Coupling structural optimization and trajectory optimization methods in additive manufacturing

Mathilde Boissier

▶ **To cite this version:**

Mathilde Boissier. Coupling structural optimization and trajectory optimization methods in additive manufacturing. Mathematics [math]. Institut Polytechnique de Paris, 2020. English. NNT : . tel-04014391

HAL Id: tel-04014391
https://hal.science/tel-04014391

Submitted on 4 Mar 2023

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.

INSTITUT POLYTECHNIQUE DE PARIS

NNT : 2020IPPAX084

Thèse de doctorat

ÉCOLE POLYTECHNIQUE

IP PARIS

école normale supérieure paris–saclay

Coupling structural optimization and trajectory optimization methods in additive manufacturing

Thèse de doctorat de l'Institut Polytechnique de Paris
préparée à l'École Polytechnique

École doctorale n°574 Ecole Doctorale de Mathématiques Hadamard (EDMH)
Spécialité de doctorat : Mathématiques appliquées

Thèse présentée et soutenue à Palaiseau, le 8 décembre 2020, par

MATHILDE BOISSIER

Composition du Jury :

Grégoire ALLAIRE Professeur, École Polytechnique (CMAP)	Directeur de thèse
Éric CHARKALUK Directeur de recherche, École Polytechnique (LMS)	Examinateur
Jean-Yves HASCOËT Professeur, Centrales Nantes (GeM)	Président du jury
Alicia KIM Professeur, University of California San Diego (Structural Engineering Department)	Rapporteure
Serge NICAISE Professeur, Université Polytechnique des Hauts de France, (LAMAV)	Examinateur
Yannick PRIVAT Professeur, Université de Strasbourg (IRMA)	Rapporteur
Christophe TOURNIER Professeur, ENS Paris Saclay (LURPA)	Directeur de thèse
Gilles WALRAND Ingénieur, AddUp	Invité

ECOLE DOCTORALE DE MATHEMATIQUES HADAMARD

NNT : 2020IPPAX084

THÈSE DE DOCTORAT

de

L'Institut Polytechnique de Paris

École doctorale de mathématiques Hadamard (EDMH, ED 574)

Établissement d'inscription : École polytechnique

Laboratoire d'accueil : Centre de mathématiques appliquées de Polytechnique, UMR 7641 CNRS

Spécialité de doctorat : Mathématiques appliquées

Mathilde BOISSIER

Coupling structural optimization and trajectory optimization methods in additive manufacturing

Date de soutenance : 8 décembre 2020

Après avis des rapporteurs : Alicia KIM (University of California)
Yannick PRIVAT (Université de Strasbourg)

Jury de soutenance :

Grégoire ALLAIRE	(CMAP, École Polytechnique) Directeur de thèse
Eric CHARKALUK	(LMS, École Polytechnique) Examinateur
Jean-Yves HASCOËT	(GeM, Centrale Nantes) Président du jury
Alicia KIM	(University of California) Rapporteure
Serge NICAISE	(LAMAV, UP Hauts de France) Examinateur
Yannick PRIVAT	(IRMA, Université de Strasbourg) Rapporteur
Christophe TOURNIER	(LURPA, ENS Paris Saclay) Directeur de thèse
Gilles WALRAND	(AddUp) Invité

Benjamin,
Certains te disent
Imaginaire, fais leur la bise
Alors ils comprendront
Par tout ce pouvoir
Qu'ils ne peuvent pas voir
Qu'ils sont des ronchons

REMERCIEMENTS

Je n'ai jamais envisagé arrêter mes études avant la thèse. Je pensais alors que les sciences étaient une recherche de la vérité sûre, intransigeantes, innébranlables et rassurantes.
Ces trois ans ont ébranlé cette vision probablement enfantine et j'ai été à la fois transbahutée et emportée par la recherche. Perplexe, curieuse, frustrée, perdue, enthousiaste. Cet enchainement permanent d'états m'a fait grandir et des branches cassées de ma vision d'antan se sont développées de nouvelles tiges. Si elles sont intransigeantes, les sciences sont aussi pleines de liberté et s'en saisir est un des plus beaux et des plus difficiles enseignements. Si elles sont une recherche de la vérité, elles restent fragiles parce qu'elles reposent sur un délicat mélange de certitudes et de doutes. Elles nécessitent de faire des choix, de les assumer et de les remettre en question. D'avancer en prenant du recul. De valoriser l'instant et s'émerveiller au jour le jour de ce qu'on apprend. A la fois solides et menacées, parce qu'elles n'assènent pas mais proposent et progressent.
Pendant trois ans, dans un monde qui ne cesse de les trahir en parlant à leur place, j'ai découvert les sciences.

Un gigantesque merci à Christophe et Grégoire. Votre encadrement était à la fois bienveillance, soutien et confiance permettant au caneton chercheur d'évoluer. Tout en me confrontant en permanence à la rigueur scientifique et à la pertinence des hypothèses, vous m'avez ouvert un espace de création et appris à assumer les choix qui en résultent. Je voulais une thèse qui me permette de mener un projet "de bout en bout": du modèle et hypothèses à la formulation théorique puis à la résolution numérique du problème suivi d'une analyse des résultats, de recul, de la critique des choix faits et donc de la progression. En rédigeant, j'ai réalisé dans un étonnement heureux que cette thèse est exactement celle que je voulais mener. Grégoire, merci pour ton enthousiasme, pour tes impressionantes connaissances sur de multiples domaines scientifiques, pour ta disponibilité et ta patience nécessaires pour répondre à toutes mes questions (parfois même plusieurs fois la même question). Merci pour les discussions plus informelles sur le monde scientifique, la presse de pommes ou la montagne: elles ont été d'une importance capitale pour que le caneton pense de temps en temps à se poser. Enfin, un immense merci pour avoir construit une équipe de recherche dynamique au sein de laquelle il a fait bon travailler et se reposer. Christophe, merci pour les apports toujours pertinents, pour ces remarques que tu qualifies de "naïves" qui m'ont souvent éclairée dans la direction à suivre ou fait réaliser que le problème était certainement plus compliqué que ce que je pensais. Merci pour l'attention que tu as porté à cette thèse en veillant à ce qu'elle soit vraiment pluridisciplinaire.

Filant comme le vent la métaphore du canard, je tiens à remercier vivement les rapporteurs du manuscrit pondu. Yannick, d'abord en cours de master puis grâce aux conférences et articles, j'ai toujours été impressionnée par la qualité de tes travaux. Je suis très touchée que tu aies accepté de rapporter ce travail. Alicia, I met you several times at conference and I really appreciate your work. I hope your wonderful example will urge women towards maths so that, one day, it will be easy and common sense to have more than one woman in a PhD jury. Je tiens aussi à remercier Serge Nicaise, Eric Charkaluk, Jean-Yves Hascoet et Gilles Walrand qui ont accepté de faire partie du jury.

Lors de ces premiers envols, de mare en mare, j'ai été accueillie par différents labos. Les collaborations et discussions qui m'ont été proposées ont été une source d'ouverture sur de nouveaux domaines scientiques et sur de nouvelles visions de la recherche. Un grand merci à Tonia Maria Alam pour les multiples discussions et le travail en commun. J'ai énormément apprécié travailler avec toi et je suis triste que le confinement nous ait freinées dans cette collaboration. Merci aussi à Serge Nicaise et à l'équipe du LAMAV à Valenciennes pour leur accueil. I also would like to thank Natasha Vermaak and her team in Lehigh University. I liked the first experience in research I had in your lab in 2016: I signed up for a PhD. Thank you for your trust and I am glad we finally managed to finish the work we initiated there! I also want to warmly thank Joanne and Tom, my aunt and uncle in the United States, who welcomed me in Bethlehem in 2016 first and again during the three weeks I spent there in PhD. Enfin, un grand merci à Emmanuel Trélat qui a pris du temps pour me prodiguer un incroyable cours particulier appliqué à mon sujet en contrôle optimal.

A barboter tout seul, on ne prend pas son envol et on se noie dans son bol. Je voudrais remercier

chaleureusement l'équipe d'optimisation de forme (élargie) dans laquelle j'ai eu la chance d'évoluer. Merci infiniment Alex, à la fois encadrant officieux de ma thèse, mieux que Nocédal and Wright pour expliquer les algos d'optim, incroyable psychologue méditant. Cette thèse, et moi aussi, te devons bien plus que ce que tu ne veux reconnaître et j'espère te recroiser souvent. Merci Florian pour ton incroyable vision des sciences, pour avoir répondu à tant de questions, pour avoir développé cet algo bien mieux que le Lagrangian Augmenté mais surtout pour l'incroyable mélange de génie et d'humilité que tu dégages et qui m'a beaucoup poussée. Ton défaut réside peut être dans le fait que tu n'aimes pas la crème fraiche... Merci Perle pour m'avoir prise sous ton aile, toujours soutenue, pour avoir gardé un regard bienveillant et d'une incroyable pertinence sur mes travaux, pour ta vivacité, et pour ton magnanisme lors de mes crises de somnambulisme. Lalaina, merci pour ta persévérance tranquille qui m'a toujours impressionnée, pour ta bonne humeur et pour les conférences partagées (impliquant d'intéressants tests culinaires à Pékin). Merci Matias pour ta bonne humeur et la découverte du pebre, merci Jeet, Abdel pour toutes les discussions et pour avoir accepté de découvrir le ski avec nous, merci Martin pour ton aide récurrente sur la méthode inherent strain. Merci aussi aux permanents de cette équipe, Béni et Samuel, pour votre gentillesse et votre disponibilité, à Alexis pour m'avoir bien aidée avec le code et avoir accepté de le faire perdurer. Enfin, merci à l'équipe étendu, en particulier à Charles et Georgios qui ont toujours été un soutien bienveillant et efficace et aux membres de l'ANR SHAPO.

Avant de pouvoir voler le caneton palmé, pataud et bien souvent penaud patauge à petits pas. Heureusement, au CMAP, d'incroyables personnes m'ont sorti de la panade dans laquelle j'avais plongé. Un merci infini à Alexandra Noiret pour m'avoir aidée avec brio à résoudre mes soucis administratifs. Je n'ai jamais autant aimé gérer les papiers parce que ça me donnait des occasions de passer et de revenir avec le moral boosté. Merci à Nasséra pour son efficacité et sa patience pour mes innombrables questions. Merci de manière plus générale à toute l'équipe administrative du CMAP. Merci aussi à Aldjia qui, non seulement m'a donné d'excellents conseils d'escalade (faire grimper un caneton, ce n'était pas gagné non plus) mais qui en plus m'a appris à les rouages d'un contrat de travail en débloquant le cas pratique que le CNRS nous avait proposé. Enfin, essentiels dans ma sortie de la pataugeoire, merci à Pierre et Sylvain pour leur patiente assistance informatique et pour leurs réponses quasi instantanées à des bugs qui me paraissaient insolubles.
Je voudrais aussi remercier tous les membres du CMAP qui ont pris le temps d'échanger avec moi. Merci beaucoup Lucas pour toutes les discussions sur les sciences, la randonnée et pour m'avoir prêté tout un tas de bien chouettes bouquins. Merci Aline pour ce monitorat qui m'a beaucoup plu. Merci aussi aux quelques motivés, et en particulier à Lucas, Giovanni, Pierre et Solange, qui m'ont suivi dans la création d'un groupe de réflexion sur la transition écolo au labo. Je vous laisse avec pas mal de boulot mais au besoin, promis, au moins par mal je ne serai jamais très loin. Merci à Antonin, François, Flore, Teddy, Amandine, Anne, et tous les permanents qui, le temps d'une pause café, d'un passage par mon bureau, de trajets en RER, de foot, de conférences, de séminaires, ont pris le temps de discuter.
Dans cette mare, on était tout un tout un tas de canetons jouant et jacassant joyeusement dans la mangeoire et à tout événement notoire. Je tiens à remercier mes cobureaux: Florian pour les débats sur la montagne, le foot et les maths, Tristan pour m'avoir accueillie avec bienveillance au 2011, Marin pour avoir joué avec fair play la bataille engagée de l'invasion du bureau et de la possession du cable internet, Omar pour son attention et sa gentillesse, mais aussi Hugo, Mehdi, Jawad, Kaitong et Abdel. Merci beaucoup à Belhal pour sa bonne humeur à toute épreuve, Geneviève qui est pour moi un exemple de détermination tranquille, Céline pour toutes ces discussions de "pause", Martin pour le partage du monitorat et d'un grand nombre de débats, Léa que je retrouve avec plaisir depuis le début d'école, Louis, Jean-Bernard, Cheikh, Othmane, Rémi, Claire, Eugénie, Corentin, Clément, Benoît, Vianney, Juliette, Fédor, Antoine, Haytem, Josué, Kevish, Alejandro, Thomas et tous les autres.

Je voudrais aussi remercier les membres de LURPA, en particulier Kévin et Kamel pour vos retours sur mes travaux, pour les très bonnes pistes que vos travaux ont lancé et parce que j'avais toujours beaucoup de plaisir à vous retrouver aux différents évènements organisés par SOFIA. Merci aussi à Sylvain pour son attention et ses questions pertinentes et utiles.

Sortant de mes labos de recherche, dans un monde à plume, à poil et à écailles, je voudrais remercier tous les copains qui ont observé ces premières envolées avec beaucoup d'attention, tout à la fois m'encourageant et préparant au cas où matelas de duvet, casque et pansements pour soigner les potentiels bobos. De tous ceux qui ont contribué à cet environnement amusant et rassurant, un merci spécial à Adèle et Mappy pour les multiples discussions philo-cinémato-litérato-potino-coinche qui, m'ayant plus que soutenue m'ont

élevée. Merci aussi à Émil et Ludo, mes deux incroyables colocs, qui ont supportée (et pas mal nourrie aussi) mes premières années de thèse puis des séances de foot, des repas, un trajet à vélo... A propos de coloc, merci à Laura et Félix qui nous ont accueilli pour ce second confinement pendant lequel j'écris ces remerciements, nous permettant de passer un automne plein de chataignes, de champignons, de grand air, de karaokés, de bons repas et de rire. Merci aussi à Lorraine et Didier qui nous laissent avec beaucoup de gentillesse occuper leur maison. Merci merci merci merci merci merci merci (latex me souligne les répétitions ce petit malin) à Martin et Marine qui, d'un rendez-vous anticipé à un repas de dernière minute, sont présents depuis de nombreuses années pour discuter et franchement s'amuser. Un merci spécial à Frédéric, qui s'est occupé avec attention de ma culture pendant ces trois ans pour m'entrainer à ma plus grande joie entre expos, théatre et même concert, et à Mathilde que j'ai eu le plaisir de régulièrement voir à Paris. Un énorme merci à Antoine C, Adrien, Rémi, Fabien, Simon, Marc, Adèle, Hugo, Clémentine, Faustine, Pierre, Geoffroy, Luc, Raphaël, Sophie, Fabien, Jérome, Jordan, Momo, Charles, Noémie, Alex, Anaïs, Coco, Camille, Solène, Camille, Amélie, Maxence et à vous tous, qui, bienveillants et patients, avez toujours de l'indulgence pour le caneton pas toujours présent que je suis. Enfin, un grand merci à tous ces gens que j'ai rencontrés pendant cette thèse et qui m'ont tant fait réfléchir aux notions d'engagement, d'écologie et de société et en particulier à Alice et Mathilde avec qui on a finalement pas mal travaillé ainsi qu'à Antoine qui, avec beaucoup de volonté, de confiance et de réflexion a facilité mes premiers pas vers la suite. Enfin, un grand merci à Mme Quintard pour les inestimables outils qu'elle m'a offerts, pour les pistes de réflexion qu'elle a ouverte, pour m'avoir bousculée et soutenue dans ce passage à l'age adulte et pour ce formidable métier qu'elle fait.

Comme tout caneton, je proviens d'un nid que mes cAnards de parents ont construit, couvé, avant de doucement nous laisser en partir. Maman et Papa, merci. Merci pour l'enfance innocente et joyeuse que vous avez permise, pour votre confiance immense qui m'a été tant utile et qui s'apparente même parfois à de la foi. Merci pour avoir toujours accepté, encouragé, initié les discussions sur d'innombrables sujets, pour m'avoir challengée, réconfortée, faite grandir et accepté de grandir avec moi. Merci aussi pour mes deux acoyltes à col verts, Noémie et Guillemette. A Numéro 1 et Numéro 3, merci d'être tout à la fois soutien inconditionnel, source d'admiration et donc de motivation, équipières parfaites pour tous les jeux d'association d'idée et source d'inspiration pour des idées d'associations. A Sylvain, merci d'avoir si joyeusement rejoint la bande et d'accepter avec tant d'humour et de bienveillance nos traits de caractères parfois marqués. Enfin un grand merci à mes grand parents, si présents à mon esprit en particulier en cette fin de thèse, pour une initiation qui prend toujours plus de sens à la nature, à la terre et à la joie d'être dehors au petit matin.

Il me reste un canard particulier à remercier, avec beaucoup d'affection, d'admiration et de respect. Merci pour les innombrables discussions scientifiques qui ont beaucoup fait évoluer cette thèse (et pour les références sur l'algo proximal), merci pour les innombrables débats de société qui me font réfléchir et évoluer, merci pour cette honneteté et ce recul qui me poussent à assumer mes choix et à les porter, merci pour ce soutien et cette confiance qui me poussent à agir, merci pour les rires, les randos à pieds, à skis, les jeux et toute cette complicité qui rend le quotidien lumineux. Pour tout cela et pour bien plus, Hadrien, merci!

Il faut finalement citer ces quelques fous qui ont avec beaucoup d'abnégation et de gentilesse accepté de relire des bouts de ce manuscrit: Adèle, Mappy, Alex, Guigui, Papa et Hadrien, merci!

ABSTRACT

This work investigates path planning optimization for powder bed fusion additive manufacturing processes, and relates them to the design of the built part. The state of the art mainly studies trajectories based on existing patterns and, besides their mechanical evaluation, their relevance has not been related to the object's shape. We propose in this work a systematic approach to optimize the path without any a priori restriction. The typical optimization problem is to melt the desired structure, without overheating (to avoid thermally induced residual stresses) and possibly with a minimal path length. The state equation is the heat equation with a source term depending on the scanning path. Two physical 2-d models are proposed, involving temperature constraint: a transient and a steady state one (in which time dependence is removed). Based on shape optimization for the steady state model and control for the transient model, path optimization algorithms are developed. Numerical results are then performed allowing a critical assessment of the choices we made. To increase the path design freedom, we modify the steady state algorithm to introduce path splits. Two methods are compared. In the first one, the source power is added to the optimization variables and an algorithm mixing relaxation-penalization techniques and the control of the total variation is set. In a second method, notion of topological derivative are applied to the path to cleverly remove and add pieces. eventually, in the steady state, we conduct a concurrent optimization of the part's shape and of the scanning path. This multiphysics optimization problem raises perspectives gathering direct applications and future generalizations.

Keywords: Path planning and control, additive manufacturing, metallic powder bed fusion, structural optimization.

