaluate, we deem tractable to use a Monte Carlo simulation for the uncertainty propagation. We generate a sample of considerable size (1 million points) then we evaluate the Kriging model to get a corresponding value of the output current of each evaluated point. The resulted distribution is shown in the histogram of Figure 5.34. One can see that the current can vary between 114.5A and 117.5A. This distribution is similar to a normal distribution shown as the orange curve of Figure 5.34.

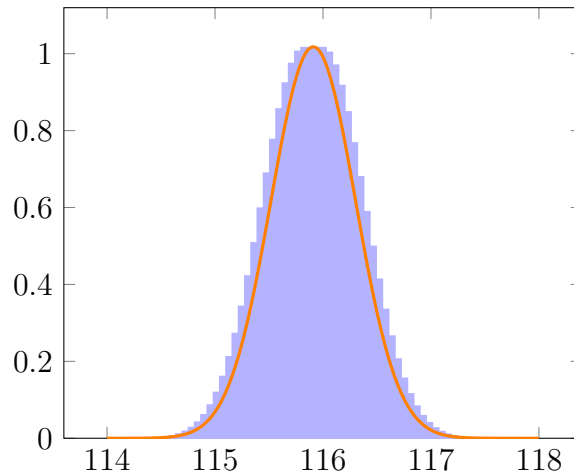


Figure 5.34: Variability of the output current resulted from variability on geometry

This variability needs to be reduced; therefore, identifying the variables that led to this variability is of high interest. In the following, we present a sensitivity analysis for this task.

### 5.5.4 Sensitivity analysis

In order to determine which input parameters influence the most the variability of an output parameter, a sensitivity analysis is carried out. It involves estimating sensitivity indices that quantify the influence of one input or group of inputs on the output. In our case, we want to study the impact of the variables that represent the deformation of the geometry on the output current of the machine. There exist many methods for sensitivity analysis [106]. One of the most known is the Sobol indices, which based on the decomposition of variance.

We compute Sobol indices using a Monte Carlo simulation by evaluating the

previously fitted Kriging model. The results are summarized in Figure 5.35. We notice that more than 90% of the variance is related to only two variables ( $a_0$  and  $b_0$ ) while the other variables do not contribute much in the variability of the output current.

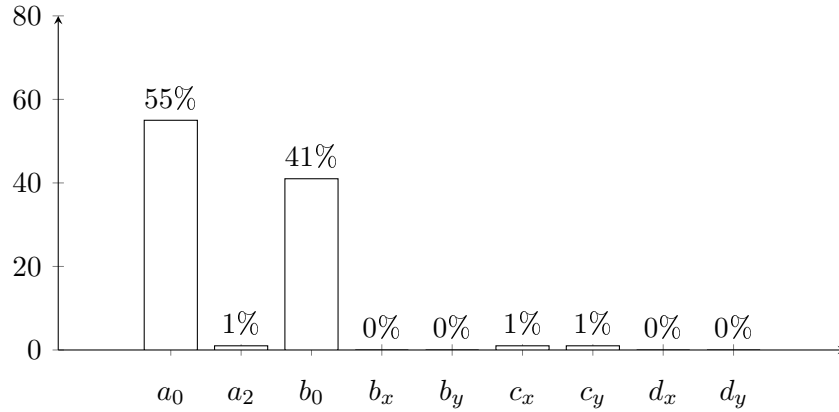


Figure 5.35: Sobol indices of the variables

If we want to reduce the variability on the output current, it will have almost no impact if we reduce the variability of other parameters than  $a_0$  and  $b_0$ . The variable  $a_0$  models the deviation of the stator radius from the nominal one, and the variable  $b_0$  models the deviation of the rotor radius from the nominal one. By reducing the variability on these two variables, the variability of the output current will be reduced. The other variables seem not to impact the output current, but they may have a significant impact on other performances of the machine, such as acoustic noise.

## 5.6 Chapter Summary

In this Chapter, we presented a complete procedure of conducting a reliability analysis of the electrical machine; we started with the presentation of the electrical machine. Then, we proceeded by presenting different metrological measurements that were conducted to characterize the variability on the geometry. From these measurements, we have derived a simple stochastic model representing the deviation of the machine's geometry according to the nominal geometry. This model is defined by several random parameters, and their probability density functions have been identified. Afterwards, we presented a parametric finite element model of the machine and how the measurement can be taken into account in the numerical model.

The numerical evaluation of the impact of geometry variables on the performances of the machine was conducted using a meta-model approach to reduce the computational cost. The variability of the performance was then characterized, and the variables that influence this variability were also identified.

Optimization under uncertainty of the electrical machine was not considered in this dissertation. Nevertheless, the finite element model is developed for this purpose. Other variables (shape, functioning temperatures, diode characteristics, ...) can be considered; thus, the optimization can be performed if further computational power is available.





# Concluding remarks

## Conclusion

This dissertation dealt with the approaches for optimization and reliability analysis of electrical machines modelled by finite element method. The finite element method is the most sophisticated tool to model electromagnetic phenomena in electrical machines. It enables the fine modelling of the electromagnetic fields in the studied domain and handling complex geometries, such as the claw-pole machine. However, it is computationally expensive because of nonlinear materials, 3D geometries, time dependency. Thus, its use for optimization and reliability analysis (iterative processes) should be made with caution since only a limited number of evaluations of the simulation tool can be tolerated.

General optimization algorithms cannot be applied as they are; some necessary modifications are needed to make them usable for optimizing electrical machines. In the first chapter, we presented the literature about optimization algorithms, and we discussed their adaptability to this dissertation's subject. From the categories, two approaches were considered, which depend on the capabilities of the model used.

1. A non-intrusive approach based on the Kriging meta-model
2. An intrusive approach based on the adjoint variable method

The non-intrusive approach using surrogate models is widely used in the context of expensive model optimization. Many researchers had highlighted the challenges of using such a strategy. In this dissertation, we have emphasized some of the very well-known issues and proposed a new methodology of using the meta-models and how to speed up the optimization times. We have highlighted the drawbacks of using the conventional approach, which consists in fitting a single meta-model on the whole design space and enriching it sequentially using infill criteria. We proposed a novel strategy that consists in building many meta-models on specified regions of the design space; each one of them is relatively easy to fit and easy to use for optimization (less modality). Then, iteratively, prune the regions that are not promising. This process explores all the design space; thus, it enables us to produce reliable solutions.

On the other hand, for the intrusive approach, we propose how to compute the derivatives from a finite element model. Most of the work has been done on the model rather than on the algorithm (we used the SQP algorithm for this dissertation). We have presented the adjoint variable method and how the gradient can be derived from a FEM code. We have developed an efficient way to compute the derivatives of the shape sensitivities for geometric parameters, which are vital for the gradient's computation. The adjoint variable method was compared to finite-difference to validate and highlight its effectiveness in terms of precision and computational time.

For the comparison between intrusive and non-intrusive approaches for optimizing electromagnetic devices using FEM, we treated two well-known benchmarks from the literature known as TEAM Workshop problems 22 and 25 [1] [2]. We used two metrics for the comparison, the first one is the quality of the solution, and the second is the computational cost.