RÉSUMÉ

Cette thèse porte sur l'optimisation des trajectoires de lasage pour la fabrication additive sur lit de poudre, ainsi que leur lien avec la géométrie de la pièce à construire. L'état de l'art est principalement constitué par des trajectoires basées sur des motifs, dont l'impact sur les propriétés mécaniques des objets finaux est quantifié. Cependant, peu d'analyses permettent de relier leur pertinence à la forme de la pièce elle-même. Nous proposons dans ce travail une approche systématique visant à optimiser la trajectoire sans restriction a priori. Le problème d'optimisation consiste à fusionner la structure en évitant de surchauffer (ce qui induirait des contraintes résiduelles) tout en minimisant le temps de fabrication. L'équation d'état est donc l'équation de la chaleur, dont le terme source dépend de la trajectoire. Deux modèles 2-d sont proposés pour contrôler la température : l'un transitoire et le second stationnaire (pas de dépendance en temps). Basés sur des techniques d'optimisation de forme pour le stationnaire et sur des outils de contrôle pour le transitoire, des algorithmes d'optimisation sont développés. Les applications numériques qui en découlent permettent une analyse critique des différents choix effectués. Afin de laisser plus de liberté dans la conception, l'algorithme stationnaire est adapté à la modification du nombre de composantes connexes de la trajectoire lors de l'optimisation. Deux méthodes sont comparées. Dans la première, la puissance de la source est ajoutée aux variables d'optimisation et un algorithme impliquant une relaxation-pénalisation et un contrôle de la variation totale est proposé. Dans la seconde, la notion de dérivation topologique est adaptée à la source. Enfin, dans le cadre stationnaire, nous détaillons le couplage de l'optimisation de la forme de la pièce, pour améliorer ses performances mécaniques, et de la trajectoire de lasage. Ce problème multiphysique ouvre des perspectives d'applications et de généralisations futures.

Mots clés : Génération et contrôle de trajectoires, fabrication additive, fusion sur lit de poudre métallique, optimisation de forme.

CONTENTS

1 POWDER BED FUSION ADDITIVE MANUFACTURING CONTEXT 29
1.1 Introduction . . . 29
1.2 Powder Bed Fusion (PBF) additive manufacturing process . . . 30
1.3 Path optimization in the PBF context . . . 36
1.4 Design to manufacturing . . . 43
1.5 Problematic of this work . . . 43

2 ALGORITHMS FOR CONSTRAINED OPTIMIZATION 45
2.1 Introduction . . . 45
2.2 Descent methods . . . 46
2.3 Unconstrained first order algorithms . . . 50
2.4 Constrained first order algorithms . . . 51
2.5 Unconstrained second order algorithms . . . 59
2.6 Non differentiable optimization . . . 60
2.7 Conclusion . . . 65

3 SHAPE OPTIMIZATION 67
3.1 Introduction . . . 67
3.2 Shape differentiation . . . 67
3.3 Shape optimization of a domain . . . 73
3.4 Conclusion . . . 77

4 PROBLEM STATEMENT: MODEL AND OBJECTIVES 79
4.1 Introduction . . . 79
4.2 Transient model presentation: hypothesis, description and optimization problem . . . 79
4.3 Model calibration . . . 84
4.4 Steady model presentation . . . 91
4.5 Conclusion . . . 95

5 PATH DISCRETIZATION 97
5.1 Introduction . . . 97
5.2 Working domain and path meshing . . . 98
5.3 Path discretization in the steady state context . . . 100
5.4 Path discretization in the transient context - using angles . . . 108
5.5 Path discretization in the transient context - using points . . . 109
5.6 Conclusion . . . 110

6 PATH OPTIMIZATION IN THE STEADY STATE CONTEXT 111
6.1 Introduction . . . 111
6.2 Steady path optimization problem . . . 112
6.3 Shape derivative computation . . . 112
6.4 Numerical algorithms . . . 114
6.5 Numerical results . . . 116
6.6 Initialization and geometry to build impact for the aluminium . . . 130
6.7 Initialization and geometry to build impact for the titanium . . . 137
6.8 Conclusion . . . 144

7 PATH OPTIMIZATION IN THE TRANSIENT CONTEXT 145
7.1 Introduction . . . 145
7.2 Path optimization based on an angle parametrization . . . 145
7.3 Path optimization based on a points parametrization . . . 158
7.4 Conclusion . . . 178

8 MODIFICATION OF THE PATH'S TOPOLOGY 179
8.1 Introduction . . . 179
8.2 Controlling the power . . . 180
8.3 Application of the power control in the steady state context . . . 184
8.4 Topology optimization of the path (steady state context) . . . 198

8.5 Further comparisons between the coupled power and path optimization and the topology optimization methods . . . 214
8.6 Conclusion . . . 228

9 SHAPE AND PATH COUPLED OPTIMIZATION IN THE STEADY STATE CONTEXT 231
9.1 Introduction . . . 231
9.2 Settings of the optimization problem and differentiation . . . 231
9.3 Numerical algorithm . . . 233
9.4 Numerical results . . . 236
9.5 Conclusion . . . 249

Bibliography 257

NOMENCLATURE

TITANIUM

Physical parameters

density	$\rho = 4420 kgm^{-3}$
specific heat	$c_p = 800 JK^{-1}kg^{-1}$
conductivity	$\lambda = 15 Wm^{-1}K^{-1}$
change of phase temperature	$y_\phi = 1900 K$

Process parameters

source power	$P = 300 W$
source radius	$r = 5.0\,10^{-5} m$
initial temperature	$y_{\text{ini}} = 773 K$

ALUMINUM

Physical parameters

density	$\rho = 2680 kgm^{-3}$
specific heat	$c_p = 800 JK^{-1}kg^{-1}$
conductivity	$\lambda = 130 Wm^{-1}K^{-1}$
change of phase temperature	$y_\phi = 870 K$

Process parameters

source power	$P = 400 W$
source radius	$r = 5.0\,10^{-5} m$
initial temperature	$y_{\text{ini}} = 773 K$

INTRODUCTION

La fabrication additive par fusion sur lit de poudre métallique est une technologie prometteuse, consistant à construire une pièce métallique couche par couche [87]. Pour chacune d'entre elles, un lit de poudre est déposé sur la surface supérieure de l'objet en cours d'impression. Cette poudre est alors fusionnée au passage d'une source d'énergie déplacée le long d'une trajectoire spécifique (trajectoire de lasage). Une phase de refroidissement permet finalement la solidification du métal (Figure 1). Les avantages liés à cet outil de production sont multiples avec, entre autres, la fabrication de pièces trop complexes pour les technologies traditionnelles ou encore la rentabilité de la fabrication de pièces uniques sans nécessité de production en série. Ces atouts rendent majeurs les enjeux d'amélioration de ce procédé [68, 142]. On distingue deux problématiques principales. La première concerne l'efficacité de la fabrication. Chaque couche requiert un temps de traitement qui, s'il est restreint, entraîne une nette réduction du temps total de construction. La seconde concerne la qualité de la pièce finale. De multiples phénomènes physiques et mécaniques sont impliqués par le processus de fabrication, du changement de phase de la poudre à des mouvements de fluide complexes. Mal contrôlés, ces phénomènes sont à l'origine de fragilités qui peuvent provoquer l'invalidité de la pièce (Figure 2).

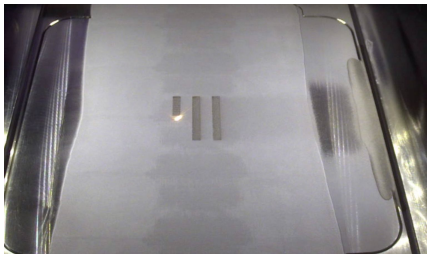


Figure 1: Fabrication d'une pièce

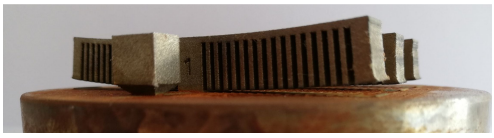
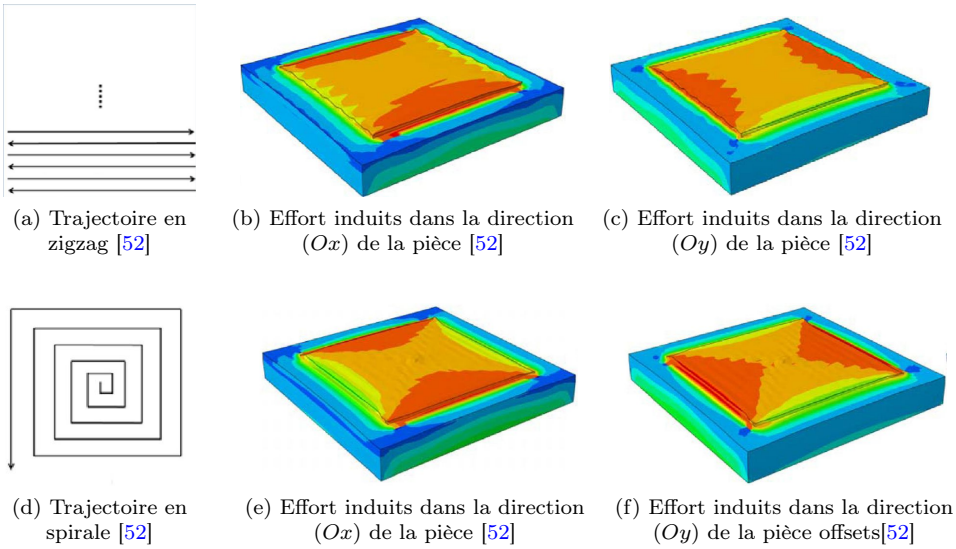


Figure 2: Invalidité de la pièce : quand on coupe le support, les extrémités de la pièce remontent au lieu de rester dans l'axe à cause des contraintes résiduelles.

Si de multiples paramètres peuvent permettre l'amélioration de ces enjeux [172], nous choisissons dans cette thèse de considérer en particulier la trajectoire de lasage, correspondant au chemin parcouru par la source afin de fusionner chaque couche de poudre. Le choix de trajectoire des outils est un enjeu récurrent de la production automatisée, que l'on retrouve notamment en usinage ou en soudure. Ce choix joue à la fois sur la productivité et sur la qualité du produit final. Si la puissance de la source l'impacte, le temps de construction est principalement relié aux propriétés cinématiques du système et notamment par la capacité des différents actionneurs à parcourir précisément et rapidement la trajectoire choisie [92]. La qualité, en fabrication additive sur lit de poudre, est quant à elle dictée par les phénomènes mécaniques et physiques engendrés par les multiples changements de phase [68]. La planification du passage de la source est donc cruciale [77]. Elle détermine la quantité d'énergie fournie à la poudre à chaque instant et en chaque point. Une mauvaise répartition de la chaleur entraine de la porosité, de l'évaporation de matière ou encore l'introduction de contraintes résiduelles qui fragilisent la pièce [52, 104] (Figure 3).

Nous décidons dans cette thèse d'ajouter au travail sur la technologie elle-même un travail sur la pièce à construire. En effet, si le procédé de fabrication est l'un des acteurs de la fabrication, l'objet et en particulier sa géométrie, en est un second. L'optimisation de forme, initiée par Hadamard en 1908 [95], consiste à concevoir des pièces optimisées selon des critères physiques, mécaniques ou géométriques, mêlant ainsi de multiples domaines théoriques (contrôle, calcul des variations, optimisation) et numériques (résolution d'équation différentielles, représentation numérique des objets, application d'algorithmes d'optimisation) des mathématiques appliquées [9, 15, 97]. Ce domaine a trouvé un nouvel essor avec le développement de l'impression 3D. En effet, l'optimisation induit dans la majeure partie des cas des géométries trop complexes pour être construites par des procédés traditionnels (fonderie, usinage). Les nouvelles possibilités apportées par la fabrication additive rendent accessible la réalisation de pièces "plus optimales" que ce qui pouvait être fait dans le passé. A l'inverse, l'optimisation de forme est aussi souvent au service du moyen de production. En effet, les développements récents prennent de plus en plus en compte des contraintes de "fabricabilité" dans l'optimisation. Parmi les récents travaux à ce sujet, on citera par exemple l'optimisation de la pièce et des supports pour répondre aux contraintes de supportage lors de la fabrication [11, 12, 13, 14] (Figure 4) ou encore l'optimisation de la pièce pour limiter les défauts mécaniques après fabrication [16] (Figure 5).



(a) Trajectoire en zigzag [52]

(b) Effort induits dans la direction (Ox) de la pièce [52]

(c) Effort induits dans la direction (Oy) de la pièce [52]

(d) Trajectoire en spirale [52]

(e) Effort induits dans la direction (Ox) de la pièce [52]

(f) Effort induits dans la direction (Oy) de la pièce offsets[52]

Figure 3: Trajectoires en zigzag et spirale et efforts résiduels induits

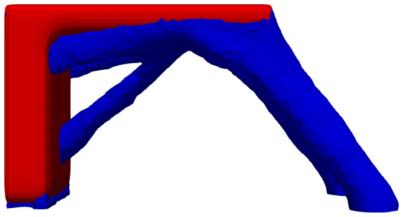


Figure 4: Support de fabrication (bleu) pour la pièce (rouge) [11]

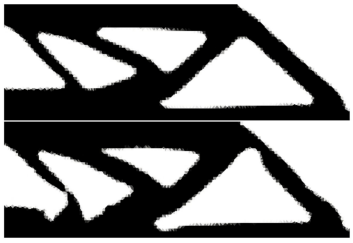


Figure 5: Minimisation de l'effort élastique maximal sans thermique (haut) et avec thermique (bas) [16]

Dans cette thèse, nous défendons la possibilité d'intégrer les contraintes de productivité et de qualité liées aux trajectoires de lasage dans la conception de la pièce elle-même. L'optimisation de sa forme permettrait alors à l'utilisateur de choisir un compromis entre l'efficacité de l'objet final et la qualité et rapidité de sa fabrication. Il s'agit donc de formuler les contraintes de pilotage du système et de physique du procédé en une contrainte mathématique intégrable dans un problème d'optimisation de formes.
Avant l'optimisation de la forme en elle-même, les contraintes que la trajectoire impose sur la forme doivent être décrites et formulées de façon à être intégrées à un code d'optimisation. Une première question implique une étude approfondie afin de développer une notion de "bonne trajectoire". On trouve dans la littérature de multiples travaux évaluant les trajectoires actuellement utilisées. Basées sur des stratégies précises (zigzag, offset du bord de la pièce, fractales), des études expérimentales et numériques permettent de déterminer leurs points forts et leurs points faibles. Cependant, évaluer a priori l'effet de cette trajectoire reste une question peu abordée. Une seconde question concerne l'impact de la géométrie de la pièce à construire sur la trajectoire. Dans les applications industrielles actuelles, la stratégie de lasage est dans la majeure partie des cas déterminée indépendamment de l'objet et on peut donc évaluer la pertinence d'une trajectoire pour un objet en particulier. En revanche, seuls quelques très récents travaux considèrent un couplage entre forme et trajectoire [6, 7, 51].

Afin de pallier le manque d'information sur la notion de bonne trajectoire et sur son évolution avec les changements de géométries des pièces, nous avons choisi dans cette thèse de développer un algorithme d'optimisation de la trajectoire basé sur des outils mathématiques, sans choix de stratégie a priori. Les résultats obtenus dans de multiples situations et en particulier pour différentes géométries ou par couplage avec un algorithme d'optimisation de forme sont dans un second temps exploités pour la formulation de contraintes de conception.

Ce type d'étude est encore peu présent dans la littérature (à notre connaissance, seuls [6, 7, 51] se sont intéressés à cette problématique) nous forçant à faire un certain nombre de choix tout au long de ce travail. Il s'agit tout d'abord de poser un cadre physique et de construire un problème d'optimisation de trajectoire. Ce cadre requiert de multiples hypothèses, afin de permettre une résolution à la fois pertinente et d'un coût de calcul faible. Limiter les phénomènes facilite de plus l'analyse des résultats et donc la prise de recul. Nous avons ensuite construit une représentation numérique du problème ce qui nous a permis de développer différents algorithmes d'optimisation de la trajectoire. Grâce aux résultats obtenus, nous avons analysé les différents choix de modélisation, discrétisation ou encore algorithme faits tout au long de ce travail. Nous complexifions ensuite le modèle pour autoriser les changements de topologie de la trajectoire avant, enfin, de coupler l'algorithme d'optimisation de trajectoires à l'optimisation de la forme à construire.

Ce travail mêlant production automatisée et mathématiques appliquées a été réalisé dans le cadre du projet SOFIA (SOlution pour la Fabrication Industrielle Additive métallique), programme initié en 2016 par la co-entreprise AddUp visant à proposer une solution de fabrication additive métallique par procédé de type lit de poudre. Ce projet a pour objectif l'amélioration des technologies existantes au moyen d'une étude poussée de l'intégralité du processus de fabrication des pièces. Afin de réunir une diversité de compétence, un consortium a été créé (Figure 6) réunissant des partenaires industriels (Aubert & Duval, ESI Group, FUSIA, Michelin, Safran, Volum-e, Zodiac Aerospace) et académiques (CNRS et établissements de recherche et d'enseignement avec en particulier CentraleSupelec, Centrale Nantes, École Polytechnique, ENS Paris-Saclay, ainsi que l'Université Paris Diderot, l'Université Paris-Sud, Université Pierre et Marie Curie – Paris Sorbonne Université).

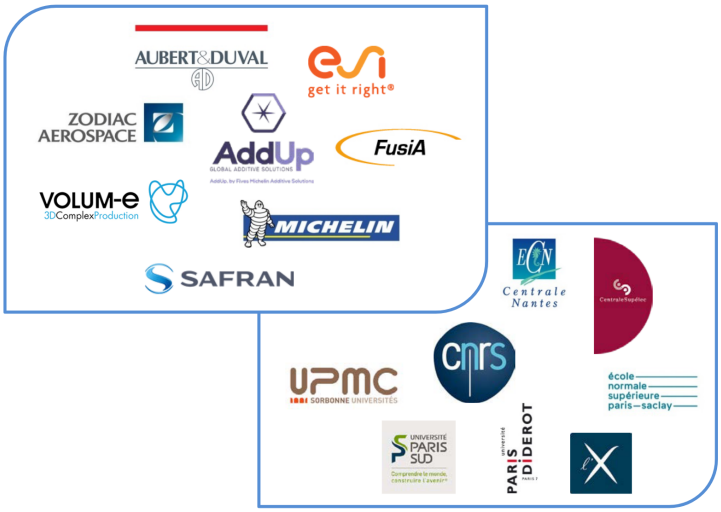


Figure 6: SOFIA project consortium

Cette thèse, portée par le CMAP et le LURPA, se situe pleinement dans l'axe numérique de ce projet, et plus particulièrement à la frontière entre optimisation du design des pièces et trajectoires de lasage.

Structure du manuscrit

Ce manuscrit s'articule en deux parties. Dans la première, nous introduisons le contexte de la thèse avec trois chapitres d'état de l'art respectivement sur la fabrication additive par fusion sur lit de poudre (Chapitre 1), l'optimisation sous contrainte (Chapitre 2) et l'optimisation de forme (Chapitre 3). La deuxième partie est consacrée aux contributions de cette thèse.

Partie 1

Nous introduisons au Chapitre 1, la fabrication additive par fusion sur lit de poudre. Après une brève description du procédé, les enjeux majeurs sont exposés en deux problématiques majeures : comprendre les phénomènes que ce soit par des caractérisations expérimentales ou une modélisation précise du procédé de fabrication, et améliorer la technologie. Nous détaillons ensuite la littérature relative aux trajectoires de lasage. Cette mise en contexte appuie la problématique de ce travail et le besoin de

développer l'optimisation de la trajectoire de lasage.