In terms of performances, the intrusive approach outperforms other strategies for both test cases; this was possible due to the computation of the gradient using the adjoint variable method. This improves the quality of the solutions drastically but comes with the expense of intrusive manipulation of the FEM code. The non-intrusive approach remains a good alternative in terms of implementation and solution quality. The developed approach was able to overcome some of the very known issues when using meta-models for optimization.

In the last Chapter, we presented a complete procedure for conducting a reliability analysis of the electrical machine; we started with the presentation of electrical machines. Then, we proceeded by presenting the different metrological measurements that were conducted to characterize the variability on its shape. Afterwards, we presented a parametric finite element model of the machine and how the measurement can be taken into account in the numerical model.

The numerical evaluation of the impact of geometry variables on the performances of the machine was conducted using a meta-model approach to reduce the computational cost. The variability of the performance was then characterized, and the variables that influence this variability were also identified.

In summary, the goals of this dissertation have been met. A comparison of the approaches appropriate for the optimization of electrical machines has been carried out. The advantages and shortcomings of each method have been highlighted.

The integration of reliability analysis in the design phase has been assessed on a real test case (a claw-pole machine). The impact of the manufacturing process on the geometry of the machine has been studied, and quantitative analysis has been accomplished.

## Perspectives

Further work is currently underway to enhance the abilities of the non-intrusive approach. The multi-objective strategy is further enhanced for reducing the computational burden. The Pareto front can overlap many sub-spaces, which increases the computational cost of the method. A new technique for space division is being tested to tackle this issue.

The calculation of the gradient with respect to geometrical variables using the adjoint variable method relies mainly on the shape sensitivities. The geometric parameterization of shape variables is still one of the shortcomings of the method. We have presented an approach based on the parametric equation of the geometric shapes; however, for very complex shapes, this can be very cumbersome. Some researchers use dedicated tools that couples the CAD software with the optimizer, such as The Computational Analysis PRogramming Interface (CAPRI) [107] [108]. CAPRI serves the purpose of providing custom communications from a computational software suite to the preferred CAD system. This allows designer access to the CAD system's geometry definitions and functionalities, providing the ability to query the CAD system whenever needed. Using this tool will enable us to compute the shape sensitivities related to variables automatically from the CAD software.

Another use of the adjoint variable method will be dedicated to topology optimization, where the number of variables is significant. The gradient of the quantities of interest is computed at a minimal cost. Different methods might be considered, such as the SIMP method and the level-set method.

For the claw-pole machine, optimization under uncertainty will be considered to exploit the model full capabilities developed during the thesis. Taking into account the uncertainty related to the geometry of the machine while varying its shape can lead to highly efficient, while reliable, machines when considering the actual manufacturing process.



# Bibliography

- [1] P. Alotto, U. Baumgartner, and F. Freschi, “SMES optimization Benchmark: TEAM workshop problem 22,” *TEAM workshop problem 22*, no. 1, pp. 1–4, 2008.
- [2] N. Takahashi, “Optimization of Die Press Model: TEAM workshop problem 25,” *TEAM workshop*, no. 1, pp. 1–9, 1996.
- [3] M. Powell, “Algorithm for minimization without exact line searches,” *Nonlinear programming*, vol. 9, p. 53, 1976.
- [4] P. Wolfe, “Convergence conditions for ascent methods. ii: Some corrections,” *SIAM review*, vol. 13, no. 2, pp. 185–188, 1971.
- [5] S. Ruder, “An overview of gradient descent optimization algorithms,” *arXiv preprint arXiv:1609.04747*, 2016.
- [6] J. R. Shewchuk *et al.*, “An introduction to the conjugate gradient method without the agonizing pain,” 1994.
- [7] R. Fletcher and C. M. Reeves, “Function minimization by conjugate gradients,” *The computer journal*, vol. 7, no. 2, pp. 149–154, 1964.
- [8] M. Avriel, *Nonlinear programming: analysis and methods*. Courier Corporation, 2003.
- [9] J. E. Dennis, Jr and J. J. Moré, “Quasi-newton methods, motivation and theory,” *SIAM review*, vol. 19, no. 1, pp. 46–89, 1977.
- [10] W. Karush, “Minima of functions of several variables with inequalities as side conditions,” Master’s thesis, University of Chicago, 1939.
- [11] H. Kuhn and A. Tucker, “Nonlinear programming,” in *Second Berkeley Symposium on Mathematical Statistics and Probability*, pp. 481–492, 1951.
- [12] M. Biggs, “Constrained minimization using recursive quadratic programming,” *Towards global optimization*, 1975.
- [13] M. Powell, “The convergence of variable metric methods for non-linearly constrained optimization calculations,” *Nonlinear Programming*, vol. 3, 1978.

- [14] R. T. Marler and J. S. Arora, “The weighted sum method for multi-objective optimization: new insights,” *Structural and multidisciplinary optimization*, vol. 41, no. 6, pp. 853–862, 2010.
- [15] Y. Haimes, “On a bicriterion formulation of the problems of integrated system identification and system optimization,” *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 1, no. 3, pp. 296–297, 1971.
- [16] J. A. Nelder and R. Mead, “A simplex method for function minimization,” *The computer journal*, vol. 7, no. 4, pp. 308–313, 1965.
- [17] J. C. Lagarias, J. A. Reeds, M. H. Wright, and P. E. Wright, “Convergence properties of the nelder–mead simplex method in low dimensions,” *SIAM Journal on optimization*, vol. 9, no. 1, pp. 112–147, 1998.
- [18] D. R. Jones, C. D. Perttunen, and B. E. Stuckman, “Lipschitzian optimization without the lipschitz constant,” *Journal of optimization Theory and Applications*, vol. 79, no. 1, pp. 157–181, 1993.
- [19] M. Sasena, P. Papalambros, and P. Goovaerts, “Global optimization of problems with disconnected feasible regions via surrogate modeling,” in *9th AIAA/ISSMO Symposium on Multidisciplinary Analysis and Optimization*, p. 5573, 2002.
- [20] K. Ljungberg, S. Holmgren, and Ö. Carlborg, “Simultaneous search for multiple qtl using the global optimization algorithm direct,” *Bioinformatics*, vol. 20, no. 12, pp. 1887–1895, 2004.
- [21] V. Torczon, “On the convergence of pattern search algorithms,” *SIAM Journal on optimization*, vol. 7, no. 1, pp. 1–25, 1997.
- [22] M. J. Powell, “A view of algorithms for optimization without derivatives,” *Mathematics Today-Bulletin of the Institute of Mathematics and its Applications*, vol. 43, no. 5, pp. 170–174, 2007.
- [23] M. S. Bazaraa, H. D. Sherali, and C. M. Shetty, *Nonlinear programming: theory and algorithms*. John Wiley & Sons, 2013.
- [24] T. Back, *Evolutionary algorithms in theory and practice: evolution strategies, evolutionary programming, genetic algorithms*. Oxford university press, 1996.
- [25] P. Alotto, C. Eranda, B. Brandstatter, G. Furntratt, C. Magele, G. Molinari, M. Nervi, K. Preis, M. Repetto, and K. R. Richter, “Stochastic algorithms in electromagnetic optimization,” *IEEE Transactions on Magnetics*, vol. 34, no. 5, pp. 3674–3684, 1998.
- [26] D. E. Goldberg and J. H. Holland, “Genetic algorithms and machine learning,” *Machine learning*, vol. 3, no. 2, pp. 95–99, 1988.

- 
- [27] K. Deb, “An efficient constraint handling method for genetic algorithms,” *Computer methods in applied mechanics and engineering*, vol. 186, no. 2-4, pp. 311–338, 2000.
- [28] E. L. Lawler and D. E. Wood, “Branch-and-bound methods: A survey,” *Operations research*, vol. 14, no. 4, pp. 699–719, 1966.
- [29] F. Messine, B. Nogarede, and J.-L. Lagouanelle, “Optimal design of electromechanical actuators: a new method based on global optimization,” *IEEE transactions on magnetics*, vol. 34, no. 1, pp. 299–308, 1998.
- [30] S. Brisset and T.-V. Tran, “Pareto-based branch and bound algorithm for multiobjective optimization of a safety transformer,” *COMPEL-The international journal for computation and mathematics in electrical and electronic engineering*, vol. 37, no. 2, pp. 617–629, 2018.
- [31] J. Fontchastagner, F. Messine, and Y. Lefèvre, “Design of electrical rotating machines by associating deterministic global optimization algorithm with combinatorial analytical and numerical models,” *IEEE Transactions on Magnetics*, vol. 43, no. 8, pp. 3411–3419, 2007.
- [32] J. Fontchastagner, *Résolution du problème inverse de conception d'actionneurs électromagnétiques par association de méthodes déterministes d'optimisation globale avec des modèles analytiques et numériques*. PhD thesis, INP Toulouse, 2007.
- [33] D. R. Jones, “A taxonomy of global optimization methods based on response surfaces,” *Journal of global optimization*, vol. 21, no. 4, pp. 345–383, 2001.
- [34] R. H. Myers, D. C. Montgomery, and C. M. Anderson-Cook, *Response surface methodology: process and product optimization using designed experiments*. John Wiley & Sons, 2016.
- [35] N. Cressie, “The origins of kriging,” *Mathematical geology*, vol. 22, no. 3, pp. 239–252, 1990.
- [36] J. D. Martin and T. W. Simpson, “Use of Kriging Models to Approximate Deterministic Computer Models,” *AIAA Journal*, vol. 43, no. 4, pp. 853–863, 2005.
- [37] M. Schonlau, *Computer experiments and global optimization*. PhD thesis, University of Waterloo, 1997.
- [38] V. Picheny, T. Wagner, and D. Ginsbourger, “A benchmark of kriging-based infill criteria for noisy optimization,” *Structural and Multidisciplinary Optimization*, vol. 48, no. 3, pp. 607–626, 2013.



- [39] M. J. Sasena, *Flexibility and Efficiency Enhancements for Constrained Global Design Optimization with Kriging Approximations*. PhD Thesis, University of Michigan, 2002.
- [40] D. D. Cox and S. John, “A statistical method for global optimization,” in *Systems, Man and Cybernetics, 1992., IEEE International Conference on*, vol. 2, pp. 1241–1246, 1992.
- [41] R. El Bechari, S. Brisset, S. Clenet, and J.-C. Mipo, “Enhanced Meta-Model-Based Optimization Under Constraints Using Parallel Computations,” *IEEE Transactions on Magnetics*, vol. 54, pp. 1–4, Mar. 2018.
- [42] Y.-C. Ho and D. L. Pepyne, “Simple explanation of the no-free-lunch theorem and its implications,” *Journal of optimization theory and applications*, vol. 115, no. 3, pp. 549–570, 2002.
- [43] D. R. Jones, M. Schonlau, and W. J. Welch, “Efficient Global Optimization of Expensive Black-Box Functions,” *Journal of Global Optimization*, vol. 13, pp. 455–492, 1998.
- [44] J. Hammersley, *Monte carlo methods*. Springer Science & Business Media, 2013.
- [45] S. Liu, *Prise en compte des incertitudes dimensionnelles introduites par les procédés de fabrication dans les modèles numériques de machines électriques*. PhD thesis, ENSAM ParisTech, Lille, 2015.
- [46] S. Liu, H. Mac, S. Clenet, T. Coorevits, and J.-C. Mipo, “Study of the influence of the fabrication process imperfections on the performance of a claw pole synchronous machine using a stochastic approach,” *IEEE Transactions on Magnetics*, vol. 52, no. 3, pp. 1–4, 2015.
- [47] T. T. Nguyen, D. H. Mac, and S. Clenet, “Uncertainty Quantification Using Sparse Approximation for Models With a High Number of Parameters: Application to a Magnetoelectric Sensor,” *IEEE Transactions on Magnetics*, vol. 52, pp. 1–4, Mar. 2016.
- [48] L. Picheral, *Contribution à la conception préliminaire robuste en ingénierie de produit*. PhD thesis, Université de Grenoble, Grenoble, France, 2013.
- [49] S. Deng, *Optimisation robuste pour des dispositifs électromagnétiques*. PhD thesis, Ecole Centrale de Lille, Lille, France, 2018.
- [50] X. Du and W. Chen, “A most probable point-based method for efficient uncertainty analysis,” *Journal of Design and Manufacturing automation*, vol. 4, no. 1, pp. 47–66, 2001.

- 
- [51] D. Pflugfelder, J. Wilkens, and U. Oelfke, “Worst case optimization: a method to account for uncertainties in the optimization of intensity modulated proton therapy,” *Physics in Medicine & Biology*, vol. 53, no. 6, p. 1689, 2008.
- [52] T. Crestaux, O. Le Maitre, and J.-M. Martinez, “Polynomial chaos expansion for sensitivity analysis,” *Reliability Engineering & System Safety*, vol. 94, no. 7, pp. 1161–1172, 2009.
- [53] G. Blatman and B. Sudret, “Adaptive sparse polynomial chaos expansion based on least angle regression,” *Journal of Computational Physics*, vol. 230, no. 6, pp. 2345–2367, 2011.
- [54] D. G. Cacuci, M. Ionescu-Bujor, and I. M. Navon, *Sensitivity and uncertainty analysis, volume II: applications to large-scale systems*. CRC press, 2005.
- [55] A. M. Hasofer and N. C. Lind, “Exact and invariant second-moment code format,” *Journal of the Engineering Mechanics division*, vol. 100, no. 1, pp. 111–121, 1974.
- [56] P.-L. Liu and A. Der Kiureghian, “Optimization algorithms for structural reliability,” *Structural safety*, vol. 9, no. 3, pp. 161–177, 1991.
- [57] Z. Wang, H.-Z. Huang, and Y. Liu, “A unified framework for integrated optimization under uncertainty,” *Journal of Mechanical Design*, vol. 132, no. 5, 2010.
- [58] X. Du and W. Chen, “Sequential optimization and reliability assessment method for efficient probabilistic design,” *J. Mech. Des.*, vol. 126, no. 2, pp. 225–233, 2004.
- [59] G. Steiner, A. Weber, and C. Magele, “Managing uncertainties in electromagnetic design problems with robust optimization,” *IEEE transactions on magnetics*, vol. 40, no. 2, pp. 1094–1099, 2004.
- [60] T. Nakata, N. Takahashi, K. Fujiwara, T. Imai, and K. Muramatsu, “Comparison of various methods of analysis and finite elements in 3-d magnetic field analysis,” *IEEE transactions on magnetics*, vol. 27, no. 5, pp. 4073–4076, 1991.
- [61] R. Albanese and G. Rubinacci, “Magnetostatic field computations in terms of two-component vector potentials,” *International journal for numerical methods in engineering*, vol. 29, no. 3, pp. 515–532, 1990.
- [62] D. J. Griffiths, *Introduction to Electrodynamics*, vol. 2. Cambridge University Press, 2017.
- [63] P. Monk *et al.*, *Finite element methods for Maxwell’s equations*. Oxford University Press, 2003.