Cette thèse ayant pour objet l'optimisation de la trajectoire et de la forme de la pièce, il est nécessaire d'introduire différents algorithmes d'optimisation sous contrainte. Ce domaine est extrêmement large : chaque application requiert des spécificités. Loin d'être exhaustifs, nous nous limitons dans le Chapitre 2 à l'introduction des méthodes de descente qui seront utilisés dans ce travail, ainsi que de la théorie sur lesquels ils sont basés.

Au Chapitre 3, nous détaillons la théorie d'optimisation de forme, en précisant la notion de dérivée par rapport à une courbe et la notion de gradient correspondant. Pour terminer, un exemple classique d'optimisation d'une structure dans le cadre de l'élasticité linéaire est développé avec l'introduction de la représentation de forme par lignes de niveaux et une illustration par quelques résultats numériques. Cette théorie nous sera utile à plusieurs reprises par la suite : pour l'optimisation de la trajectoire dans un cadre spécifique et pour le couplage entre optimisation de forme et de trajectoire mentionné dans les objectifs de cette thèse.

Le contexte dressé dans le Chapitre 1 et les notions introduites dans les Chapitres 2 et 3 sont présents tout au long des contributions que nous détaillons dans la seconde partie.

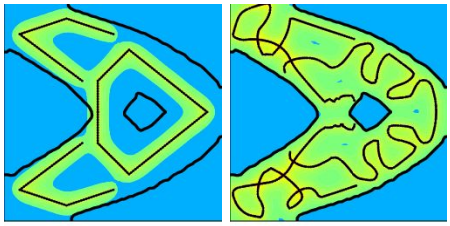
Partie 2

Le Chapitre 4 est consacré au modèle utilisé dans le reste de ce travail et au problème d'optimisation de trajectoire choisi. Il nous est impossible de considérer ici une modélisation précise de toute la physique impliquée dans le processus de fabrication: les phénomènes trop complexes induisent un coût de calcul important. Nous présentons dans ce chapitre le compromis utilisé.
Nous choisissons une focalisation sur la thermique uniquement, caractérisée par une équation de la chaleur en deux dimensions n'impliquant que la conduction, dont le terme source parcourt la trajectoire à vitesse constante. Les contraintes de trajectoire sont prises en compte par la minimisation du temps de parcours de la trajectoire (minimisation du temps de fabrication) et des contraintes sur la température (pour maîtriser la qualité finale de la pièce). Afin de pallier ces simplifications et de permettre malgré tout une analyse physique pertinente des résultats, nous calibrons le modèle obtenu avec des données provenant de modèles plus complets.
Un second modèle est enfin décrit, dans lequel la dépendance en temps est abandonnée. On ne considère plus de mouvement le long de la trajectoire mais au contraire que la source est appliquée brutalement sur l'intégralité de la ligne. Nous calibrons ce second modèle de manière à ce que la température simulée corresponde au maximum de température obtenue lors de la simulation du modèle transitoire. Ce modèle stationnaire est évidemment un modèle jouet. Il permet néanmoins la mise en place d'algorithmes d'optimisation et un premier recul à moindre coût sur les difficultés qui seront rencontrées pour le modèle transitoire.

Le modèle proposé dans le Chapitre 4 nécessite d'être discrétisé pour pouvoir être numériquement résolu. Nous présentons dans le Chapitre 5 les choix numériques pour permettre la simulation. Nous optons pour une méthode de "front-tracking", dans laquelle un maillage du domaine de calcul est fixé pendant toutes l'optimisation et la trajectoire est discrétisée par une ligne brisée, modifiée à chaque itération.
Dans le cadre stationnaire, les noeuds de cette ligne brisée sont les variables d'optimisation: ils sont donc modifiés à chaque itération. Cette stratégie nécessite des adaptations numériques : la distance entre chaque noeud doit être maîtrisée et une étape de rediscrétisation de la ligne à chaque itération est nécessaire, le calcul de la normale et de la courbure le long de la ligne doit être précisé, un transfert d'informations entre la ligne brisée et le maillage physique doit être instauré et la différentiation et intégration le long de la ligne doivent être décrits.
Dans le cadre transitoire, nous détaillons deux choix différents de contrôle de cette ligne. Dans le premier cas, on considère la distance entre les noeuds fixe. Les variables d'optimisation sont alors la position du premier point, le nombre de points total ainsi que l'angle formé par chaque segment de la ligne brisée avec l'axe horizontal. Dans le second cas, l'approche correspond au cas stationnaire avec contrôle par les points eux-mêmes. La distance entre deux noeuds consécutifs n'est plus fixe et une rediscrétisation de la ligne est nécessaire à chaque itération.

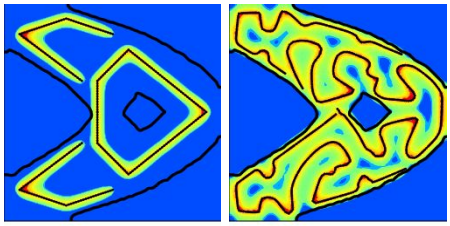
Dans le Chapitre 6, nous entrons dans le vif du sujet avec l'optimisation de la trajectoire dans le cas

stationnaire. Le modèle proposé par le Chapitre 4 suivi des choix numériques détaillés par le Chapitre 5 nous permettent la mise en place d'un algorithme d'optimisation. Dans le contexte stationnaire, optimiser la trajectoire consiste à optimiser la forme d'une courbe qui porte la source de chaleur. Nous pouvons donc utiliser les outils d'optimisation de forme présentés au Chapitre 3. Une première partie consiste à déterminer la dérivée de forme du problème avant de la transformer en un gradient discret, défini en chaque noeud de la ligne brisée. De là, deux algorithmes d'optimisation sous contrainte d'ordre 1, décrits dans le Chapitre 2 peuvent être mis en place.
Les premiers résultats numériques permettant d'acter le bon comportement de l'algorithme, nous menons une analyse critique des différents choix pris au cours des étapes successives: choix de discrétisation, d'optimisation, de modèle physique. Enfin, nous lançons une série de tests partant de multiples initialisations pour la fabrication de différentes géométries. Les nombreux résultats obtenus permettent une première prise de recul sur la pertinence de la méthode d'optimisation mise en place tout en alimentant l'intuition de la notion de "bonne trajectoire" (Figures 7 and 8).



(a) Trajectoire initiale (b) Trajectoire finale

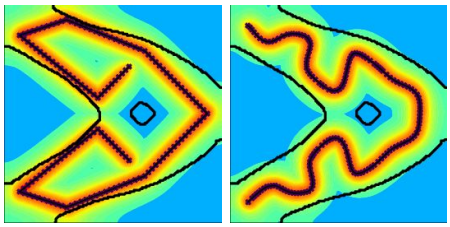
Figure 7: Trajectoire pour la fusion d'un objet en aluminium, modèle stationnaire (bleu: pas de fusion, rouge: température trop élevée)



(a) Trajectoire initiale (b) Trajectoire finale

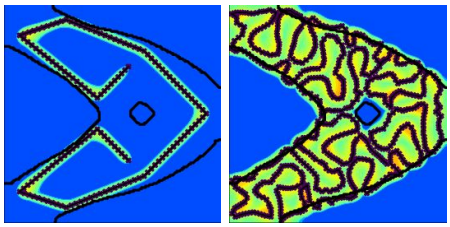
Figure 8: Trajectoire pour la fusion d'un objet en titane, modèle stationnaire (bleu: pas de fusion, rouge: température trop élevée)

Délaissant le modèle jouet stationnaire, on se concentre dans le Chapitre 7 sur l'optimisation de la trajectoire pour le modèle transitoire. Nous considérons dans un premier temps la paramétrisation de la courbe par les angles. La dérivation par rapport aux variables d'optimisation permet la mise en place d'un algorithme d'optimisation. Cependant, les premiers résultats mettent en avant les défauts de cette paramétrisation. La gestion du temps nécessaire pour que la source parcoure la trajectoire (correspondant à la longueur de la trajectoire sous hypothèse de vitesse de parcours constante) est compliquée.
Nous testons alors la seconde paramétrisation, dans laquelle les noeuds eux-mêmes sont la variable d'optimisation. La distance entre les noeuds étant maintenant autorisée à varier, le temps de calcul est fortement augmenté. Cependant, les résultats sont bien meilleurs. Tout comme dans le cas stationnaire, une analyse des paramètres de discrétisation et des algorithmes d'optimisation est menée. Nous lançons enfin une batterie de tests pour tester l'impact des initialisations et de la géométrie à construire sur la trajectoire finale (Figures 9 and 10).



(a) Trajectoire initiale (b) Trajectoire finale

Figure 9: Trajectoire pour la fusion d'un objet en aluminium, modèle transitoire (bleu: pas de fusion, rouge: température trop élevée)

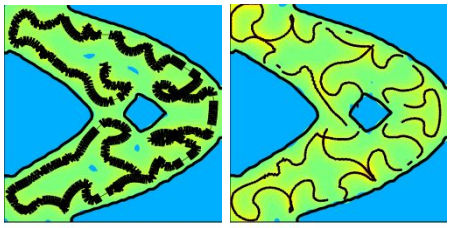


(a) Trajectoire initiale (b) Trajectoire finale

Figure 10: Trajectoire pour la fusion d'un objet en titane, modèle transitoire (bleu: pas de fusion, rouge: température trop élevée)

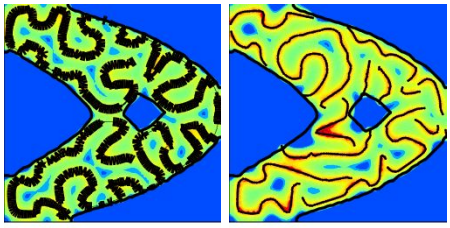
Nous cherchons dans le Chapitre 8 à complexifier le modèle. Afin de laisser plus de liberté dans la détermination de la trajectoire, on autorise la modification du nombre de composantes connexes la

constituant. Deux méthodes sont mises en place et testées dans le cadre stationnaire.
La première est basée sur la physique du procédé et fait l'objet d'une collaboration avec Tonia Maria ALAM (LAMAV, UPHF). Couper la trajectoire correspond physiquement à éteindre puis rallumer la source d'énergie. La puissance devient une variable d'optimisation supplémentaire, autorisée uniquement à prendre les valeurs extrêmes 0 et $P_{\max}$, où $P_{\max}$ est la puissance de la source lorsqu'elle est allumée (les technologies actuelles ne permettent pas de variations continues de puissance; il faudrait pour cela modifier la vitesse de parcours ce qui reste une perspective de ce travail). Afin de permettre une telle optimisation, nous utilisons une méthode de relaxation-pénalisation. Enfin, pour éviter l'obtention de trajectoires dégénérées qui nécessiteraient un nombre trop important d'allumage de la source, un contrôle des sauts de puissance est ajouté. Cette technique permet l'obtention de nouveaux résultats numériques. Si le chapitre est restreint au cadre stationnaire, cette technique pourrait être adaptée au cadre transitoire.
La seconde méthode est spécifique au stationnaire. Elle consiste à utiliser la notion de dérivée topologique [54, 153, 164, 175]. Principalement développée en deux ou trois dimensions pour changer la topologie du domaine optimisé, cette notion existe aussi en une dimension, notamment pour la caractérisation de fissures. On l'adapte ici à la trajectoire stationnaire. Nous modifions l'algorithme d'optimisation du Chapitre 6 pour inclure la séparation de composantes connexes de la trajectoire ou l'ajout d'une nouvelle. Des résultats numériques complètent cette étude, dont une comparaison pour différentes géométries des deux approches détaillées (Figures 11 and 12).



(a) Méthode basée sur la puissance (b) Utilisation du gradient topologique

Figure 11: Trajectoire pour la fusion d'un objet en aluminium, modèle stationnaire - modification du nombre de composantes connexes (bleu : pas de fusion, rouge : température trop élevée)



(a) Méthode basée sur la puissance (b) Utilisation du gradient topologique

Figure 12: Trajectoire pour la fusion d'un objet en titane, modèle stationnaire - modification du nombre de composantes connexes (bleu : pas de fusion, rouge : température trop élevée)

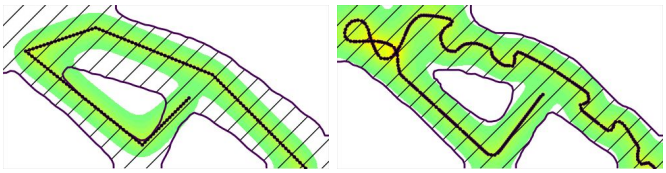
Enfin, afin d'étudier plus précisément l'impact des formes sur les trajectoires, le Chapitre 9 est consacré à l'optimisation couplée de la pièce à construire et de la trajectoire de lasage. On considère le design d'une pièce dont on veut minimiser la compliance sous contrainte de volume maximal. Cette optimisation de forme dans le cadre de l'élasticité linéaire est classique. On ajoute ici, dans le cadre stationnaire uniquement, la détermination de la trajectoire de lasage correspondante. Les résultats obtenus dans ce chapitre restent préliminaires. Néanmoins, ils permettent de mettre en avant la modification de la géométrie lors de l'ajout des contraintes de trajectoires (Figures 13 et 14). L'analyse de ces modifications permet finalement d'obtenir de l'intuition sur les contraintes de conception et confirment la pertinence de l'approche choisie pour pousser plus loin cette étude.

Communications et publications scientifiques : ce travail a donné lieu aux communications suivantes:

Articles de journaux

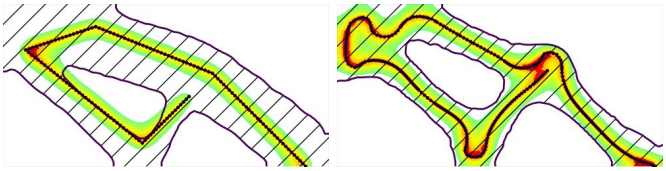
- **Scanning path optimization using shape optimization tools**, M. Boissier, G. Allaire, C. Tournier, Structural and Multidisciplinary Optimization, 61:6, pp. 2437-2466, 2020 (récompensé par le prix ISSMO/Springer 2019)

- futures publications en préparation



(a) Optimisation de la forme sans trajectoire (b) Optimisation de la forme et de la trajectoire

Figure 13: Comparaison des résultats d'optimisation de forme seule et de forme et de trajectoire pour le design d'une pièce en aluminium, modèle stationnaire (pièce hachurée, en blanc : pas de fusion, rouge : température trop élevée)



(a) Optimisation de la forme sans trajectoire (b) Optimisation de la forme et de la trajectoire

Figure 14: Comparaison des résultats d'optimisation de forme seule et de forme et de trajectoire pour le design d'une pièce en titane, modèle stationnaire (pièce hachurée, en blanc : pas de fusion, rouge : température trop élevée)

Communications orales

- **Optimisation de trajectoire pour le procédé de fabrication Laser Powder Bed Fusion (LPBF)**, CSMA 2019 (Mai 2019), Giens, France
- **Laser path optimization for additive manufacturing**, WCSMO (Mai 2019), Beijing, Chine
- **Laser path optimization using shape optimization tools for the Laser Powder Bed Fusion process**, SimAm 2019 (Septembre 2019), Pavie, Italie

Séminaires

- **Simulation thermique et optimization de trajectoire pour la fabrication additive métallique par le procédé Selective Laser Melting (SLM)**, journées de mi-projet SOFIA (Avril 2019), Clermont-Ferrand, France
- **Utilisation de l'optimisation de forme pour la détermination de trajectoires de lasage**, séminaire FAPS (Février 2020), Palaiseau, France

Posters

- **Couplage de méthodes d'optimisation topologique de formes et d'optimisation de trajectoires en fabrication additive**, CANUM 2018 (Mai 2018), Cap d'Agde, France
- **Path optimization for the laser powder bed fusion additive manufacturing process**, New trends and challenges in the mathematics of optimal design, Cambridge, 2019
- **Path optimization for the laser powder bed fusion additive manufacturing process**, New Trends in PDE Constrained Optimization, Linz 2019

Ainsi que

Article de journal

- **Elastoplastic topology optimization of cyclically loaded structures via direct methods for shakedown**, M. Boissier, J. Deaton, P. Beran, N. Vermaak, submitted (hal-02935509)

Powder bed fusion additive manufacturing is a very promising process, in which a metallic part is built layer by layer [87]. For each of these layers, powder is spread on the top of the already built part. An energy source is then traveled along a prescribed path to melt the powder in a specific domain. The metal solidifies when cooling down (Figure 15). Several advantages come with this technology: among them the ability to manufacture new part designs too complex for traditional machining, or the cost effectiveness of building one part without requiring mass production. These advantages make any technological improvement crucial which stakes are split into two main issues [68, 142]. First, the process must be efficient: reducing the processing time of each layer decreases drastically the total building time. Then, the quality of the part must be insured. Several physical and mechanical phenomena are involved such as change of state or fluid mechanics computations. If ill-controlled, these phenomena may weaken the final part (Figure 16).

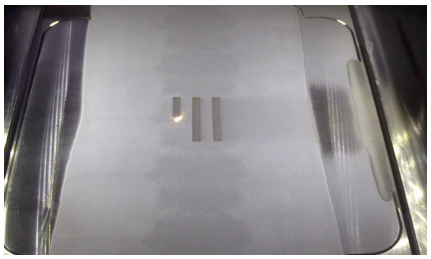


Figure 15: Part building

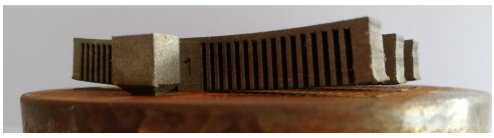
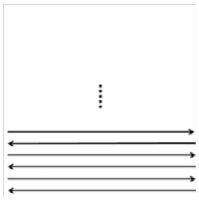
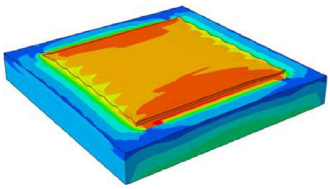


Figure 16: Invalid final part: when removed from the base plate, instead of staying still, both ends of the part go up because of residual stresses.

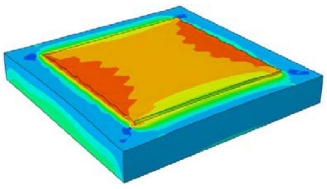
Several different building parameters can be worked on to improve the final result [172]. We choose in this work to focus on the scanning path, the trajectory the source goes along to melt the powder on each layer. Improving the path is a classic issue in automated production such as milling and welding for example. Indeed, the path is an important factor for both the building time and final part quality. As for the first one, whether impacted by the source power, it is mainly related to the system's kinematics and especially the ability of the actuators to accurately and rapidly travel along the path [92]. The quality of the final part is on the contrary related to the physical and mechanical phenomena caused by the many changes of state [68]. Path planning is thus a crucial step since it determines the heat distribution evolution over time and space [77]. Badly controlled, it may cause porosity, element loss or residual stresses weakening the part [52, 104] (Figure 17).



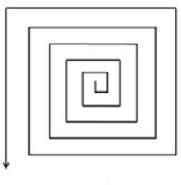
(a) Zigzag path [52]



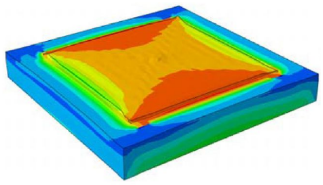
(b) Resulting stress in the (Ox) direction [52]



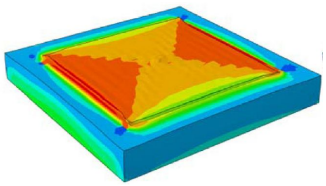
(c) Resulting stress in the (Oy) direction [52]



(d) Spiral path [52]



(e) Resulting stress in the (Ox) direction [52]



(f) Resulting stress in the (Oy) direction [52]

Figure 17: Zigzag and spiral paths with the residual stresses induced

We choose in this work to involve not only the scanning path but also the part's design. Indeed, if the process is of course one main component of the building, the part to build constitutes another one. Struc-

tural optimization has been initiated by Hadamard in 1908 [95] and consists in designing optimal shapes with respect to physical, mechanical or geometrical constraints. Mixing several theoretical (control, optimization) and numerical (differential equation resolutions, numerical representation, algorithms) fields of applied mathematics [9, 15, 97], it goes especially well with additive manufacturing. Indeed, structural optimization usually results in complex designs that cannot be built with traditional machining. Additive manufacturing allows for the building of "more optimal" parts. Conversely, structural optimization is also a tool for the building technology. Many recent developments take into account production constraints. On supporting issues, this is for example the object of [11, 12, 13, 14] (Figure 18). Another example is [16] in which the design includes the limitation of mechanical defects after building (Figure 19).

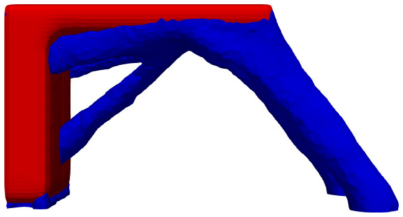


Figure 18: Part (red) and building support (blue) [11]

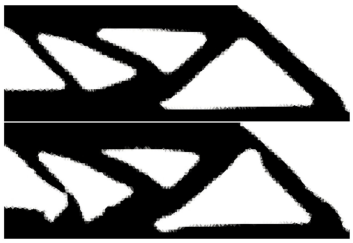


Figure 19: Minimization of the elastic stress without thermal considerations (up) and with thermal considerations (bottom) [16]

In this work, we argue that it is possible to involve the efficiency and quality constraints related to the scanning path in the part's design itself. Then, shape optimization would allow the user to compromise between the optimality of the part's design and the efficiency of the building process. To this purpose, process control and physical constraints must be transformed into a mathematical formulation then involved in a shape optimization algorithm.
Before performing the shape optimization step, a first question consists in further studying the path to detail a notion of "good path". Many works in the literature experimentally and numerically evaluate the paths already in use, mainly based on specific strategies (zigzag, offsets, fractals). For each strategy, the strengths and weaknesses are detailed but no a priori evaluations have been developed. A second question focuses on the impact of the part geometry on the scanning path. In most industrial applications, the scanning path is chosen independently from the part to build: each strategy can be evaluated for each object. On the other hand, only very few works consider coupling the path and the part optimization [6, 7, 51].

In order to answer both questions, we have chosen to work on a scanning path optimization algorithm, using mathematical tools and without any a priori fixed strategy. Then, we analyze the results from several different situations and in particular for different parts geometries as well as from the coupling between shape and path optimization.
Such a study is not widely considered in the literature (to our knowledge, only [6, 7, 51]) urging us to make several choices. First, we set a physical framework and a corresponding optimization problem. This framework rests upon many modeling assumptions required for decreasing the computational costs. Moreover, because of the rare literature, the settings must be kept simple at first to ease the production of the first results and allow for a relevant analysis. Then, we propose a numerical representation of this framework then leading to the development of optimization algorithms. Using the results, we analyze the different modeling, discretizing and optimization choices. We then make the problem more complex by the allowing topology modifications of the path. We finally couple both the path and part shape optimization.

This work mixing automated production and applied mathematics has been realized in the framework of the SOFIA (SOlution pour la Fabrication Industrielle Additive métallique) project, initiated in 2016 by the company AddUp and aimed at developing an industrial powder bed fusion additive manufacturing solution. The goal of this project is the improvement of the existing technologies through research at each of the process scales. To gather various competences, a consortium has been created (Figure 20) with industrial actors (Aubert & Duval, ESI Group, FUSIA, Michelin, Safran, Volum-e, Zodiac Aerospace) and academics (CNRS and especially CentraleSupelec, Centrale Nantes, École Polytechnique, ENS Paris-

Saclay, Université Paris Diderot, Université Paris-Sud, Université Pierre et Marie Curie - Paris Sorbone Université) .

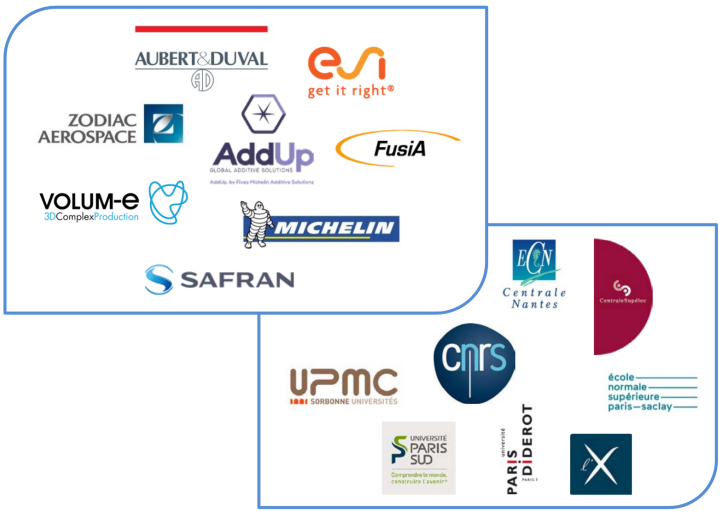


Figure 20: SOFIA project consortium

This work, collaboration between the CMAP (Ecole Polytechnique) and LURPA (ENS Paris-Saclay), comes within the numerical aspects of this project, at the boundary between shape optimization and scanning path planning.

Manuscript structure

This work is divided into two different parts. In the first one, we introduce the context with three chapters detailing the state of the art: Chapter 1 focuses on the powder bed fusion additive manufacturing, Chapter 2 on constrained optimization and Chapter 3 on structural optimization. The second part details the PhD contributions.

Part 1

In Chapter 1, we describe the powder bed fusion additive manufacturing process. The process is first described and two main stakes of the technology are further detailed: better understanding the phenomena through experimental works or numerical simulation and improving the technology. The literature related to scanning paths is then discussed.

This PhD being aimed at optimizing the path and the shape, constrained optimization algorithms must be introduced. This domain is widely developed: each application requires its own specificity. We limit here the study to the descent methods used in the remaining of this work and on the theory they are based on.

In Chapter 3, we detail structural optimization theory, defining the differentiation and gradient with respect to curves. To end with, a classic example of shape optimization in the context of linear elasticity is performed, the shape being represented by a level set. Numerical results illustrate the process. We use the shape optimization theory at several places in this work: to optimize the path itself and to couple path and shape optimization, as mentioned in this work's objective.

The context set in Chapter 1 and the theoretical notions introduced in Chapters 2 and 3 are useful all along the contributions presented in Part 2.

Part 2

Chapter 4 is dedicated to the physical model set and the path optimization problem chosen. An exact model involving all the physical phenomena cannot be considered in this work: the complexity of the phenomena causes a far too high computational cost. We detail the compromises made.

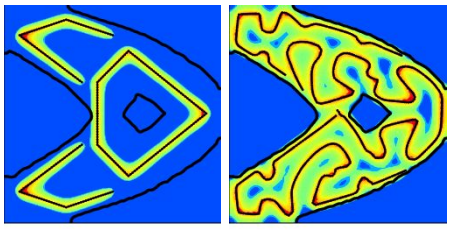
We focus on the thermal problem only, represented by a two dimensional heat equation involving conduction only, in which the source travels along the path with constant velocity. The path constraints are taken into account by the traveling time (called final time) minimization and by bounds on the temperature (to constraint the part's quality). This model being very simple, we apply a calibration process to adapt the coefficients involved and allow for physical interpretation of the results.
A second model is then introduced. Called the steady state, it does not involve any source movement anymore but on the contrary that the source is applied brutally on the whole path. We calibrate this model in order to fit the maximum temperature over time simulated in the transient context. Although a toy model, this framework eases the adaptation of the optimization algorithms and thus provides a first cheap analysis on the difficulties we may bump into working on the transient model.

The model proposed in Chapter 4 must be discretized to numerically solve the problem. We present in Chapter 5 the choices we make: a front tracking method is used, in which a physical mesh is created and fixed for the whole optimization process. The path is discretized as a broken line and modified at each optimization iteration.
In the steady state case, the broken line nodes are the optimization variables. To maintain the discretization coherence, the distance between two consecutive nodes is controlled by a re-discretization process after each iteration. The normal and curvature computations are detailed as well as the methods to carry the information from the broken line to the physical mesh and the way back, and the integration and differentiation processes along the path.
In the transient case, we present two different line control choices. In the first one, the distance between two consecutive nodes is fixed and the optimization variables are the first node position, the final time and the angles between the tangent to the path and the $(Ox)-$axis. In the second one, as in the steady state approach, the nodal points are the optimization variables which requires a rediscretization process and induces non constant distances between consecutive points.

Chapter 6 is the first optimization chapter, focusing on the steady state. The model detailed in Chapter 4 and the numerical choices given in Chapter 5 lead to setting an optimization algorithm. In the steady state context, optimizing the path amounts to optimizing a curve carrying the source. Shape optimization theory, presented in Chapter 3, can be applied to determine the derivative and the gradient to the problem with respect to the path. Then, the gradient is discretized to numerically modify the path. Two different order one algorithms, presented in Chapter 2 are then tested.
The first tests confirm the correct behaviour of the algorithm. We analyze the different choices made all along this work: modeling, discretization, optimization algorithm. Then, several tests starting from different initializations aimed at scanning different geometries are run. The results lead us to a first step back on the optimization method and on the notion of "good path". (Figures 21 and 22).

(a) Initial path (b) Final path

Figure 21: Path to build a part in aluminium, steady state model (blue: not melted, red: temperature too high)

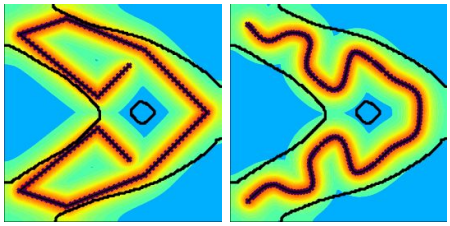


(a) Initial path (b) Final path

Figure 22: Path to build a part in titanium, steady state model (blue: not melted, red: temperature too high)

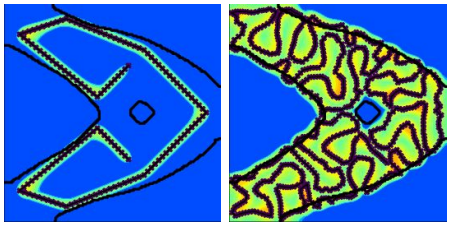
We focus in Chapter 7 on the transient model. We first consider the angle based path discretization. Differentiating with respect to the corresponding variables allow for setting a first optimization algorithm. However, the results highlight the drawbacks of this method: the final time management is complicated. The point based path discretization is then chosen . The distance between two consecutive nodes being unfixed, the computational time increases a lot. However, the results are from far better and, as in the steady state case, allow for an analysis of the different choices made. Finally, several tests are run to

increase the intuition on the "good path" notion (Figures 23 and 24).



(a) Initial path (b) Final path

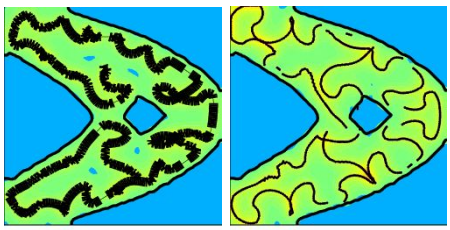
Figure 23: Path to build a part in aluminium, transient model (blue: not melted, red: temperature too high)



(a) Initial path (b) Final path

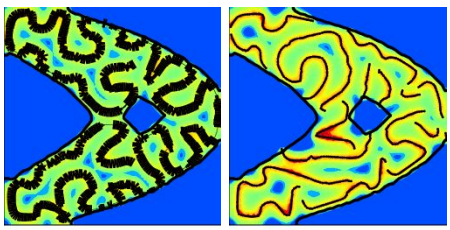
Figure 24: Path to build a part in titanium, transient model (blue: not melted, red: temperature too high)

We intend in Chapter 8 to increase the admissible paths domain: to give more freedom to the path design, we allow the modification of the number of path connected components. Two methods are set and tested in the steady state case.
The first method is based on the physic of the process and corresponds to a collaborative work with Tonia Maria ALAM and Serge NICAISE (LAMAV, UPHF). Cutting the path physically amounts to switching off and on the source. The power is added to the optimization variables and allowed to be either 0 or P_{max}, with P_{max} the source power when the source is switched on (industrial solutions do not allow for continuously modifying the source power; to add such a feature, the velocity should be modified and remains a perspective of this work). To deal with this new variable, we choose a relaxation-penalization technique. A constraint on the power jumps is added to prescribe degenerated solutions in which the source would be switched off and on an infinite number of times. Numerical results are then given. If not presented in this work, this method could be adapted to the transient case.
The second method is specific to the steady state and consists in using the topological derivative [54, 153, 164, 175]. Mainly developed in the literature to modify the topology of a two or three dimensional domain, this notion also exists in one dimension for cracks characterization. It is here adapted to the steady state case with a modification of the algorithm developed in Chapter 6: the cut or addition of new connected components is now possible. Numerical results illustrate this study and both methods are compared for different geometries (Figures 25 and 26).



(a) Power based method (b) Topological gradient based method

Figure 25: Path to build a part in aluminium, steady state model - number of path connected components modified (blue: not melted, red: temperature too high)

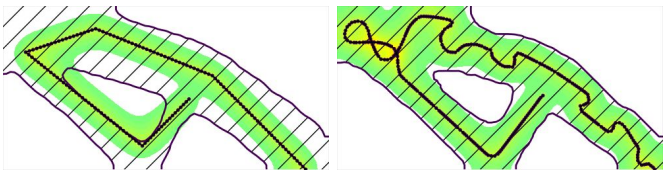


(a) Power based method (b) Topological gradient based method

Figure 26: Path to build a part in titanium, steady state model - number of path connected components modified (blue: not melted, red: temperature too high)

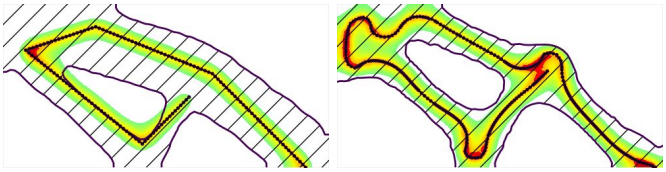
Finally, to further study the impact of the part's shape on the path, we present in Chapter 9 an algorithm of coupled optimization between the shape and the scanning path. The shape optimization problem is a classic compliance minimization under a volume constraint. In the steady state case only, we add the path constraints. The results obtained in this Chapter remain preliminary but already highlight the shape modification when adding the path constraints (Figures 27 and 28). Analyzing these modifications finally give intuition of the design requirements and confirm the relevance of the approach developed to

further study this problem.



(a) Shape optimization without path constraints (b) Concurrent shape and path optimization

Figure 27: Results comparison when the shape is optimized without path constraints and when a concurrent path and shape optimization is run, to design an aluminium part, steady state model(part geometry hatched, white: not melted, red: temperature too high)



(a) Shape optimization without path constraints (b) Concurrent shape and path optimization

Figure 28: Results comparison when the shape is optimized without path constraints and when a concurrent path and shape optimization is run, to design a titanium part, steady state model(part geometry hashed, white: not melted, red: temperature too high)

Scientific communications and publications: the work presented in this thesis led to the following communications:

Journal articles

- **Scanning path optimization using shape optimization tools**, M. Boissier, G. Allaire, C. Tournier, Structural and Multidisciplinary Optimization, 61:6, pp. 2437-2466, 2020 (awarded the ISSMO/Springer 2019 prize)
- future publications in preparation

Communications

- **Optimisation de trajectoire pour le procédé de fabrication Laser Powder Bed Fusion (LPBF)**, CSMA 2019 (Mai 2019), Giens, France
- **Laser path optimization for additive manufacturing**, WCSMO (Mai 2019), Beijing, Chine
- **Laser path optimization using shape optimization tools for the Laser Powder Bed Fusion process**, SimAm 2019 (Septembre 2019), Pavie, Italie

Seminars

- **Simulation thermique et optimization de trajectoire pour la fabrication additive métallique par le procédé Selective Laser Melting (SLM)**, journées de mi-projet SOFIA (Avril 2019), Clermont-Ferrand, France
- **Utilisation de l'optimisation de forme pour la détermination de trajectoires de lasage**, séminaire FAPS (Février 2020), Palaiseau, France

Posters

- **Couplage de méthodes d'optimisation topologique de formes et d'optimisation de trajectoires en fabrication additive**, CANUM 2018 (Mai 2018), Cap d'Agde, France
- **Path optimization for the laser powder bed fusion additive manufacturing process**, New trends and challenges in the mathematics of optimal design, Cambridge, 2019
- **Path optimization for the laser powder bed fusion additive manufacturing process**, New Trends in PDE Constrained Optimization, Linz 2019

As well as

Journal article

- **Elastoplastic topology optimization of cyclically loaded structures via direct methods for shakedown**, M. Boissier, J. Deaton, P. Beran, N. Vermaak, submitted (hal-02935509)

CHAPTER 1

POWDER BED FUSION ADDITIVE MANUFACTURING CONTEXT

Contents

1.1 Introduction . . . 29
1.2 Powder Bed Fusion (PBF) additive manufacturing process . . . 30
1.2.1 Process presentation . . . 30
1.2.2 Characterizing and modeling the process . . . 32
1.2.3 Improving the process . . . 34
1.3 Path optimization in the PBF context . . . 36
1.3.1 A brief history of path optimization for manufacturing tools . . . 37
1.3.2 Stakes and path constraints . . . 37
1.3.3 Optimizing the laser properties along a fixed path . . . 38
1.3.4 Existing scanning paths . . . 38
1.3.5 Islands scanning path strategies . . . 42
1.4 Design to manufacturing . . . 43
1.5 Problematic of this work . . . 43

1.1 Introduction

Additive manufacturing (AM) [33, 87, 180] is a technology that consists in, according to the standard ISO/ASTM 52900 [101], "joining materials to make parts from 3D model data usually layer upon layer" (Figure 1.1). Widely developed during the past few decades, it can be derived into many different processes usually gathered into 7 categories. Each of them has specific advantages, costs, manufacturing scales and compatible materials. This diversity justifies the large range of domains in which this method is used or at least seriously investigated: biomedical [4, 174], construction [69, 179], aerospace [100, 116, 184] are part of this non exhaustive list.