- [64] V. Girault and P.-A. Raviart, “Finite element methods for navier-stokes equations: Theory and algorithms,” *NASA STI/Recon Technical Report A*, vol. 87, 1986.
- [65] D. Meeker, “Inductance calculation example,” 2009. <http://www.femm.info/wiki/InductanceExample>.
- [66] R. Jin, W. Chen, and T. W. Simpson, “Comparative Studies of Metamodeling Techniques Under Multiple Modeling Criteria,” *Structural and Multidisciplinary Optimization*, vol. 23, no. 1, pp. 1–13, 2001.
- [67] M. Li, F. Gabriel, M. Alkadri, and D. A. Lowther, “Kriging-Assisted Multi-Objective Design of Permanent Magnet Motor for Position Sensorless Control,” *IEEE Transactions on Magnetics*, vol. 52, no. 3, pp. 1–4, 2016.
- [68] J. P. C. Kleijnen, W. Van Beers, and I. Van Nieuwenhuysse, “Expected improvement in efficient global optimization through bootstrapped kriging,” *Journal of Global Optimization*, vol. 54, no. 1, pp. 59–73, 2012.
- [69] T. W. Simpson, *A concept exploration method for product family design*. PhD thesis, School Mechanical Engineering, Georgia Institute of Technology, 1998.
- [70] J. Santiago, M. Claeys-Bruno, and M. Sergent, “Construction of space-filling designs using wsp algorithm for high dimensional spaces,” *Chemometrics and Intelligent Laboratory Systems*, vol. 113, pp. 26–31, 2012.
- [71] J. Sacks, W. J. Welch, T. J. Mitchell, and H. P. Wynn, “Design and analysis of computer experiments,” *Statistical science*, pp. 409–423, 1989.
- [72] K. Mardia and A. Watkins, “On multimodality of the likelihood in the spatial linear model,” *Biometrika*, vol. 76, no. 2, pp. 289–295, 1989.
- [73] C. Durantin, J. Marzat, and M. Balesdent, “Analysis of multi-objective Kriging-based methods for constrained global optimization,” *Computational Optimization and Applications*, vol. 63, no. 3, pp. 903–926, 2016.
- [74] M. J. Sasena, P. Papalambros, and P. Goovaerts, “Exploration of Metamodeling Sampling Criteria for Constrained Global Optimization,” *Engineering Optimization*, vol. 34, pp. 263–278, Jan. 2002.
- [75] S. Rojas-Gonzalez and I. Van Nieuwenhuysse, “A survey on kriging-based infill algorithms for multiobjective simulation optimization.,” *Computers & Operations Research*, p. 104869, 2019.
- [76] M. T. Emmerich, K. C. Giannakoglou, and B. Naujoks, “Single-and multiobjective evolutionary optimization assisted by gaussian random field metamodels,” *IEEE Transactions on Evolutionary Computation*, vol. 10, no. 4, pp. 421–439, 2006.

- 
- [77] M. Li, G. Li, and S. Azarm, “A kriging metamodel assisted multi-objective genetic algorithm for design optimization,” *Journal of Mechanical Design*, vol. 130, no. 3, 2008.
- [78] S. Deng, R. El Bechari, S. Brisset, and S. Clenet, “Iterative Kriging-Based Methods for Expensive Black-Box Models,” *IEEE Transactions on Magnetics*, vol. 54, pp. 1–4, Mar. 2018.
- [79] H. Mohammadi, R. L. Riche, N. Durrande, E. Touboul, and X. Bay, “An analytic comparison of regularization methods for gaussian processes,” *arXiv preprint arXiv:1602.00853*, 2016.
- [80] R. Ababou, A. C. Bagtzoglou, and E. F. Wood, “On the condition number of covariance matrices in kriging, estimation, and simulation of random fields,” *Mathematical Geology*, vol. 26, no. 1, pp. 99–133, 1994.
- [81] J. Iott, R. T. Haftka, and H. M. Adelman, “Selecting step sizes in sensitivity analysis by finite differences,” *NASA Technical Memorandum*, vol. 86382, 2007.
- [82] R. Mathur, *An analytical approach to computing step sizes for finite-difference derivatives*. PhD thesis, University of Texas, 2012.
- [83] C. L. Bottasso, D. Detomi, and R. Serra, “The ball-vertex method: a new simple spring analogy method for unstructured dynamic meshes,” *Computer Methods in Applied Mechanics and Engineering*, vol. 194, no. 39-41, pp. 4244–4264, 2005.
- [84] C. Farhat, C. Degand, B. Koobus, and M. Lesoinne, “Torsional springs for two-dimensional dynamic unstructured fluid meshes,” *Computer methods in applied mechanics and engineering*, vol. 163, no. 1-4, pp. 231–245, 1998.
- [85] P. Hansbo, “Generalized laplacian smoothing of unstructured grids,” *Communications in numerical methods in engineering*, vol. 11, no. 5, pp. 455–464, 1995.
- [86] R. P. Dwight, “Robust mesh deformation using the linear elasticity equations,” in *Computational fluid dynamics 2006*, pp. 401–406, Springer, 2009.
- [87] T. Henneron, A. Pierquin, and S. Cl enet, “Mesh deformation based on radial basis function interpolation applied to low-frequency electromagnetic problem,” *IEEE Transactions on Magnetics*, vol. 55, no. 6, pp. 1–4, 2019.
- [88] A. De Boer, M. Van der Schoot, and H. Bijl, “Mesh deformation based on radial basis function interpolation,” *Computers & structures*, vol. 85, no. 11-14, pp. 784–795, 2007.