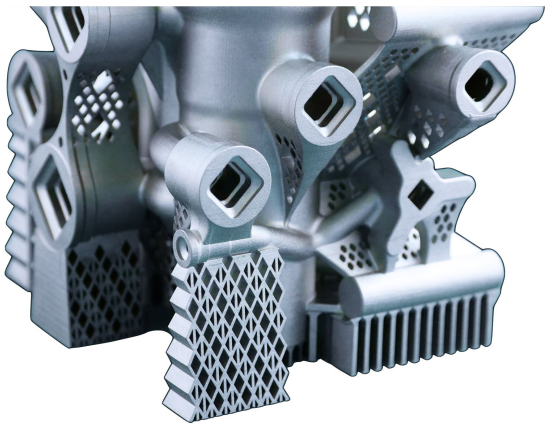


Figure 1.1: Example of a part built by metallic additive manufacturing [2]

Now in rapid expansion, this technology presents many advantages [87, 180]. The first benefit, and maybe the most obvious, affects the freedom in the part design: this technology allows for more complicated shapes which is a crucial feature. Indeed, at the part scale, not only it impacts the aesthetics but also the inner structure, improving the performance, allowing for lighter objects thus saving material and facilitating the inclusion of embedded items. At a microscopic scale, the ability to manufacture convoluted structures impacts the material organization (such as metallurgy for metals), its composition and porosity, modifying the macroscopic properties of the final item. Finally, this freedom might simplify

the design by decreasing the number of objects to assemble, thus improving the final resistance of the part.

A second benefit consists in the ease to successively build different items. Indeed, once the part's CAD and the building path are set, the data are read and the manufacturing process starts: building a new part induces a computational cost but requires no material investment, unlike traditional tools such as molding. Among several fields, prototyping, objects repairing, personalized medical implants or prostheses illustrate quite well the profit this induces [4, 116, 130].
If already in use, this technology is still under study to reduce its drawbacks [87, 180]. First, the final quality of the part must be quantified and controlled. This implies experimental work as well as modeling research to be able to provide a priori insurance of the part resistance and ability to achieve its goals. These quality limitations must then be translated into manufacturing constraints split into two different stakes. The first one consists in improving the technology, in order to prevent the damage to occur whereas the second one involves the formulation of design constraints to avoid these issues. A huge part of additive manufacturing research concentrates on solving these drawbacks and this work comes within this scope.
In this chapter, the context of this work is introduced. Dealing on metal additive manufacturing and especially in powder bed fusion technologies, Section 1.2 describes the different processes and depicts the main stakes under actual research. In Section 1.3, the scanning path literature is further expanded with the stakes involved and the existing scanning strategies. In Section 1.4, following the problematic described in the Introduction, structural optimization in the context of metal additive manufacturing is detailed. Section 1.5 restates the objective of this work given the detailed context.

1.2 Powder Bed Fusion (PBF) additive manufacturing process

Among the materials available in AM, this work focuses only on metal which can be used with four different processes. Binder jetting, powder bed fusion and directed energy deposition all work with metallic powder. In binder jetting, the building is realized in two steps. The first step is a layer by layer process in which, iteratively, some powder is spread before binder deposition at specific places. A sintering step is then applied to the assembled part to slightly melt the metallic powder and achieve the building. Powder bed fusion (PBF), further detailed below, is also a layer by layer technique in which energy instead of binder is deposed, to directly melt the powder and thus avoid the sintering step. Direct energy deposition differs from layer by layer processes since the energy and the metallic powder are deposed at the same time, getting closer to welding processes. Finally, sheet lamination consists in deposing a metallic sheet before cutting and bonding it to the previous layers (again a layer by layer process). If each method has its advantages and drawbacks as well as its specific fields of application [68, 98, 142], only powder bed fusion processes are considered in this work.

This section further details the PBF technology and present its main stakes: characterizing and modeling the process in a first time to then perform improvements.

1.2.1 Process presentation

The PBF manufacturing technology is a layer by layer process [35, 68, 98, 142, 144]. Starting from a metallic build platform, for each layer, a metal powder layer is spread with a recoater blade on top of the part in construction. Then, energy is deposited along a scanning path. The metal heats until melting or at least coagulating, while the cooling leads to solidification (Figure 1.2). Two main categories are set apart, depending on the energy source: laser or electron beam. Powder bed fusion processes using lasers are referred to as Laser Power Bed Fusion (L-PBF) whereas the ones using electron beams come as Electron Beam Powder Bed Fusion (EB-PBF). Each of them involves specifications and requirements.

Comparison between L-PBF and EB-PBF

In L-PBF, the source is a high power-density laser, which power can be of some hundreds of watts. The energy comes from photons first focused by lenses and then moved along a prescribed path. This motion is carried out by mirrors controlled by galvanometers (Figure 1.3). The inertia of the latter bounds the available scanning speed. Part of the energy is lost by reflection, which parameter is related to the metal powder used (specific absorption coefficient depending on the wavelength) [142]. This process does not require high temperature preheating (though it is a crucial building parameter (see Section 1.2.3)) nor a

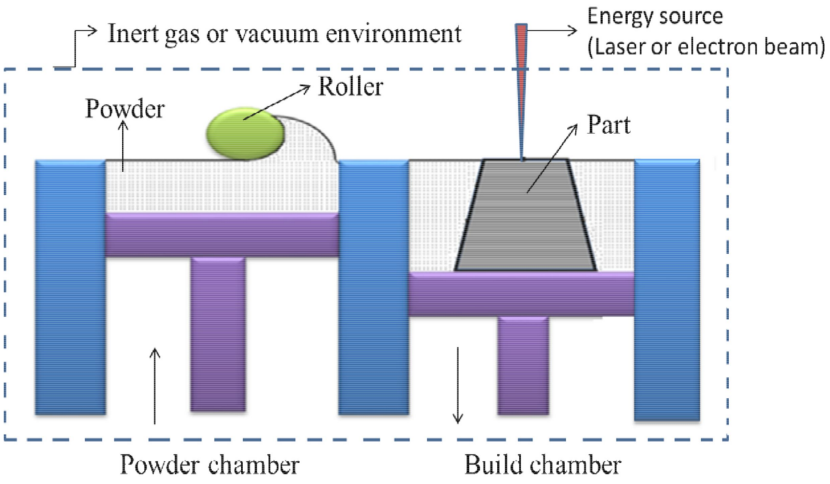


Figure 1.2: Powder bed fusion technology [35]

partial vacuum environment. However, the building chamber is actually filled with an inert gas to avoid any reaction between the metal fused and the environment [142, 199].
In the EB-PBF, electrons, controlled by magnetic fields, are the energy source (Figures 1.4). The resulting power can be a lot higher than in the laser case (Figure 1.1). On the contrary to the galvanometers, no inertia problems are involved in magnetic fields control and the scanning speeds can be ten times larger. However, the preheating temperature must be higher than using lasers: using electrons implies electrostatic repulsion that must be avoided by heating the metal to increase its conductivity. Finally, to avoid any reaction of the electrons before they get to the powder, the chamber environment must be as close as possible to vacuum [142, 166, 199]. This last point, quite restrictive, is actually one of the main issues of this energy source.

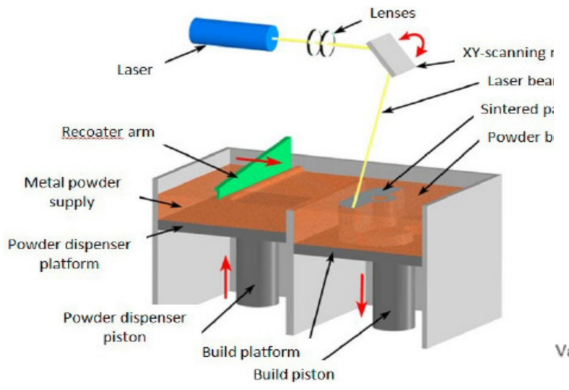


Figure 1.3: L-PBF process [35]

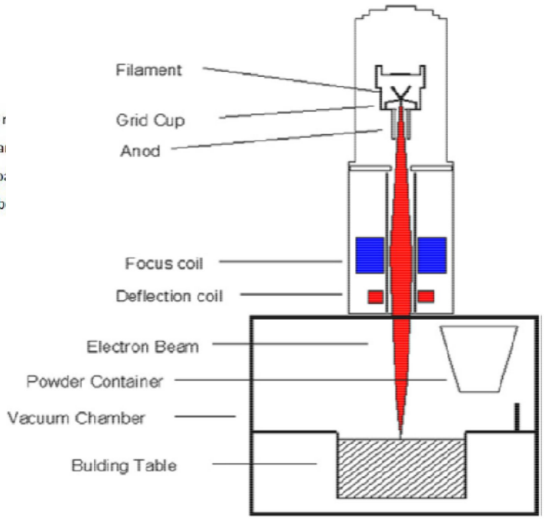


Figure 1.4: EB-PBF process [35]

The different characteristics are summed up by Figure 1.1.

Parameters	Form Up machines [1] AddUp (L-PBF)	EBM S12 [199] Arcam (EB-PBF)
Environment	argon and nitrogen	vacuum ($10^{-4} - 10^{-5} mbar$)
Preheating ($^\circ C$)	$200 - 500$	700
Maximum beam power (W)	500	3500
Average powder layer thickness (μm)	$20 - 100$	$50 - 200$
Beam scan speed ($m \cdot s^{-1}$)	10	> 1000

Table 1.1: Quantitative results of the power optimization for the aluminium

Feedstocks material

Two main features characterize the feedstock: the powder geometry and the powder material.

The first characteristic is deeply related to the means of powder production. Among the different existing techniques, three main ones are mentioned: water jet atomization, gas atomization and plasma atomization. In the three of them, the objective is to melt metal at the top of the atomization tower and let the drops fall down. During this fall, cooling effects are introduced changing the liquid drop into a powder grain. In the three processes, the metal comes as a spool which end is melted. Then, in water jet atomization, the cooling comes from water whereas in both gaz and plasma atomizations, this cooling comes from gaz jets [98, 142, 144]. The process impacts the final geometry and density of the powder, thus influencing the building process (the geometry is directly related to the energy absorption) [68, 86, 113, 198, 213].

The second characteristic is the powder material. The most common ones in the L-PBF process are steel, aluminium, nickel and titanium alloys [98]. The choices in the EB-PBF context are a bit more limited. References related to researches regarding different metals and different technologies can be found in [121]. In this work, we focus on titanium and aluminium only, although the techniques developed can naturally be applied to other materials.

Post-treatments

To finalize the item, post processes are applied. A first one is the removal of any part that does not belong to the expected geometry: the un-melted powder, the basis platform, the building supports (see 1.2.3) used to hold the part against gravity or to limit thermal stresses during the building. To end with, surface treatment, such as blasting, polishing or chemical, are also applied to limit the roughness. Heat and pressure treatments are the last ones, to get the engineering and mechanical requirements by dealing with residual stresses, porosity or microstructural issues [142].

As promising as this process seems, some issues yet need to be solved. Many thermal, mechanical and metallurgical phenomena occur, affecting the part's quality [68, 98, 130] and among them: residual stresses, anisotropy related to the microstructure, porosity, surface roughness and loss of alloy elements. Moreover, especially in the L-PBF context, industries are willing to decrease manufacturing times. These different issues are the actual technology's stakes and are further developed.

1.2.2 Characterizing and modeling the process

To begin with, we consider the process' modeling. This issue is crucial: it is mandatory to compute and quantify the occurring phenomena to guarantee the final quality of the part and thus its efficiency. It is also essential to improve the understanding of the process to allow for its enhancement and for the formulation of corresponding design constraints. However, this task is challenging and is a current field of research: the high energy deposition inducing at least melting and even evaporation, drags down many nonlinear phenomena [68, 130, 142, 143, 154]. Three physics, thermal, mechanical and metallurgical, interweave (Figure 1.5) and, with four physical states involved (solid, powder, liquid, gas), there is a real challenge in fully understanding the building process and an even harder one to realize high-fidelity simulations satisfying both accurate predictability and low computational cost requirements.

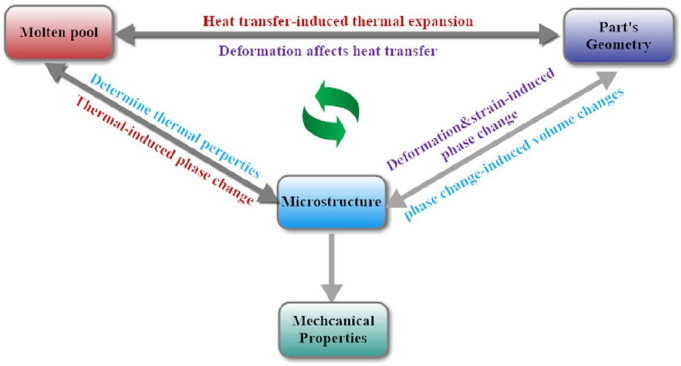


Figure 1.5: Three physics interweaving [79]

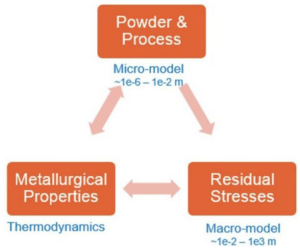


Figure 1.6: Modeling scales [139]

In order to handle both precise modeling and fast computing issues, the study of the process is split into physical scales resulting from the chosen compromise. Most commonly in the literature, the microscopic and macroscopic scales are set apart [85, 139] (Figure 1.6).

Microscopic scale

The phenomena taken into accounts by this scale involve the interaction between the heat source and the feedstock and a full model of the melting pool including the liquid and gaseous states (Figure 1.7). A careful model of the powder-source interaction requires characterizing the feedstock geometry as well as the energy deposition. Different simulation techniques have been developed, among then discrete particles, finite element or Lattice-Boltzman models [86, 94, 113, 198].
Then, an advanced study of the fluid mechanics phenomena is realized, taking into account the non linearity dependence of the material properties with respect to the temperature. The metal evaporation and the surface tension, including normal and tangential Marangoni stresses, are considered. The shape and size of the melting pool are then determined, highlighting defects' formation [68, 139]. Involving Navier Stokes equations, computational fluid mechanics methods are used for simulations, combined with different numerical techniques (see [59] for a review) such as for example adaptive grids, level set methods or simulations of the thermics only [59, 88, 112, 139, 147, 148, 202, 209, 211]. However, the huge number of variables, equations, as well as their non linearity make the use of optimization algorithms at this scale computationally too expensive.

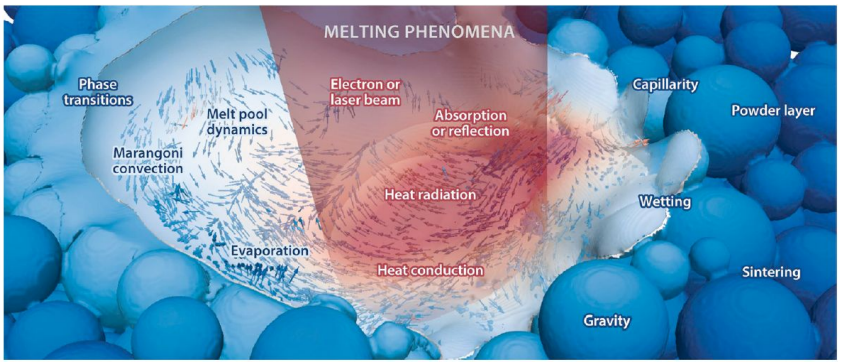


Figure 1.7: Phenomena happening at a microscopic scale [59]

Macroscopic scale

In order to reduce the computations, the macroscopic scale does not consider a full fluid mechanics model. Through approximations coming from the microscopic scale's results, it mostly focuses on two states, powder and solid, and the change of phase is considered as an abrupt change in the material properties. Only three thermal phenomena are taken into accounts: conduction, convection and radiation whereas the interaction between the heat source and the powder is simplified [139]. This approach does not allow for an accurate quantification of the different phenomena. However, it is enough to highlight

thermal expansion and the appearance of residual stresses during the manufacturing, both of them implying important defects resulting into early damage and fatigue. Focusing on thermo-mechanical effects, this scale allows for coarser meshes. Many numerical methods have been developed depending on the assumptions and the phenomena emphasized (related to residual stresses and thermal expansions in most cases), among them birth and death of elements techniques, adaptive grids or level set methods to follow the evolution of the powder area [50, 70, 124, 139, 167, 187, 204, 210]. All these approximations result in computational costs from far lower than in the microscopic scale, compatible with optimization algorithms.

1.2.3 Improving the process

The progresses in experimental and modeling researches improve the understanding and facilitate the computations. This leads to a second class of ongoing research topics: improving the process. Indeed, the experimental and modeling studies point out the defects related to one or more process' parameters. Some studies actually count 150 different ones [172] (such as the powder characteristics, scanning path criteria or building chamber environment for example), which, carefully chosen and even optimized, could drastically improve the final item's quality. In the following, a review of the main defects emphasizes some of them.

Loss of an alloy element

To start with (and following the ordering given by [68]), the loss of an alloy element. Going beyond the change of phase temperature (which depends on the material) causes the melting and even the evaporation of the metal powder. In case of an alloy material, this may lead to an unbalance proportion of the different elements in the final part. Modeling this phenomenon requires microscopic scales' techniques, to relate the energy given by the heat source to the evaporation depending on the building chamber parameters [68, 138, 154].

Porosity

The porosity increases the final ductility and might be the starting points of microcracks [130]. Three main reasons entail it: keyhole phenomena, lack of fusion, and balling effects. Keyhole effects are directly related to the melt pool dynamics: if high enough, the energy might create a deep melting pool. Some gas (might they come from the environment chamber or from metal evaporation) gets trapped in there during the cooling. Lack of fusion, on the contrary, results from lack of deposed energy. Finally, balling effects come out from melt pool instabilities, creating both metal aggregation that increases the surface roughness and porosity.
These phenomena deeply involve the melting pool and the powder quality. Characterizing them requires computations at a microscopic scale that may then be turned into building constraints. Indeed, each of them seems to involve the scanning parameters and studies have especially featured correlations with the source velocity and power (Figure 1.8) [68, 110, 130, 154, 193]. Moreover the impact of the powder quality is also under study and improvements [19, 161, 213].

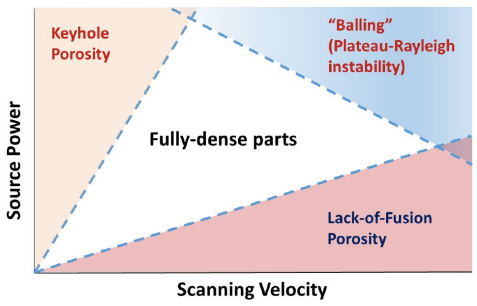


Figure 1.8: Schematic influence of the scanning velocity and power on the porosity, the limit depending on the material used [154]

Post processes and especially hot isostatic pressing (HIP) can close the pores thus reducing the porosity [68, 142]. Still an essential step, it remains time consuming, expensive and does not help with the micro cracks.

Surface roughness

Surface roughness is also a defect. Indeed, might it be for aesthetics or for a correct functioning when assembled to other components, the final item must have a correct surface quality. This is actually also required during the building: each layer starts by powder spreading. Too much roughness could prevent the tool's crossing and even break the spreading mechanism. Among the different causes that step in, the layer by layer process entails a staircase effect on the item boundary. This process also requires the building of supports to balance gravity and to improve the thermal distribution (further developed below). Removing these supports damage the surface smoothness. The contact surface can be reduced by an optimization of the part building orientation as well as the support [11, 126, 160]. Finally, thermal studies point out that heat accumulation [103] or balling effects also damage the surface. To control them, the material and scanning parameters must be chosen with care [61]. Post processes are obviously available and include machining polishing, peening, tumbling or chemical treatments. They nevertheless are time consuming and increase the manufacturing costs [68].