- [89] M. S. Berkani, S. Giurgea, C. Espanet, J.-L. Coulomb, and C. Kieffer, "Study on optimal design based on direct coupling between a fem simulation model and l-bfgs-b algorithm," *IEEE Transactions on Magnetics*, vol. 49, no. 5, pp. 2149–2152, 2013.
- [90] H.-b. Lee and N. Ida, "Interpretation of adjoint sensitivity analysis for shape optimal design of electromagnetic systems," *IET Science, Measurement & Technology*, vol. 9, no. 8, pp. 1039–1042, 2015.
- [91] I.-h. Park, B.-t. Lee, and S.-y. Hahn, "Design sensitivity analysis for nonlinear magnetostatic problems using finite element method," *IEEE transactions on magnetics*, vol. 28, no. 2, pp. 1533–1536, 1992.
- [92] D.-H. Kim, K. Ship, and J. Sykulski, "Applying continuum design sensitivity analysis combined with standard em software to shape optimization in magnetostatic problems," *IEEE Transactions on Magnetics*, vol. 40, no. 2, pp. 1156–1159, 2004.
- [93] M. P. Bendsøe, "Optimal shape design as a material distribution problem," *Structural optimization*, vol. 1, no. 4, pp. 193–202, 1989.
- [94] B. S. B. Mohamodhosen, *Topology optimisation of electromagnetic devices*. PhD thesis, Ecole Centrale de Lille, 2017.
- [95] Y. Okamoto, K. Akiyama, and N. Takahashi, "3-d topology optimization of single-pole-type head by using design sensitivity analysis," *IEEE Transactions on Magnetics*, vol. 42, no. 4, pp. 1087–1090, 2006.
- [96] T. W. Sederberg and S. R. Parry, "Free-form deformation of solid geometric models," in *Proceedings of the 13th annual conference on Computer graphics and interactive techniques*, pp. 151–160, 1986.
- [97] L. Piegl and W. Tiller, *The NURBS book*. Springer Science & Business Media, 2012.
- [98] Z. Bontinck, J. Corno, H. De Gersem, S. Kurz, A. Pels, S. Schöps, F. Wolf, C. de Falco, J. Dölz, R. Vázquez, *et al.*, "Recent advances of isogeometric analysis in computational electromagnetics," *arXiv preprint arXiv:1709.06004*, 2017.
- [99] D. E. Finkel, "Direct optimization algorithm user guide," *Center for Research in Scientific Computation, North Carolina State University*, vol. 2, pp. 1–14, 2003.
- [100] I. The MathWorks, *Matlab, Global Optimization Toolbox*. Natick, Massachusetts, United State, 2019.

- [101] M. El Youssef, A. Van Gorp, S. Clenet, A. Benabou, P. Faverolle, J.-C. Mipo, Y. Lavalley, C. Cour, and T. Lecuppe, "Slinky stator: The impact of manufacturing process on the magnetic properties," in *2017 IEEE International Electric Machines and Drives Conference (IEMDC)*, pp. 1–8, IEEE, 2017.
- [102] M. Borsenberger, C. Baudouin, A. Benabou, R. Bigot, P. Faverolle, and J.-C. Mipo, "Methodology for the analysis of the impact of the forging parameters on metallurgy and mechanical properties in case of solid electromagnetic manufactured parts," in *AIP Conference Proceedings*, p. 130002, AIP Publishing LLC, 2016.
- [103] S. Liu, S. Clenet, T. Coorevits, and J. Mipo, "Influence of the stator deformation on the behaviour of a claw-pole generator," in *2014 17th International Conference on Electrical Machines and Systems (ICEMS)*, pp. 358–362, IEEE, 2014.
- [104] J. Kaufmann and A. Schering, "Analysis of variance anova," *Wiley StatsRef: Statistics Reference Online*, 2014.
- [105] M. Stein, "Large sample properties of simulations using latin hypercube sampling," *Technometrics*, vol. 29, no. 2, pp. 143–151, 1987.
- [106] S. Cl enet, "Approximation methods to solve stochastic problems in computational electromagnetics," in *Scientific Computing in Electrical Engineering*, pp. 199–214, Springer, 2016.
- [107] W. Brock, C. Burdyshaw, S. Karman, V. Betro, B. Hilbert, K. Anderson, and R. Haimes, "Adjoint-based design optimization using cad parameterization through capri," in *50th AIAA Aerospace Sciences Meeting including the New Horizons Forum and Aerospace Exposition*, p. 968, 2012.
- [108] R. Haimes, "Capri cae gateway," 2007.



# Appendix A

## Shape sensitivities

In this appendix, we derive the shape sensitivities for some recurrent types of geometries.

### A.1 Rectangle

A parametric equation of the rectangle can be written as

$$\begin{aligned}x(t) &= a + \frac{d}{2} \text{sign}(\cos(t)) \\y(t) &= b + \frac{h}{2} [1 + \text{sign}(\sin(t))] \quad , t \in [0, 2\pi[ \end{aligned}$$

Then, the shape sensitivities are calculated with respect to the variable defining the rectangle are

$$d_a x = 1$$

$$d_a y = 0$$

$$d_b x = 0$$

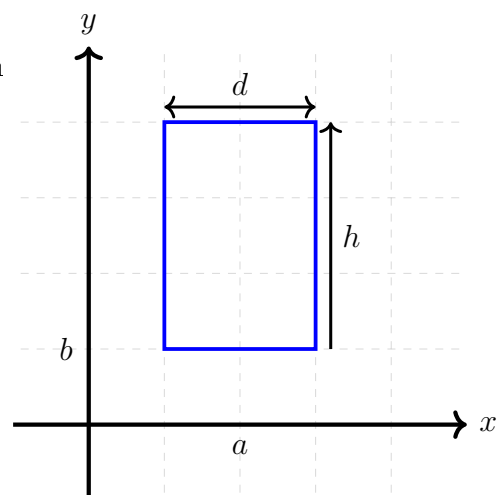
$$d_b y = 1$$

$$d_d x = \frac{1}{2} \text{sign}(\cos(t)) = \frac{x - a}{d}$$

$$d_d y = 0$$

$$d_h x = 0$$

$$d_h y = \frac{1}{2} [1 + \text{sign}(\sin(t))] = \frac{y - b}{h}$$





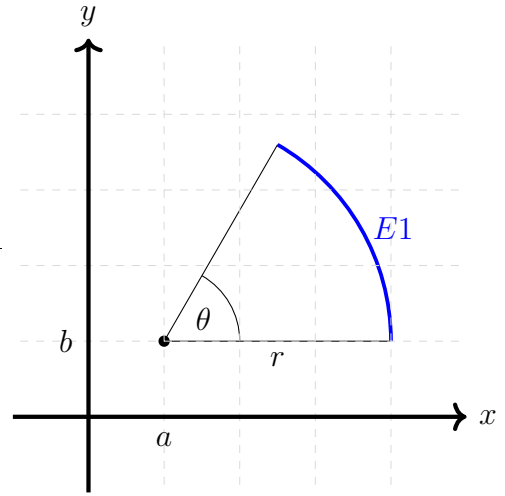
## A.2 Circle

A parametric equation of a circle arc  $E1$  can be written as

$$\begin{aligned}x(t) &= a + r \cos(t) \\y(t) &= b + r \sin(t), \quad t \in [0, \theta]\end{aligned}$$

Then, the shape sensitivities are calculated with respect to  $r$  are

$$\begin{aligned}d_r x &= \cos(t) = \frac{x - a}{r} \\d_r y &= \sin(t) = \frac{y - b}{r}\end{aligned}$$



## A.3 Other shapes

In a general manner, one can compute the shape sensitivities if the considered shape can be written as parametric equations of the edges defining it. We showed the shape sensitivities for the circular arc and the rectangle since there will be used in the application of the end of this chapter and the following chapter, where we will treat some optimization benchmarks.

# Appendix B

## FEM formulation in cylindrical coordinates

This Appendix deals with the variational formulation for axisymmetric electromagnetic devices. There is few researchers who addressed this aspect. Here we present, the formulation that we used in MagFEM. The formulation is used for the modelling of the SMES device of the TEAM Workshop problem 22.

### B.1 General equations

As detailed in Chapter 1, Maxwell equations are partial differential equations (PDE) (Hard formulation). These equations can be solved analytically for simple systems. For complex ones, numerical methods are best suited using weak formulations, *i.e.* variational formulation.

The magnetostatic problem is written as follows.

$$\nabla \times \left( \frac{1}{\mu} \nabla \times \mathbf{A} \right) = \mathbf{J}_s \quad \text{in } \mathcal{D} \quad (\text{B.1})$$

$$\mathbf{n} \times \mathbf{A} = \mathbf{0} \quad \text{on } \Gamma_B \quad (\text{B.2})$$

$$\left( \frac{1}{\mu} \nabla \times \mathbf{A} \right) \times \mathbf{n} = \mathbf{0} \quad \text{on } \Gamma_H \quad (\text{B.3})$$

These equations describe the electromagnetic fields and the corresponding boundary conditions.

## B.2 Variational Formulation

Multiplying the PDE from B.1 by an arbitrary test function  $\mathbf{v}$  such that  $\mathbf{n} \times \mathbf{v}|_{\Gamma_B} = \mathbf{0}$  and integrating over the domain leads to

$$\int_{\mathcal{D}} \left[ \nabla \times \left( \frac{1}{\mu} \nabla \times \mathbf{A} \right) \right] \cdot \mathbf{v} = \int_{\mathcal{D}} \mathbf{J}_s \cdot \mathbf{v} \quad (\text{B.4})$$

Applying Green-Ostrogradski formulas on the l.h.s

$$\begin{aligned} \int_{\mathcal{D}} \left[ \nabla \times \left( \frac{1}{\mu} \nabla \times \mathbf{A} \right) \right] \cdot \mathbf{v} &= \int_{\mathcal{D}} \left( \frac{1}{\mu} \nabla \times \mathbf{A} \right) \cdot \nabla \times \mathbf{v} \\ &\quad - \int_{\Gamma} \left[ \left( \frac{1}{\mu} \nabla \times \mathbf{A} \right) \times \mathbf{n} \right] \cdot \mathbf{v} \end{aligned} \quad (\text{B.5})$$

Since  $\mathbf{v}$  vanishes on  $\Gamma_B$  (Dirichlet boundary) and  $\left( \frac{1}{\mu} \nabla \times \mathbf{A} \right) \times \mathbf{n}$  vanishes on  $\Gamma_H$  (Neumann boundary), the boundary integral vanishes too.

This leads to the variational formulation

$$a(\mathbf{A}, \mathbf{v}) = \langle \mathbf{F}, \mathbf{v} \rangle \quad (\text{B.6})$$

with

$$a(\mathbf{A}, \mathbf{v}) = \int_{\mathcal{D}} \left( \frac{1}{\mu} \nabla \times \mathbf{A} \right) \cdot \nabla \times \mathbf{v} \quad (\text{B.7})$$

$$\langle \mathbf{F}, \mathbf{v} \rangle = \int_{\mathcal{D}} \mathbf{J}_s \cdot \mathbf{v} \quad (\text{B.8})$$

In axisymmetric case, the potential vector  $\mathbf{A}$  and the flux density  $\mathbf{B}$  are reduced to the following thanks to the invariance around the symmetry axis.

$$\mathbf{A} = \begin{pmatrix} 0 \\ A_\theta(r, z) \\ 0 \end{pmatrix} \quad (\text{B.9})$$

$$\mathbf{B} = \nabla \times \mathbf{A} = \begin{pmatrix} -\frac{\partial A_\theta}{\partial z} \\ 0 \\ \frac{1}{r} \frac{\partial r A_\theta}{\partial r} \end{pmatrix} \quad (\text{B.10})$$

Then, the problem is reduced to

$$a(\mathbf{A}, \mathbf{v}) = \int_{\mathcal{D}} \left( \frac{1}{\mu r} \nabla_c(r A_\theta) \right) \cdot \frac{1}{r} \nabla_c(r v_\theta) r dr d\theta dz \quad (\text{B.11})$$

$$\langle \mathbf{F}, \mathbf{v} \rangle = \int_{\mathcal{D}} J_{s\theta} v_\theta r dr d\theta dz \quad (\text{B.12})$$

where  $\nabla_c f = \begin{pmatrix} \frac{\partial f}{\partial r} \\ \frac{\partial f}{\partial \theta} \\ \frac{\partial f}{\partial z} \end{pmatrix}$  is the gradient of  $f$  in the Cartesian coordinates.

The variational formulation can be further simplified by putting it in a form similar to the 2D case; thus, we set a new test function

$$w = rv_\theta$$

and replace the unknown potential by setting

$$u = rA_\theta$$

Then the variational formulation becomes

$$a(u, w) = \int_{\mathcal{D}} \left( \frac{1}{\mu r} \nabla_c u \right) \cdot \nabla_c w \, dr d\theta dz \quad (\text{B.13})$$

$$\langle \mathbf{F}, w \rangle = \int_{\mathcal{D}} J_{s\theta} w \, dr d\theta dz \quad (\text{B.14})$$

Except for the term  $\frac{1}{r}$  in  $a(u, w)$ , this formulation is similar to the 2D. Thus, the discretization is the same and the quantities of interest can be computed from the  $u$ , for example, the flux density is calculated as follows.

$$\mathbf{B} = \frac{1}{r} \begin{pmatrix} -\frac{\partial u}{\partial z} \\ 0 \\ \frac{\partial u}{\partial r} \end{pmatrix} \quad (\text{B.15})$$



# Appendix C

## Adjoint equation derivation for TEAM Workshop problem 25

This appendix treats the derivation of the gradient equations using the adjoint variable method for the TEAM Workshop problem 25 [2].

### C.1 Gradient equations

In general, Finite element model solve a set of equations  $g(u, p) = K(u, p)u - b(p) = 0$  for  $u$  parameterized by variables  $p$  (design parameters) and compute a quantity of interest  $f(u, p)$

To recall from Section 3.2.2, for computing the gradient of a function  $f$ , the following steps has to be conducted :

1. Compute the partial derivative for  $u$  :  $\partial_u \tilde{f}$  and  $\partial_u \tilde{g}$
2. Solve the linear system:  $\lambda \partial_u \tilde{g} = -\partial_u \tilde{f}$  for  $\lambda$
3. Compute the partial derivatives for  $p$  :  $\partial_p \tilde{f}$  and  $\partial_p \tilde{g}$
4. Calculate the gradient using

$$\partial_p f = \partial_p \tilde{f} + \lambda \partial_p \tilde{g} \tag{C.1}$$

- $p$  : Design variables
- $u$  : State variables
- $\tilde{g}(u, p)$  : State equation
- $\tilde{f}(u, p)$  : Quantity of interest
- $\lambda$  : Adjoint variables

### C.2 TEAM Workshop problem 25

The full description of the problem is detailed in Section 4.1.2.

### C.2.1 Design variables

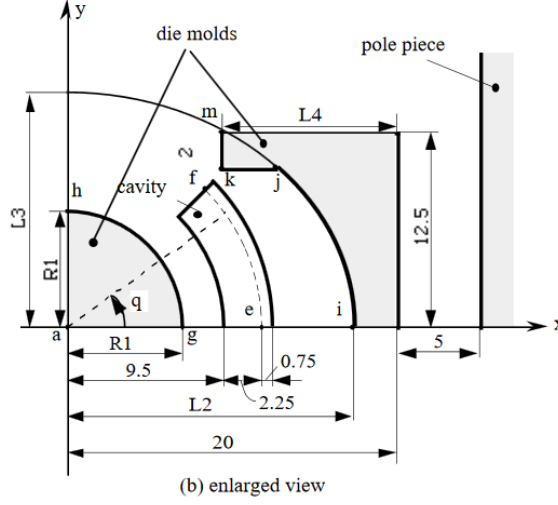


Fig.1 Model of die press with electromagnet.