Residual stresses and thermal expansion

During the manufacturing, the metallic powder is subjected to cyclic heating and cooling. This creates strains within the part that might end up as residual stresses. These residual stresses correspond to the stresses remaining in the material after it has reached the equilibrium with its environment. They affect strongly the mechanical properties of the part. If sometimes strengthening it (tensile stresses for example or shakedown behaviour which introduces them to then get back in an elastic state [38, 190]), it might also accelerate fatigue and plastification, leading to earlier cracks and damage [68, 154] (Figure 1.9). If these stresses occur at different scales, the macroscopic ones constitute the main issue and are mainly focused on [27, 140].

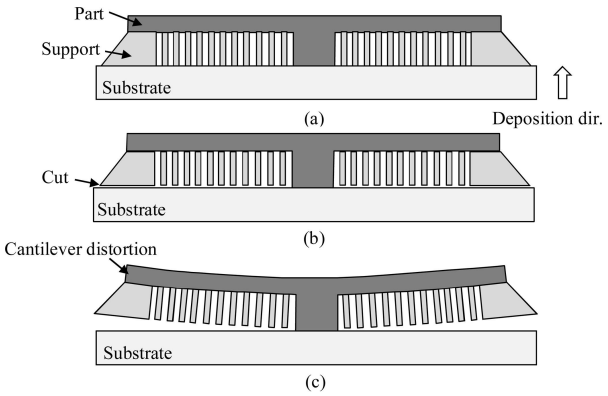
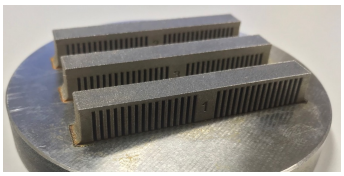


Figure 1.9: Illustration of the residual stresses effect on the final part. (a)Building strategy, (b)Removing the basis plate, (c)Effect of the residual stresses [122]



(a) Part before support removal

(b) Part after support removal

Figure 1.10: Photographs of heating and cooling effects on thermal expansion

Two main phenomena cause these residual stresses [27, 140], both related to the thermal expansion of the material. The first one is due to the spatial temperature gradients in the layer. Indeed, because of the energy deposed by the source, the layer surface tends to expand fast whereas the conduction, on the contrary, is slower and prevents a rapid expansion of the area surrounding the energy spot. This temperature gradient thus results in a difference of expansion which brings distortion and residual stresses (Figure 1.11) [27, 68, 139, 140]. A second phenomenon is related to the cooling. Indeed, the solidification tends to contract the zone deeper than its surroundings (Figures 1.11 and 1.10).
Computing these residual stresses is a challenge. Even if involving macroscopic effects only, it still requires the resolution of a full thermo-mechanical system. This issue has been widely addressed would it be through classic Finite Element Methods (FEM) or alternative techniques [16, 70, 124, 139, 167, 187, 189]. A fast computation is indeed essential. Being critical points in the fatigue characteristics of the final item, the process must be improved to prevent them. An algorithm allowing for a rapid evaluation hence allows for several parameters tests and even optimization. Among the main parameters, the scanning path is once again crucial [27, 51, 68, 154, 159, 172, 189] as well as the preheating temperature [27, 68, 154, 172] and the cooling time between each layer [187]. In addition to the parameters choice, the design of the part to build itself could help with improving the process. To begin with, supports can be added to

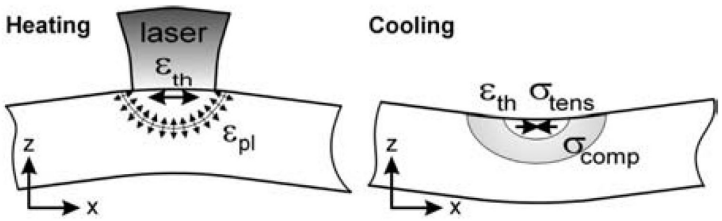


Figure 1.11: Heating and cooling effects on thermal expansion [140]

facilitate heat conduction during the building [11, 28, 53, 185]. Then, the part itself can be designed as an optimized compromise between its purpose and its ability to being built limiting the residual stresses [16].
Post processes can in this case again help to reduce these stresses and especially heat treatments and hot isostatic pressing [68, 79].

Control of the microstructure and anisotropy

The metallic microstructure is a key factor in the mechanical properties of the part. Not only do the grain size and shape affect the fatigue properties and the cracks propagation [68, 128, 154] but they also determine the solid's anisotropy. In the PBF technologies, the microstructure can be predicted. Indeed, it is deeply related to the thermal history, to the depth and shape of the melt pool. In this way, it is correlated with the layer by layer process, the scanning path chosen and of course the powder characteristics. Modeling the microstructure requires both the macroscopic and the microscopic scales. Based on thermal equations only, a specific scale has thus been developed: the mesoscopic scale. Then, many different models based on experiments and sample's characterizations have been formulated as well as computational techniques to predict the microstructure [3, 31, 88, 114, 128].
Some characteristics seem to be key factors of the possible improvements: the temperature gradient, the cooling rate and the metal liquid properties [154]. In terms of control parameters, this coincides with the material and the powder geometry [68, 154, 161], with the scanning paths on the different layers [118, 128, 154, 172], with finally the shape of the part itself which could, like in the residual stresses control, master the final anisotropy [127, 208].
Once again, heat treatment, as well as hot isostatic pressing can modify the microstructure [68, 142].

Support issues

Finally, supporting issues are a crucial point. If not related to a specific defect, they must be carefully considered for two reasons.
First, considering gravity issues: assume that you want to build a part with overhang (let say a T-shape part). Layer after layer, the part is grown and the unused powder left. When the first layer in overhang comes, some powder under which the metal is not solid is melted. However, the solid's density is higher than the powder's. Without any supporting structure, the overhang newly built may fall down. Overhang supports literature has thus been developed: experimental and modeling studies have quantified their impacts and topology optimization has been widely used to design these supports, to reduce the objects' overhang areas and even to design self-supporting items [11, 12, 13, 14, 117, 126, 127, 145].
Then, supports can also be added for thermo-mechanics reasons. Indeed, many defects and especially residual stresses come from heat accumulation and high temperature gradients. Thus, to even out the temperature, thermal supports are added, bringing out some of the part's heat [11, 28, 53, 185].
In both cases, the support's building requires both energy and material. Thus, their design must be carefully elaborated to spare metallic powder and building time. Finally, their attachment to the structure must also be well thought to ease their withdrawal still preserving the surface quality.

1.3 PATH OPTIMIZATION IN THE PBF CONTEXT

The PBF processes can be controlled through several different parameters [172], some of the most crucial related to the scanning strategy. Indeed, the scanning velocity and source power impact the porosity, the surface roughness, the residual stresses and the microstructure. Moreover, many of these defects being tightly related with the temperature distribution, the scanning path itself is essential. This section

reviews the existing scanning paths' strategies for PBF process. It starts by a brief presentation of the scanning strategy issue for traditional manufacturing techniques, additive manufacturing's precursors. Then, the main constraints involved in a scanning path choice are developed. Finally, two types of strategies optimizations are described: first, the path shape being fixed, the velocity and power are optimized, and then, different path shapes are designed to better satisfy the objectives.

1.3.1 A brief history of path optimization for manufacturing tools

Additive manufacturing processes remain relatively new with first studies related to the scanning strategies started in the 2000's. However, other manufacturing processes also involving paths, such as milling or welding, have been used for years. With the development of Computer Numerical Controlled (CNC) technologies, the research on scanning strategies have increased. These strategies constitute an interesting basis for powder bed processes and additive manufacturing in general. Indeed, various challenges considered in PBF processes are also addressed by these technologies such as build time control or, in welding for example, surface quality, metallurgical and mechanical constraints. If several approaches have been considered, two main path categories have mainly been worked on: parallel and offsets.
The parallel category gathers raster, zigzags and any scanning strategy based on parallel straight lines. If they were first optimized to fulfill filling and geometrical requirements, physical constraints have then been added, with the consideration of the hatch distance, the lines orientation, the starting points and part decomposition in cells [20, 62, 123, 158].
The offset category focuses on any strategy based on the part contour. In this case too, geometrical as well as physical requirements were considered and controlled through the hatch distance, starting points, part decomposition as well as smoothing path processes limiting the curvature [39, 62, 80, 123, 203, 212]. These scanning strategies (the two main categories supplemented by various others such as skeleton based [108] for example) have deeply inspired the scanning paths in the additive manufacturing context.

1.3.2 Stakes and path constraints

Before getting in any further in the consideration of the different paths, the main parameters regarding their optimization are developed. Then, follows the issues involved in PBF processes scanning strategies, gathered around two main goals: the part quality and the manufacturing time. Both involve specific constraints, in many cases concurrent and thus forcing to compromise.

Scanning path parameters

If many criteria are involved in the scanning path, five of them mainly impact the process. First, the path's shape itself, that determines the location of the energy deposition and thus highly impacts the temperature distribution. Then, the power and the velocity determine the amount of energy at each of the scanning path points, as well as the scanning time. Finally, the layer's thickness impacts the effect of the energy deposed: melting of the powder or remelting of an already solid part. Combining these parameters give the volumetric energy along the scanning and are related to keyholes phenomena, balling effects or lack of fusion. This volumetric energy is a very interesting parameter, still under examination [43, 48, 83] (and references therein). Indeed, it gathers the source power and velocity but also the material properties and could be a precious tool to optimize the building process.

Object quality constraints

As already mentioned, PBF processes must be carefully run to meet the quality requirements. Else, some of the defects listed above (porosity, surface roughness, residual stresses, microstructure issues or loss of alloy elements) might occur, affecting the usefulness and efficiency of the final part [68, 154]. The scanning strategy is crucial. Controlling the source power and the scanning velocity is proved numerically and experimentally to be essential parameters in the porosity, the microstructure and in residual stresses issues [93, 119, 154]. Moreover, the scanning path affects the temperature distribution along time and is thus related to the solid's thermal expansion leading to residual stresses formation [52, 104] as well as to the introduction of anisotropy [127].

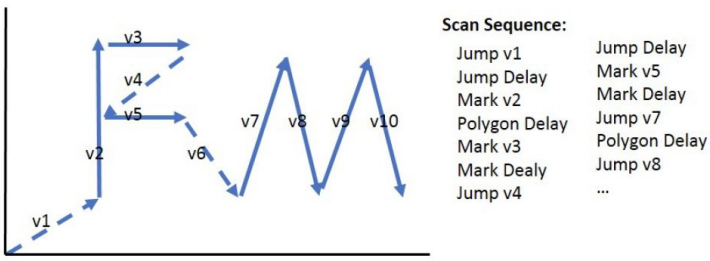
Manufacturing time

Like traditional machining tools, one of the objective of additive manufacturing is a minimal building time. In a layer by layer process, this time is first related to the slicing, which determine the layers' num-

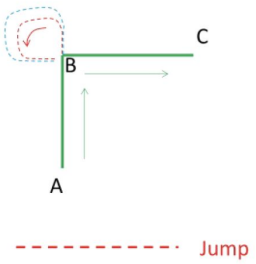
ber, thickness and manufacturability. Also involved in the support generation issue, slicing is actually a field in itself [11, 40, 160, 177, 201]. Then the scanning strategy comes, determining how long does each layer take and being thus an essential parameter in the total building time. This strategy must take into account the machine possibilities.

To illustrate them, we now focus on the L-PBF process and especially on the scanning of one layer only. The scanning strategy, namely pattern, speed and power, being chosen, it is then sent to the machine to be realized. A first step in this process is the transformation of this scanning path into a point wise signal. The laser power source is then moved from point to point thanks to mirrors controlled by galvanometer (Figure 1.3). The mirrors having inertia, it is thus a real challenge to perfectly control both their position and their scanning speed [92, 165, 205]. If straight lines do not require that much difficulty, the higher the curvature gets, the more complicated is the scanning and any lack of smoothness results in an issue. Indeed, to accurately go along this path, the velocity must be reduced, and thus the building time increased. The manufacturing time requirement is in this way more complex than simply reducing the path length and implies curvature and jumps control, as well as uncertainty issues.

Moreover, in addition to being involved in the scanning time, any speed modification impacts the temperature distribution. For example, to be well executed, a corner requires a speed decrease. This leads to an augmentation of the temperature and thus to porosity defects. A first solution consists in controlling the power. However, the actual technologies do not allow a full mastering of the source energy. Else, one could think about switching the laser on and off (Figure 1.12). However, any "jump" in the path is time consuming: in addition to the mirrors inertia, the laser has a thermal inertia that fix a "jumping delay" [200].



(a) Scanning path to take into account to "build" the word "FM" (movement called "jump" when the laser is OFF and "mark" when ON) [200]



(b) Scanning strategy to build a corner: the laser is switched off before turning, to create an accurate right angle [134]

Figure 1.12: Building constraints related to the energy spot control

In EB-PBF, the movement of the energy source is controlled by magnetic fields, leading to different constraints, which, to the author's knowledge, have not been gathered into specific research papers.

1.3.3 Optimizing the laser properties along a fixed path

A first way to optimize the scanning strategy is to first fix a pattern and then optimize the source power and the velocity [191, 192]. Straight from the traditional machining processes, these strategies are mainly applied to parallel lines, offsets and spiral patterns. These source power and velocity control studies can be split into two different categories. The first one is based on numerical computations and experiments, to understand the sensitivity of the building defects to the power and velocity. These studies generate value tables guiding choices [26, 44, 60, 119, 154, 189]. The second ones run optimization algorithms on simplified models to find the best parameters fitting their requirements. Among the strategies in development, control and inverse problems get attention [23, 75] as well as machine learning approaches and neural networks [29, 137].

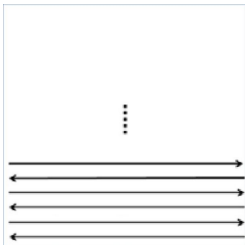
1.3.4 Existing scanning paths

A second way to optimize the scanning strategy is to optimize the path itself. This path development, started with traditional machining, has evolved to adapt to new manufacturing processes. This paragraph

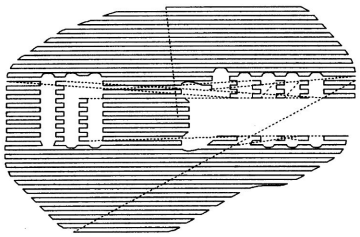
is not restricted to PBF but covers the main approaches in any additive manufacturing techniques. The scanning path classification used here is mainly taken from [73]. It covers the three main traditional machining paths: parallel lines, offsets and spirals, followed by their combination into hybrid paths. Then, fractals and skeleton-based paths and, more recently in the additive manufacturing context, level set optimization paths.

Parallel lines

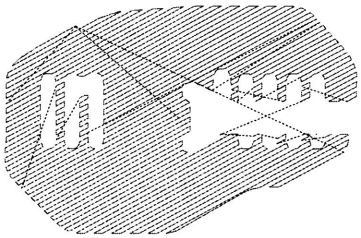
The first strategy type is parallel lines (Figure 1.13(a)). Developed first for machining, it consists in a set of straight lines, from simple raster to zigzags, their continuous version. Filling spaces with this strategy class is very intuitive and optimization based on geometric criteria is easy through the modification of the lines orientation (Figure 1.13(b-c)), the hatch distance and even, in case of zigzags, the number of connected components they are composed of [20, 123, 131, 158, 163, 168]. Optimizing these is actually really important in terms of productivity: if a straight line is easy to scan, raster paths also imply stopping the source and zigzags include sharp corners both slowing down the process. As for the physical properties, they have been widely studied in the case of metal processes. They appear to generate some defects. The scanning vectors being unidirectional, the microstructure of the final part is severely anisotropic. As for the residual stresses, since they are related to the temperature gradients, they are mainly generated perpendicular to the scanning direction (Figure 1.13(d-e)). Getting all the same orientation, they are expected to induce cracks [52, 62, 84, 104, 115, 119, 159]. Moreover, managing to produce a good quality surface is a challenge [106]. To control these, islands strategies have been developed to increase the design possibilities [77, 115, 133, 146], getting some cracks and anisotropy reductions (further developed in the following).



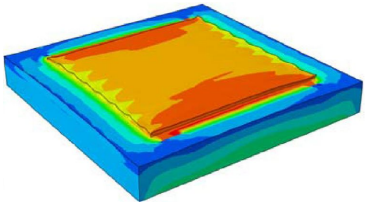
(a) Raster path [52]



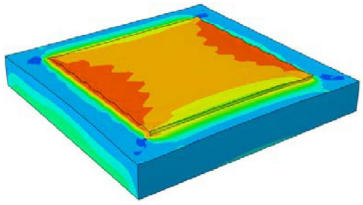
(b) Optimization of the number of jumps thanks to line orientation - direction a [158]



(c) Optimization of the number of jumps thanks to line orientation - direction b [158]



(d) Residual stresses in the X-axis [52]



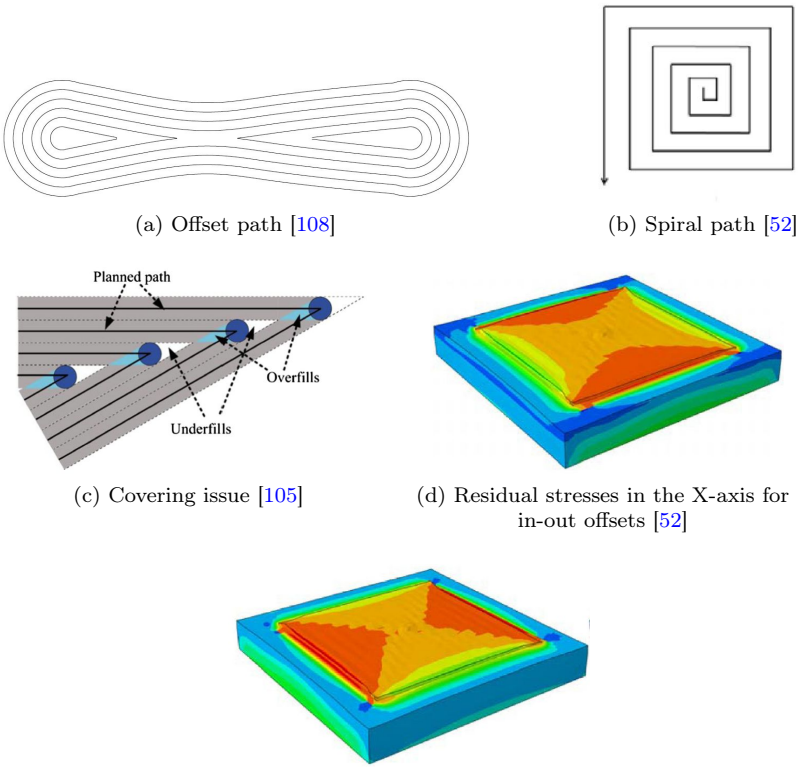
(e) Residual stresses in the Y-axis [52]

Figure 1.13: Parallel lines. (a)Exemple of raster line, (b-c)Illustration of the impact of the zigzags orientation on the number of jumps, (d-e)Impact of parallel lines on the apparition of residual stresses

Offsets

This strategy, also called contour strategy, has also widely been considered by traditional machining processes. It consists in offsets from the contour, which hatch distance can be controlled (Figure 1.14(a)). As for the parallel lines category, continuous versions have been developed linking the different offsets together and resulting in spiral scanning paths [39, 105] (Figure 1.14(b)). The filling properties of these strategies is not straightforward. The curvature of the path is not controlled and might be quite high whereas, on the other hand, some of the layer may not be covered [105, 108, 212] (Figure 1.14(c)). The curvature also impacts the productivity. Indeed, the higher it is, the slower the laser can go. As for the PBF applications, this path category leads to a crucial choice: should the heating be started from the inside, pushing the heat out of the part, or from the outside, keeping it inside. Both cases introduce

defects. Indeed, residual stresses still occur within the solid, while overheating must be taken care of by limiting the curvature [52, 62, 84, 104, 162]



(a) Offset path [108]

(b) Spiral path [52]

(c) Covering issue [105]

(d) Residual stresses in the X-axis for in-out offsets [52]

(e) Residual stresses in the Y-axis for in-out offsets[52]

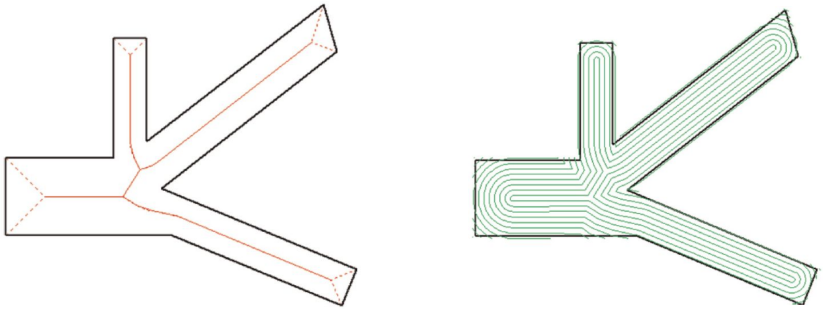
Figure 1.14: Offset lines. (a)Offset, (b)Spiral, (c)Covering issues, (d-e)Impact of offsets on the apparition of residual stresses with the laser starting from inside to go outside

Medial Axis Transform (MAT)

The Medial Axis Transform has been mainly developed for welding and material deposition additive manufacturing techniques [71, 72]. This method is actually included in the offsets strategies. Yet, instead of the part's boundary, the offsets are here based on the part's skeleton (medial axis). The method consists first to determine the medial axis of the part, that is to say, "the loci of centers of locally maximal balls inside a part" [71], before generating the offsets (Figure 1.15). On the contrary to contour paths, the resulting strategy is quite smooth, thus avoiding over and under heating/ filling. However, the path is stopped each time it meets the part boundary which results in poor surface quality or in post processes [71, 108]. To deal with this, other scanning parameters have been optimized in order to vary the size of the offsets [74]. To the authors' knowledge, no tests of this scanning method have been realized in PBF context. Note that such a strategy has also been developed in structural optimization and especially for design under thickness constraints [141].