Figure C.1: Molds of the Die press (design region)

$p = (R1, L2, L3, L4)$  are the design variables. The design variables are geometric ones, then, from Section 3.3.3, the partial derivatives  $\partial_p \tilde{f}$  and  $\partial_p \tilde{g}$  are decomposed as follows

$$\partial_p \tilde{f} = \sum_{k=1}^N \left[ \partial_{x_k} \tilde{f} d_p x_k + \partial_{y_k} \tilde{f} d_p y_k \right] \quad (C.2)$$

$$\partial_p \tilde{g} = \sum_{k=1}^N \left[ \partial_{x_k} \tilde{g} d_p x_k + \partial_{y_k} \tilde{g} d_p y_k \right] \quad (C.3)$$

where  $(x_k, y_k)$  for  $k = 1, \dots, N$  are the node coordinates used for the FEA.

The quantities  $\partial_{x_k} \tilde{f}$ ,  $\partial_{y_k} \tilde{f}$ ,  $\partial_{x_k} \tilde{g}$  and  $\partial_{y_k} \tilde{g}$  are computed independently for  $p$  and will be detailed later. Here, we focus on the computation of  $d_p x_k$  and  $d_p y_k$ .

#### A Variable $R1$

The variable  $R1$  parameterize a circle arc ( $gh$  in Figure C.1), then, from Appendix A, we can computed the quantities  $d_{R1} x_k$  and  $d_{R1} y_k$  for the mesh node coordinates on the arc.

$$d_{R1} x_k = \frac{x_k}{R1}$$

$$d_{R1} y_k = \frac{y_k}{R1}$$

It is worth noting that these quantities are equal to 0 on all the other nodes.

## B Variables $L2$ and $L3$

The variables  $L2$  and  $L3$  parameterize an ellipse arc ( $ij$  in Figure C.1). The equation of an ellipse is written as

$$\left(\frac{x}{L2}\right)^2 + \left(\frac{y}{L3}\right)^2 = 1 \quad (\text{C.4})$$

Then, the arc  $ij$  can be parameterized by the following equations.

$$\begin{aligned} x(t) &= L2\sqrt{1 - \left(\frac{t}{L3}\right)^2} \\ y(t) &= t \end{aligned} \quad , t \in [0, y_m]$$

where  $y_m = 10.5\text{mm}$  is  $y$ -coordinate of the vertex  $j$  in Figure C.1.

$$\begin{aligned} d_{L2}x_k &= \sqrt{1 - \left(\frac{t}{L3}\right)^2} &&= \frac{x_k}{L2} \\ d_{L2}y_k &= 0 \\ d_{L3}x_k &= \frac{L2t^2}{L3^3\sqrt{1 - \left(\frac{t}{L3}\right)^2}} &&= \frac{L2^2y_k^2}{L3^3x_k} \\ d_{L3}y_k &= 0 \end{aligned}$$

## C Variable $L4$

The variable  $L4$  parameterize the segment  $km$  in Figure C.1. The parametric equation of this segment can be written as

$$\begin{aligned} x(t) &= 20 - L4 \\ y(t) &= t \end{aligned} \quad , t \in [10.5, 12.5]$$

Then, the derivatives on the segment  $km$  are deduced:

$$\begin{aligned} d_{L4}x_k &= -1 \\ d_{L4}y_k &= 0 \end{aligned}$$

### C.2.2 State equations

The device modeled as a 2D planar nonlinear magnetostatic problem, The state equations are detailed in Section 1.3.3



$$g(u, p) = K(u, p)u - b(p) = 0 \quad (\text{C.5})$$

where

$$K_{ij} = \sum_{T \in \mathcal{T}_h} \sum_g \frac{1}{\mu} \nabla \alpha_i^g J_T^{-1} \cdot \nabla \alpha_j^g J_T^{-1} |J_T| w_g$$

$$b_j = \sum_{T \in \mathcal{T}_h} \sum_g J_{s3} \alpha_j^g |J_T| w_g$$

### A Derivatives w.r.t mesh nodes

The derivatives  $\partial_{x_k} \tilde{g}$  and  $\partial_{y_k} \tilde{g}$  are computed as follows

$$\partial_{x_k} \tilde{g} = \partial_{x_k} K u - \partial_{x_k} b \quad (\text{C.6})$$

$$\partial_{y_k} \tilde{g} = \partial_{y_k} K u - \partial_{y_k} b \quad (\text{C.7})$$

such as

$$\begin{aligned} \partial_{x_k} K_{ij} &= \sum_{T \in \mathcal{T}_h} \sum_g \left[ \frac{1}{\mu} \nabla \alpha_i^g \partial_{x_k} (J_T^{-1}) \cdot \nabla \alpha_j^g J_T^{-1} |J_T| w_g \right. \\ &\quad + \frac{1}{\mu} \nabla \alpha_i^g J_T^{-1} \cdot \nabla \alpha_j^g \partial_{x_k} (J_T^{-1}) |J_T| w_g \\ &\quad \left. + \frac{1}{\mu} \nabla \alpha_i^g J_T^{-1} \cdot \nabla \alpha_j^g J_T^{-1} \partial_{x_k} (|J_T|) w_g \right] \\ \partial_{x_k} b_j &= \sum_{T \in \mathcal{T}_h} \sum_g J_{s3} \alpha_j^g \partial_{x_k} (|J_T|) w_g \\ \partial_{y_k} K_{ij} &= \sum_{T \in \mathcal{T}_h} \sum_g \left[ \frac{1}{\mu} \nabla \alpha_i^g \partial_{y_k} (J_T^{-1}) \cdot \nabla \alpha_j^g J_T^{-1} |J_T| w_g \right. \\ &\quad + \frac{1}{\mu} \nabla \alpha_i^g J_T^{-1} \cdot \nabla \alpha_j^g \partial_{y_k} (J_T^{-1}) |J_T| w_g \\ &\quad \left. + \frac{1}{\mu} \nabla \alpha_i^g J_T^{-1} \cdot \nabla \alpha_j^g J_T^{-1} \partial_{y_k} (|J_T|) w_g \right] \\ \partial_{y_k} b_j &= \sum_{T \in \mathcal{T}_h} \sum_g J_{s3} \alpha_j^g \partial_{y_k} (|J_T|) w_g \end{aligned}$$

These formula may seem intimidating, however, there are similar to the ones in the FE code. Only the derivatives of the jacobian  $J_T$  are needed to be computed and then put into the formulas.

### B Derivatives w.r.t state variables

The derivative  $\partial_u \tilde{g}$  is computed as follows

$$\partial_u \tilde{g} = \partial_u K u + K \quad (\text{C.8})$$

such as

$$\partial_u K_{ij} = \sum_{T \in \mathcal{T}_h} \sum_g \frac{-\partial_u \mu}{\mu^2} \nabla \alpha_i^g J_T^{-1} \cdot \nabla \alpha_j^g J_T^{-1} |J_T| w_g$$

The quantity  $\partial_u \tilde{g}$  is often computed in the finite element code. the subroutine could be reused in the adjoint variable code.