Fractal

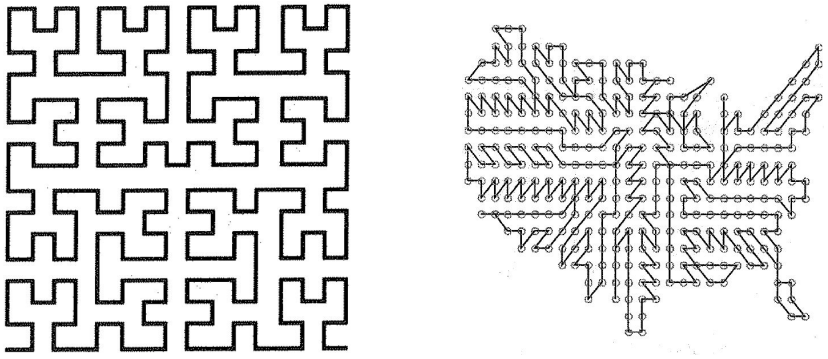
In order to break with classic parallel lines and offsets strategies, and to provide a new type of continuous path, fractal paths were developed (Figure 1.16). First using Hilbert curves to be then extended [34, 45], they have been evaluated in the PBF context. These shapes induce several modifications of the scanning direction. If such a path can reduce the residual stresses, its design must be carefully chosen to avoid excessive temperature gradients[45, 135]. As for kinematics considerations, this strategy involves an excessively high number of sharp corners which extremely increases the scanning time [73].
Getting rid of the non intersection constraint, strategies have also been developed based on points linking [150, 194]. Different discrete optimization and graphs strategies were then run to get new continuous



(a) Medial axis of the shape (b) Scanning path induced

Figure 1.15: Medial Axis Transform lines [71]

paths. These methods have been mostly developed for welding or metal deposition and no physical conclusions have been given regarding the PBF context.

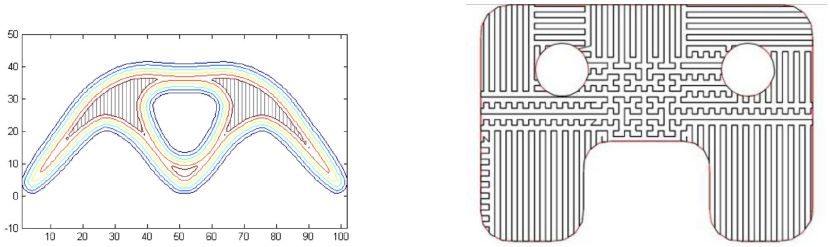


(a) Hilbert curve [34] (b) Discrete points linking [194]

Figure 1.16: Continuous fractal lines

Hybrid

Each of the proposed strategies have different advantages regarding the part quality or the ease of scanning. Hence, mixing them might result in a better efficiency (Figure 1.17). Indeed for example, the straight lines are very easy to scan but give a poor surface quality. Keeping this strategy for the inner part of the part and using offsets around the boundary leads to a quality improvement without much change in the scanning time. First tested with parallel lines and offsets [73, 80, 105, 106, 123, 129], the strategy has been optimized to get the best ratio between both path patterns [129]. Then, it has been extended to the other categories [109].



(a) Hybrid path with offsets and zigzags [129] (b) Hybrid path with fractals and zigzags [109]

Figure 1.17: Hybrid lines

Level-set

First developed in the context of steered fiber composites, scanning paths optimization based on level set methods have been newly adapted to the additive manufacturing context [51]. The strategy consists in matching the scanning path to the level sets of an arbitrarily fixed function called the level set function. Adapted to scanning strategies, it has been combined to mechanical computations (with the inherent strain method) to optimize the level set function and meet the quality requirements [51]. Finally, thanks to multi level sets techniques, concurrent optimization of the part shape and scanning path can be carried out [51] (Figure 1.18). However, the kinematics is not explicitly involved. Indeed, the inherent strains method the path optimization is based upon is based on steady state considerations. To the authors knowledge, this promising strategy has not been experimentally tested.

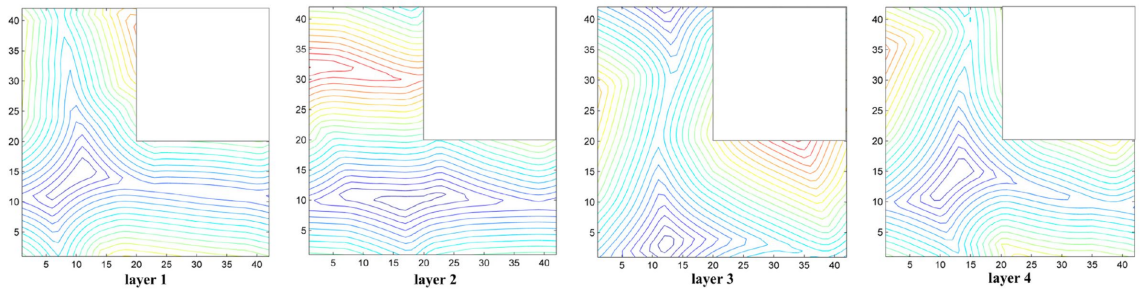
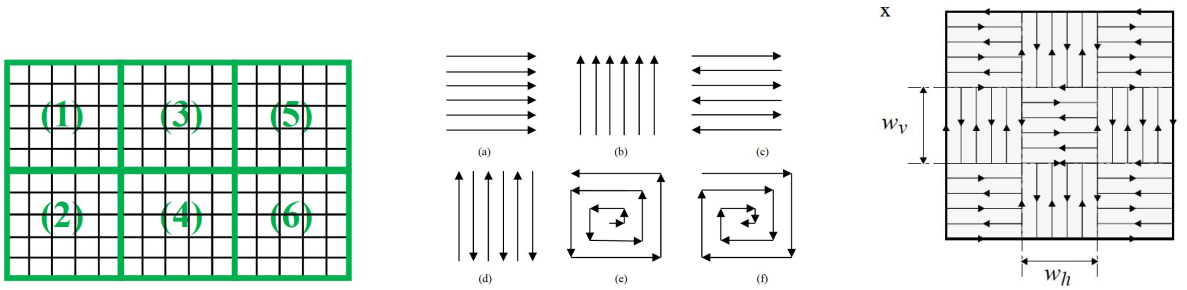


Figure 1.18: Optimized scanning strategy for four consecutive layers regarding residual stresses, based on level set optimization [51]

1.3.5 Islands scanning path strategies

Each of the previous strategies has been discussed considering that the whole part was to be built "in a row". However, many researches have developed islands or checkerboard strategies, authorizing the path to differ from one zone to another (Figure 1.19). Thus, the layer to build is split into many distinct zones to which are affected different strategies: from different orientations [104, 162, 168, 173] to different strategy types chosen thanks to genetic algorithms matching cell and path [5, 146]. The islands themselves have finally been optimized (Figure 1.17)



(a) Splitting of the part's mesh into 6 zones [5]

(b) Strategies to be put in the different zones [5]

(c) Checkerboard strategy [104]

Figure 1.19: Checkerboard strategy

To end with path optimization, additional topics must be mentioned. PBF processes require a scanning path for each layer and methods have been looked for to deal with this, would it be through a full optimization (Figure 1.18) or simply by strategy rotation. Finally, online path control is currently under study, with the development of several captors to characterize the part in building[178, 206]. Using data and models to store most of the physical computations and especially the time consuming ones, data acquisition can then be realized during the process to then proceed to online computations and provide a feedback control of the scanning process.

1.4 Design to manufacturing

Additive manufacturing allows for far more complicated designs than traditional machining. It is thus a building process matching well topology and shape optimization fields which results might be intricate. However, this new process still induces manufacturing constraints that must be included in the design process. Among them, the overhanging, the mechanical quality and the infill design issues have mostly drawn the attention [57, 66, 126].

Overhanging design

Already developed (see Section 1.2.3), this first issue is intimately related to topology optimization. Indeed, to allow for a non collapsing manufacturing, two solutions are considered: adding a support to be removed afterwards or modifying the part's design to avoid any overhangs.

Built part quality constraint

In addition to overhang issues, the quality constraints mentioned in Section 1.2 can be considered. Indeed, it is quite clear that, depending on the technology, some shapes get built easily whereas others might end up tending to crack. Due to different physical phenomena, this issue is hard to be tackled all in once and is usually split. A brief overview of the different works is now given.

The anisotropy has been considered in different studies. As mentioned (see Section 1.2.3), it has been considered by [127], using a concurrent optimization of the path and shape. This work has been followed by [66] with the additional possibility to choose the infill shapes.

The residual stresses have also been studied, and different solutions have been thought about. The first one consists in adding thermal support to the structure. Indeed, this would help the heat to spread and control high temperature gradients [11, 28, 53, 185]. A second technique consists in optimizing the shape itself to minimize, layer after layer, these stresses [16]. The third one, already pointed out, consists in a concurrent optimization of the scanning path and the layer shape [51].

As for the other defects, they mainly occur at a microscopic scale. This makes them quite hard to control through the global part's shape, which is a macroscopic parameter. However, the part to build can be designed with a porous structure and many works focus on lattice structures optimization[89].

The path as an additional optimization variable

If many works focus on the influence of the scanning strategy on the final part's properties whereas many others consider using topology optimization, very few exist interweaving them. To the authors knowledge, [127, 51] are the only existing works regarding this issue.

1.5 Problematic of this work

This overall view of the powder bed fusion works leads to a first assessment: if many parameters are involved in the different issues related to this manufacturing process, the scanning strategy strongly affects both the final quality and the manufacturing time. This analysis has lead to several studies aimed at evaluating the scanning patterns' efficiency and at controlling the spot speed and power.

These studies come out on scanning parameters optimization. New works, mostly experimental, learn and quantify from data the impact of the scanning speed, source power and energy deposition rates. As for optimization of the path itself, many works focus on optimizing parameters to improve the existing patterns (modifications of the hatch space, starting point or scanning orientation for example). Finally, researches focus layer decomposition with optimal matching between cells and patterns.

However, most of these studies rely on already existing patterns and very few works consider optimizing the path shape itself. Two recent works must be mentioned here. The first one [6, 7] focuses on scanning path control and aims at building different geometries while limiting the residual stresses. To this end, a thermal model only is considered supplemented by the assumption that the residual stresses mainly come from temperature gradients that must be limited. A second assumption consists in considering that the

melted surface around the source remains keeps the same dimensions along the scanning. Thus, to insure that the chosen path correctly melt the path's geometry, a geometric covering constraint is introduced. An analysis of the system's controllability is provided in [7] supplemented by promising numerical tests in [6].
The second work [51], has been already mentioned and is based on the level set method. The scanning path is given by a function's level sets. This function is optimized to meet residual stresses objective computed thanks to the inherent strain method. On the contrary to [6, 7], the full thermo-mechanical problem is taken into account, but based on a steady state model [124]. Using multi-level sets techniques and topology optimization theory, a concurrent optimization between the shape and the path is then performed.

As for topological optimization, if many researches have been conducted to deal with new manufacturing issues, very few focus on including the scanning path as a parameter affecting the design ([51, 127]). Yet, it is quite clear that both are strongly related. Indeed, without any physical consideration and whatever pattern is chosen, the path length, curvature and thus the scanning time already depend on the shape. As for the part quality, it is quite clear that both are related.

These issues being quite substantial, why aren't there more studies regarding them? The first answer lies in the actual available technologies. An accurate control of the energy source remains complicated leading the printers makers to favor traditional patterns. Most of the PBF machines can only follow the main scanning paths categories. A second answer lies in the lack of information on the process. Indeed, the sensitivity of the design to the path shape has not been studied, nor the sensitivity of the path shape to the design. If different patterns have been focused on, it did not bring any knowledge on how their performance could be related to the shape to build. Would a pattern applied to the scanning of a specific part be still efficient in the scanning of a different part? How to define a good scanning path? And how characterizing an "easily built shape"?

This analysis strengthen the objectives of this work. The first objective consists in optimizing the path itself, especially its shape and source's power. Using a thermal model, the goal is to develop an optimization framework to deduce new scanning strategies. The results can yield some intuitions on the problem and hints at good scanning patterns. The second objective consists in concurrent optimization of the structural shape and the scanning path. The goal is, once again, to have a better understanding of the relationship between shapes and paths, leading to possible new design constraints.

CHAPTER 2

ALGORITHMS FOR CONSTRAINED OPTIMIZATION

Contents

2.1 Introduction . . . 45
2.2 Descent methods . . . 46
2.2.1 Direction . . . 47
2.2.2 Descent step . . . 48
2.2.3 Stopping criteria . . . 49
2.2.4 Choosing the optimization algorithm . . . 49
2.3 Unconstrained first order algorithms . . . 50
2.3.1 Optimality conditions . . . 50
2.3.2 Gradient descent method . . . 51
2.3.3 Projected gradient descent algorithm . . . 51
2.4 Constrained first order algorithms . . . 51
2.4.1 Optimality conditions . . . 51
2.4.2 Lagrangian duality . . . 53
2.4.3 Lagrangian algorithm . . . 56
2.4.4 Penalization algorithm - Augmented Lagrangian method . . . 57
2.4.5 Null space gradient flow . . . 58
2.5 Unconstrained second order algorithms . . . 59
2.5.1 Newton descent algorithm . . . 59
2.5.2 Algorithms adapted to least-square problems . . . 60
2.6 Non differentiable optimization . . . 60
2.6.1 Optimality conditions regarding non differentiable convex functions . . . 61
2.6.2 Proximal operator . . . 62
2.6.3 Proximal algorithm . . . 63
2.6.4 Proximal gradient algorithm . . . 64
2.6.5 Combining the Augmented Lagrangian and proximal gradient methods . . . 64
2.7 Conclusion . . . 65

2.1 Introduction

This chapter is a review of some algorithms in continuous optimization. Consider a Hilbert space $(X, \|.\|_X, (.,.)_X)$ where $(.,.)_X$ is a scalar product on X and $\|.\|_X$ the associated norm (X could also be a Banach space yet complicating the resolution). Let $K \subset X$ be a subset and $f : X \to \mathbb{R}^+ \cup \{+\infty\}$. Optimizing the function f on the subset K consists in solving the optimization problem (2.1.1).

$$\min_{x \in K} f(x). \tag{2.1.1}$$

The subset $K \subset X$ is called the admissible domain and is defined by the constraints the solution must fulfill. The function f is the objective function, the cost that must be decreased.

This general optimization problem raises two main questions:

- does the solution exist and is it unique? Among the several problem-dependent conditions, convexity of the admissible domain and a lower bound on the objective function seem to be the easiest [10, 170]. We do not address this issue in this work. The interested reader can refer to [7] for existence and uniqueness of optimal path in the optimal control context. As for shape optimization, a general theory is given in [9, 97].

- How, from an initial guess x^0, can we improve our choice and get as close as possible to a hypothetical solution? This question is crucial in applied fields. Indeed, many industrial issues focus on improving solutions and thus on minimization algorithms. The challenges lie in matching optimization problems and algorithms, in ensuring the low computational costs and convergence of the guess sequence, in controlling the accuracy of the final guess and its satisfaction of the constraints.

In this work, we aim at proposing scanning path optimization algorithms and at using the obtained results to better understand their sensitivity to the modeling assumptions and to the part's shape. In order to efficiently determine optimizing sequences, the choice of adapted optimization algorithms is crucial. Along the manuscript, varying problems are run into, resulting in the use of different methods. This chapter aims at giving some background and to ease the understanding. To begin with, Section 2.2 introduces the concept of descent algorithms, intended to generate optimizing sequences converging to one of the minima. These methods are especially relevant in this work which objective is not to find the exact minimum but to improve the initial guess. The remaining is a presentation of the different descent algorithms. In Sections 2.3 and 2.4 any function involved in the cost or constrained is assumed to be differentiable, constituting first order algorithms. Whereas the problem is unconstrained in Section 2.3 (the admissible domain K corresponds to the whole Hilbert space X), it is more restrictive in Section 2.4 and the satisfaction of the constraints must be taken into account. In Section 2.5, an order of regularity is added and second order algorithms, focusing on twice differentiable functions, are given. Finally, Section 2.6 presents algorithms developed in case of convex non differentiable functions.