### C.2.3 Quantity of interest

The aim is to minimize the function  $W(p)$

$$W(p) = \sum_{i=1}^{10} (B_{xip}(p) - B_{xio})^2 + (B_{yip}(p) - B_{yio})^2 \quad (\text{C.9})$$

where  $B_{xio} = 0.35 \cos(\frac{\pi}{40}i)$ ,  $B_{yio} = 0.35 \sin(\frac{\pi}{40}i)$ . The quantities  $B_{xip}(p)$  and  $B_{yip}(p)$  are flux densities computed by the FEA of the device, thus, we need to compute the derivatives of these quantities w.r.t. the mesh node coordinates  $(x_i, y_i)$  and the state variables  $u$ .

The flux density at position  $i$  is computed as follows

$$\mathbf{B}_i = \begin{pmatrix} B_{xip} \\ B_{yip} \end{pmatrix} = \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix} \sum_j \nabla \alpha_j^i J_{T_i}^{-1} u_j \quad (\text{C.10})$$

where  $\nabla \alpha_j^i$  is the gradient of the base functions evaluated in the reference element and  $J_{T_i}$  is the jacobian of transformation from the reference element to the element  $T_i$  where the the point  $i$  is located.

Now, the derivatives of these quantities w.r.t. mesh node coordinates and state variables are detailed.

#### A Derivatives w.r.t mesh nodes

The derivatives  $\partial_{x_k} B_{xip}$ ,  $\partial_{y_k} B_{xip}$ ,  $\partial_{x_k} B_{yip}$  and  $\partial_{y_k} B_{yip}$  are easily computed since the dependence is only limited to the jacobian of transformation. Only the derivative of the jacobian  $J_T$  is needed to be computed and then put into the formulas.

#### B Derivatives w.r.t state variables

The derivative  $\partial_u \mathbf{B}_i$  is computed as follows

$$\partial_{u_k} \mathbf{B}_i = \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix} \sum_j \nabla \alpha_j^i J_{T_i}^{-1} \delta_{jk} \quad (\text{C.11})$$

such as  $\delta_{jk}$  is the Kronecker delta.

### C Calculation of gradient

The objective function is written as follows

$$W(p) = \sum_{i=1}^{10} (B_{xip}(p) - B_{xio})^2 + (B_{yip}(p) - B_{yio})^2 \quad (\text{C.12})$$

Its gradient is written as

$$\begin{aligned} d_p W &= \partial_p W + \lambda \partial_p \tilde{g} \\ &= \sum_{k=1}^N [\partial_{x_k} W + \lambda \partial_{x_k} \tilde{g}] d_p x_k \\ &\quad + \sum_{k=1}^N [\partial_{y_k} W + \lambda \partial_{y_k} \tilde{g}] d_p y_k \end{aligned} \quad (\text{C.13})$$

where

$$\begin{aligned} \partial_{x_k} W &= \sum_{i=1}^{10} 2(\partial_{x_k} B_{xip}(p))(B_{xip}(p) - B_{xio}) + 2(\partial_{x_k} B_{yip}(p))(B_{yip}(p) - B_{yio}) \\ \partial_{y_k} W &= \sum_{i=1}^{10} 2(\partial_{y_k} B_{xip}(p))(B_{xip}(p) - B_{xio}) + 2(\partial_{y_k} B_{yip}(p))(B_{yip}(p) - B_{yio}) \\ \lambda &= \left[ \sum_{i=1}^{10} 2(\partial_u B_{xip}(p))(B_{xip}(p) - B_{xio}) + 2(\partial_u B_{yip}(p))(B_{yip}(p) - B_{yio}) \right] (\partial_u \tilde{g})^{-1} \end{aligned}$$

## **Optimisation et analyse de fiabilité des machines électrique modélisées par la méthode des éléments finis**

La méthode des éléments finis est l'outil le plus sophistiqué pour modéliser le phénomène électromagnétique. Cependant, elle est coûteuse en temps de calcul. Ainsi, son utilisation pour l'optimisation et l'analyse de fiabilité (processus itératifs) doit être faite avec prudence car seul un nombre limité d'évaluations du modèle peut être toléré. De plus, l'impact du processus de fabrication sur les machines électriques est peu étudié dans la littérature. L'intégration de cet aspect dans la phase de conception est l'un des apports de cette thèse aux côtés de la principale contribution, qui est le développement et la comparaison des approches d'optimisation pour les machines électriques. Nous exposons les approches adaptées au sujet de la thèse et en développons de nouvelles. D'une part, le modèle d'éléments finis peut être considéré comme une "boîte noire" pour laquelle nous développons une approche non intrusive basée sur des méta-modèles. D'autre part, nous considérons une approche intrusive, nous améliorons le modèle pour fournir les dérivées des quantités d'intérêt. Les dérivées sont calculées efficacement en utilisant la méthode de la variable adjointe. Finalement, les méthodes sont comparées pour donner un aperçu des avantages et des inconvénients de chacune d'entre elles. Enfin, une étude de cas réel est abordée ; elle consiste à étudier l'impact du procédé de fabrication sur la machine à griffes fabriquée par Valeo. Sur la chaîne de production, les machines sont prélevées pour mesurer leurs dimensions et caractériser leur écart par rapport aux dimensions nominales. Ensuite, une analyse statistique est menée pour évaluer la fiabilité et l'impact sur les performances.

*Mots-clefs* : Machines électriques, Meta-modèles, Méthode de la variable adjointe, Propagation d'incertitudes, Méthode des éléments finis.

## **Optimization and Reliability Analysis for Electrical Machines modeled by Finite Element Method**

This dissertation deals with the approaches for optimization and reliability analysis of electrical machines modelled by the finite element method. The finite element method is the most sophisticated tool to model the electromagnetic phenomenon. However, it is computationally expensive. Thus, its usage for optimization and reliability analysis (iterative processes) should be made with caution since only a limited number of evaluations of the model can be tolerated. Furthermore, the impact of the manufacturing process on the electrical machines is scarcely studied in the literature. The integration of this aspect in the design phase is one of the contributions of this thesis alongside the main contribution, which is the development and comparison of optimization approaches for electrical machines.

We present the approaches adapted to the subject and develop new ones. On the one hand, the finite element model can be seen as a "black-box" for which we develop a non-intrusive approach based on Kriging meta-models. On the other hand, we consider an intrusive approach as we look inside the "black-box," we upgrade the model to provide the derivatives of the quantities of interest. The derivatives are essential to some optimization and reliability analysis tools. They are computed efficiently using the adjoint variable method. Finally, the methods are compared to give insight into the advantages and the shortcomings of each of them.

Lastly, a real case study is considered; it consists of studying the impact of the manufacturing process on a claw-pole machine manufactured by Valeo. From the production line, machines are withdrawn to measure their dimensions and characterize their deviation from the nominal one. Then a statistical analysis is conducted to assess the reliability and impact on the performances.

*Keywords* : Electrical machines, Meta-models, Adjoint variable method, Uncertainty propagation, Finite element method.