2.2 DESCENT METHODS

Optimization algorithms intend to, starting from an initial point x^0, build minimizing sequences. In the particular case of descent methods, these sequences must be decreasing the objective function [170]. The objective is thus to build a sequence $(x^k)_{k\in\mathbb{N}}$, initialized at $x^0 \in K$ such that $\forall k \in \mathbb{N}$,

$$x^{k+1} = x^k + s^k d^k, \tag{2.2.1}$$

where $s^k \in \mathbb{R}$ a step, $d^k \in X$ an update direction, satisfying, $\forall k \in \mathbb{N}$,

$$\begin{cases} f(x^{k+1}) \leq f(x^k), \\ x^{k+1} \in K. \end{cases} \tag{2.2.2}$$

Algorithm 2.1 gives the general formulation of descent algorithms.

```
initialize k = 0, s^0, x^0
compute the objective function f(x^0) and the descent direction d^0
while the stopping criteria is not reached do
    compute x^{k+1} = x^k + s^k d^k
    compute the new descent direction d^{k+1}
    k = k + 1
end
```

Algorithm 2.1: Descent algorithm

To run this very general algorithm, three main issues must be considered: the determination of a direction d^k to ensure the descent and the constraints satisfaction, the stopping criterion and the step s^k. Before detailing each of them, Definition 2.1 recalls the notion of differentiability.

Definition 2.1. Let X be a Hilbert space, $x \in X$ and f defined on a neighboring of x with values in $\mathbb{R}$. The function f is Fréchet differentiable at x if there exists $Df_x : X \to \mathbb{R}$ linear form (linear, continuous, taking values in $\mathbb{R}$) such that

$$f(x+w) = f(x) + Df_x(w) + o(w), \quad \lim_{w\to 0} \frac{|o(w)|}{\|w\|_X} = 0. \tag{2.2.3}$$

Note that this linear form depends on the point x. If this property holds for any point $x \in X$, the function f is Fréchet differentiable. Since the space X is a Hilbert space, the gradient of the function

can then be defined such that

$$\nabla f : \left\{ \begin{array}{ccc} X & \to & X \\ x & \mapsto & \nabla f(x) \end{array} \right. \tag{2.2.4}$$

where, $\forall x \in X$, the element $\nabla f(x)$ is uniquely defined by Riesz theorem as satisfying

$$\forall w \in X, \quad Df_x(w) = (\nabla f(x), w)_X. \tag{2.2.5}$$

If there exists other notions of differentiability (and especially Gâteau differentiability), in the following a function is said differentiable if it is Fréchet differentiable.

2.2.1 Direction

In descent algorithms, the minimizing sequence is built iteratively through the repeated determination of update directions. From point to point, the guess solution is improved. In order to correctly build these directions, two main properties must be taken into account: the descent condition (2.2.2) and the constraints satisfaction, requiring that the guess belongs to the admissible domain K [170].

Descent direction

The first property to fulfill is the descent condition: it must induce a decrease of the objective function f (Definition 2.2).

Definition 2.2. Let $x \in X$. A direction $d \in X$ is called descent direction at x if there exists $\eta > 0$ such that, $\forall s \in]0, \eta]$, $f(x + sd) \leq f(x)$.

This definition can be translated in terms of directional derivative: in case of infinitesimal variations in a descent direction, the function will decrease. Considering that the function f is differentiable, this leads to a new characterization, illustrated by Figure 2.1 (b):

Proposition 2.1. *Let $f : X \to \mathbb{R}$ be a differentiable function, $x \in K$. The direction d is a descent direction at x if and only if*

$$(\nabla f(x), d)_X \leq 0. \tag{2.2.6}$$

Of course, at a point x, choosing $d = -\nabla f(x)$ leads to the highest decrease (assuming a well chosen step). However, other options are available and might, depending on the problem, prove to make the sequence converge more efficiently.

Admissible direction

The direction must also fulfill another condition. Indeed, choosing the best descent direction leads to the minimization of the objective function over the Hilbert space X. Yet, in case of constrained problems ($K \neq X$), the optimizing sequence and especially the minimum must belong to the admissible domain K. Thus, a second notion of direction is given (Definition 2.3): the admissible direction.

Definition 2.3. Let $x \in K$. A direction $d \in X$ is called admissible direction at x if there exists $\eta > 0$ such that, $\forall s \in]0, \eta]$, $x + sd \in K$.

This definition actually corresponds to a "descent direction" related to the constraints. If considering infinitesimal variations, this definition corresponds to the tangential cone to the constraints (Definition 2.4).

Definition 2.4. Let $x \in K$. The tangent cone to the subset K at x is

$$T_x(K) = \left\{ \begin{array}{c} v \in X : \exists (v_n)_{n\in\mathbb{N}} \in X^{\mathbb{N}}, (\epsilon_n)_{n\in\mathbb{N}} \in (\mathbb{R}_+^*)^{\mathbb{N}}, \quad \lim_{+\infty} v_n = v, \quad \lim_{+\infty} \epsilon_n = 0, \\ \forall n \in \mathbb{N}, \quad x + \epsilon_n v_n \in X. \end{array} \right\} \tag{2.2.7}$$

This cone can be seen as the closure of the admissible directions space, containing any vector such that an infinitesimal move in its direction is still in the admissible domain K. From Definition 2.4, the tangential cone is closed and, if the point x belongs to the interior of the admissible space $x \in \overset{\circ}{K}$, $T_x(K) = X$. An illustration of this object is given by Figure 2.1 (c)

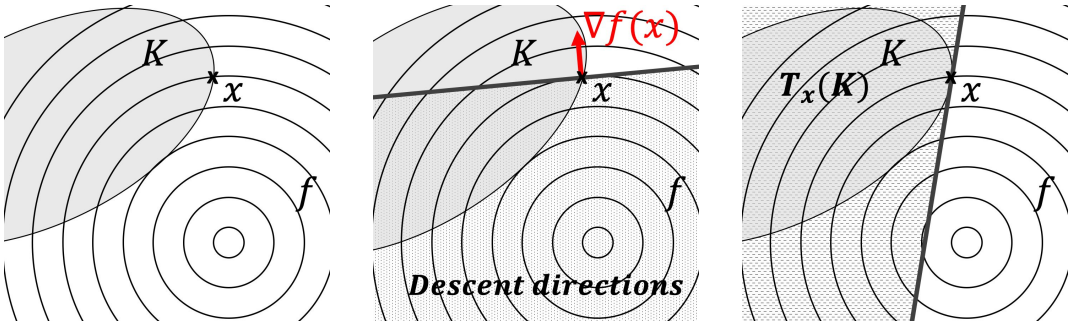
Note that the descent and admissible directions spaces are different (see Figure 2.1) and might even not intersect (Figure 2.1 (e)). These conditions and their impact on the implications on the optimization algorithms are further developed.

Optimality condition

In case of differentiability, these definitions lead to a first general optimality condition, helpful to characterize a potential solution, to determine the good update directions and a relevant stopping criterion. This condition is illustrated by Figure 2.1 (d) and (e).

Proposition 2.2. *Let $f : X \to \mathbb{R}$ differentiable and x^* a solution of the minimization of f on an admissible domain K. Then,*

$$\forall v \in T_{x^*}(K), \quad (\nabla f(x^*), v)_X \geq 0. \tag{2.2.8}$$



(a) Settings (b) Descent directions at point x (c) Tangential cone at point x

(d) Existence of $v \in T_x(K)$ such that $(\nabla f(x), v)_X < 0$ - optimality condition unsatisfied

(e) Non existence of $v \in T_{x^*}(K)$ such that $(\nabla f(x^*), v)_X < 0$ - optimality condition satisfied

Figure 2.1: Illustration of the optimality condition in $\mathbb{R}^2$

2.2.2 Descent step

In addition to a direction d, the magnitude of the update along this direction must be chosen. This parameter, called the step s is crucial [170]. If too big, the optimizing sequence could not converge. If too small, the number of required iterations might explode as well as the computational time or even generate a sequence converging to a non optimal value. Theoretical bounds on the step, called Wolfe conditions [152, 170], are a first tool to address this issue. A second approach consists in looking for the optimal step s^* at each iteration k:

$$s^* = \arg\min_{s \in \mathbb{R}} f(x + sd).$$

Yet, both techniques require differentiating the objective functions, which easily get expensive. In this work, none of these methods is used in aid of a line search approach. At each iteration, the step s^k depends on the direction d^k and on a coefficient C_s^k such that, $\forall k \in \mathbb{N}$,

$$s^k = C_s^k F_{step}(d^k), \tag{2.2.9}$$

with F_{step} fixed. The coefficient C_s is initialized to C_s^0 and updated depending on the success of each iteration. If, up to a tolerance tol, the new optimization variable guess improves the objective function then the coefficient C_s is increased and the iteration is accepted. Else, it is decreased (and so is the step) and the iteration rejected (no update of the optimization variable). The increase and decrease

respectively correspond to $C_s^{k+1} = \max\left(C_s^0, \eta_{\text{acc}}^s C_s^k\right)$ and $C_s^{k+1} = \eta_{\text{ref}}^s C_s^k$ with $\eta_{\text{acc}}^s = 1.2$ and $\eta_{\text{ref}}^s = 0.6$. The tolerance is initialized to 2 and multiplied by 0.9 every 50 iterations. Algorithm 2.2 illustrates this line-search algorithm. As for the function F_{step}, it is problem dependent and will be made precise in Chapters 6, 7, 8 and 9.

2.2.3 Stopping criteria

Although the minimizing sequence is built to converge, it is very unlikely to reach the potential minimum in a reasonable number of iterations. To choose an approximation, a stopping criterion must be designed in consistence with the guess' relevance. Three main strategies exist:

- based on the stagnation of the iterate x, with ϵ a fixed threshold: $\|x^{k+1} - x^k\|_X \leq \epsilon$.
- based on the stagnation of the objective function f, with ϵ a fixed threshold: $|f(x^{k+1}) - f(x)^k| \leq \epsilon$.
- based on the gradient smallness at the iterate x (in case of differentiability), with ϵ a fixed threshold: $\|\nabla f(x^k)\| \leq \epsilon$. In case of a constrained problem ($K \neq X$), the gradient is replaced by its projection on the admissible space.
- based on a maximum number of authorized iterations.

In this work, the number of iteration is bounded by N_{MAX} supplemented by a condition on the step coefficient C_s: if the step coefficient gets to small ($C_s < \epsilon$), the optimization is stopped (see Algorithm 2.2). This criterion actually corresponds to the stagnation of the iterate. Indeed, the step adaptation process approximately corresponds to an optimal step strategy. A strong decrease of the step coefficient means that the next guess is close to the actual one and that the iterates stagnate.

initialize $k = 0$, C_s^0, x^0 and a tolerance tol
compute the objective function $f(x^0)$ and the descent direction d^0
while $C_s^k \geq \epsilon$ *and* $k \leq N_{\text{MAX}}$ **do**
  compute $s^k = C_s^k F(\|d^k\|)$
  compute $x^{k+1} = x^k + s^k d^k$
  compute the new objective function $f(x^{k+1})$
  if $f(x^{k+1}) < f(x^k) * tol$ **then**
    iteration accepted
    compute the descent direction d^{k+1}
    $C_s^{k+1} = \max(\eta_{\text{acc}}^s C_s^k, C_s^0)$
  end
  else
    iteration rejected
    $x^{k+1} = x^k$
    $d^{k+1} = d^k$
    $C_s^{k+1} = \eta_{\text{ref}}^s C_s^k$
  end
  $k = k + 1$
end

Algorithm 2.2: Descent algorithm with line search

2.2.4 Choosing the optimization algorithm

As mentioned, optimizing a problem under constraints finally amounts to compromising between both admissible and descent directions, leading to several specific algorithms. Choosing one requires an analysis of the problem, with a special focus on three main features.

Objective function's regularity

A first feature is in the regularity of the objective function. To start with, continuity, which has a huge impact on existence conditions. Then, the Fréchet differentiability given by Definition 2.1, which gives the existence of the gradient (working with Hilbert spaces). However, this might not be enough to

ensure the convergence of the algorithms. The stronger notion of Lipschitz gradient functions is given by Definition 2.5 [170]. If this notion is not strong enough to certify the convergence of any optimization algorithm, it is enough for this work.

Definition 2.5. Let $f : X \to \mathbb{R}$ be a differentiable function. Its gradient ∇f is Lipschitz if there exists $L \geq 0$ such that

$$\forall u, v \in X, \quad \|\nabla f(u) - \nabla f(v)\|_X \leq L\|u - v\|_X. \tag{2.2.10}$$

Note that a function with a Lipschitz gradient belongs to $\mathcal{C}^1$ functions.

Objective function's convexity

A second feature is the objective function convexity, simplifying the existence and uniqueness conditions as well as convergence results. Proposition 2.3 recalls some properties of convex functions [10, 170].

Proposition 2.3. *Let X be a Hilbert space and $f : X \to \mathbb{R}$. If f is differentiable, then, the following assertions are equivalent:*

1. *f is convex on X,*
2. *$\forall u, v \in X,\ f(v) \geq f(u) + (\nabla f(u), v - u)_X$,*
3. *$\forall u, v \in X,\ (\nabla f(u) - \nabla f(v), u - v)_X \geq 0$.*

If f is twice differentiable, with $H[f](x)$ the Hessian matrix of f at x, the previous assertions are equivalent to

$$\forall u, v \in X,\ (H[f](u)v, v)_X \geq 0.$$

Constraints definition

Finally, a careful analysis of the constraint space is essential [170]. Two main cases are considered in this work. The first one consists of an unconstrained problem, where $K = X$. The second type is defined by equality and inequality constraints: $\exists n_{\text{eq}}, n_{\text{ineq}} \in \mathbb{N}$, $\exists h : X \to \mathbb{R}^{n_{\text{eq}}}$, $g : X \to \mathbb{R}^{n_{\text{ineq}}}$ such that

$$K = \{x \in X :\ h(x) = 0,\ g(x) \leq 0\}. \tag{2.2.11}$$

Of course, the regularity and convexity of the functions h and g must also be taken into account.

2.3 UNCONSTRAINED FIRST ORDER ALGORITHMS

The first class of algorithms focus on unconstrained problems, in which the admissible space K matches the whole Hilbert space X, with a differentiable objective function f. The descent directions are thus related to the objective function's gradient.

2.3.1 Optimality conditions

In this context, the optimality conditions given by (2.2.8) can be made precise. Indeed, the problem being unconstrained, the admissible space K actually corresponds to the entire Hilbert space X. The tangent cone definition given by Definition 2.4 becomes:

$$\forall x \in K = X, \quad T_x(X) = X. \tag{2.3.1}$$

This simplified description leads to new optimality conditions, called first order local necessary conditions [10, 170].

Proposition 2.4. *Assume that $K = X$ and that f is differentiable on X. Then, if x^* is a minimum, it satisfies*

$$\nabla f(x^*) = 0. \tag{2.3.2}$$

If the objective function is convex, this condition is sufficient to be a global minimum.

2.3.2 Gradient descent method

Among the first order algorithms, the gradient descent method consists of choosing the descent direction as the opposite of the objective function gradient:

$$d = -\nabla f(x). \tag{2.3.3}$$

Since the gradient corresponds to the steepest increase direction, the opposite direction ensures the minimization of the objective function (given a small enough step s). Note that this problem being unconstrained, this direction is also admissible. Under correct step conditions, the algorithm converges [151, 170].

Proposition 2.5. *Let f be a lower bounded function with a Lipschitz gradient of constant $L > 0$. If $\forall k \in \mathbb{N}$, the step at iteration k, s^k, satisfies $s^k < \frac{2}{L}$, then the gradient algorithm is a descent algorithm and it converges to a critical point (at which the gradient vanishes).*

2.3.3 Projected gradient descent algorithm

There exists one type of constraint for which the algorithm is very close to the unconstrained gradient descent: if the subset $K \neq X$ is convex and the corresponding orthogonal projection $\mathbb{P}_K$ is easy to compute. The projected gradient algorithm involves updating the optimization variable through

$$x^{k+1} = \mathbb{P}_K\left(x^k - s^k \nabla f(x^k)\right). \tag{2.3.4}$$

Proposition 2.6. *Let f be a lower bounded with a Lipschitz gradient of constant $L > 0$. If $\forall k \in \mathbb{N}$, the step at iteration k, s^k, satisfies $s^k < \frac{2}{L}$, then the projected gradient algorithm is a descent algorithm and it converges towards a critical point (at which the gradient vanishes).*

The use of this constrained algorithm is possible only if the projection is easy to compute. In this work, only box constraints are considered this way (if $x \in \mathbb{R}^2$, box constraints consist in $K = [x_{inf}, x_{sup}] \times [y_{inf}, y_{sup}]$).

Remark 2.1. Note that in Algorithm 2.2, none of these step guarantees are given. Indeed, the step could get out of the converging values. However, a careful setting of the tolerance *tol* and of the coefficient η^s_{acc} and η^s_{ref} allows to control this.

2.4 Constrained first order algorithms

This section focuses on constrained problems solved by first order algorithms. We consider here the admissible space

$$K = \{x \in X : h(x) = 0,\, g(x) \leq 0\}, \tag{2.4.1}$$

with $n_{\text{eq}}, n_{\text{ineq}} \in \mathbb{N}$, $h : X \to \mathbb{R}^{n_{\text{eq}}}$, $g : X \to \mathbb{R}^{n_{\text{ineq}}}$. Each function ($f$, h, g) is differentiable.

This section is organized as follows: first of all, the very general optimality condition stated by Proposition 2.2 are simplified thanks to the constraints formulation. These conditions lead to the presentation of the Lagrangian duality theory. Based on this, three algorithms are developed.

2.4.1 Optimality conditions

Let us first recall the optimality conditions.

Tangential cone computation

To account for the constraints, a characterization of the tangential cone must be derived, adapting Definition 2.4 to the admissible space description given by (2.4.1). This characterization depends on the functions h and g. These are not unique and it might exist other equality and inequality functions resulting in the same admissible subset K. The first notion to introduce is the qualification of the constraints, aimed at evaluating the relevance of functions h and g in the admissible domain and tangential cone descriptions. This property corresponds to the fact that the tangential cone can be expressed through the gradient of the constraints [170].

Before any formalization, assume that the admissible domain is defined by an equality constraint h (with $n_{\text{eq}} = 1$). Starting from a point satisfying this constraint, in order to remain in this admissible domain, this constraint should not be modified. The tangent domain is given by any direction orthogonal to the gradient of h. Assume now that the admissible domain is defined by an inequality constraint g (with $n_{\text{ineq}} = 1$) and that the starting point x satisfies this constraint. Then, $g(x) \leq 0$. If the point satisfies $g(x) < 0$, any infinitesimal modification of the point keeps the constraint satisfied and the tangential cone is constituted of the whole Hilbert space X. On the other hand, if $g(x) = 0$, any element v of the tangential cone must satisfy $(\nabla g(x), v)_X \leq 0$.
The set of active (inequality) constraints $\mathcal{I}(x)$ is defined at a point x as the constraints restraining the tangential cone, i.e.

$$\mathcal{I}(x) = \{i \in [\![1, n_{\text{ineq}}]\!] : g_i(x) \ = 0\}. \tag{2.4.2}$$

The analysis would thus lead to the condition

$$T_x(K) = \{v \in \mathbb{R} : \forall i \in [\![1, n_{\text{eq}}]\!], (\nabla h_i(x), v)_X = 0 \text{ and } \forall j \in \mathcal{I}(x), (\nabla g_j(x), v)_X \leq 0\}. \tag{2.4.3}$$

The formal qualification condition chosen here and given by Definition 2.6 is called Mangasarian-Fromowitz. The satisfaction of this condition implies (2.4.3) [30]. Definition 2.6 is however easier to handle and avoid handling degenerated cases.

Definition 2.6. The constraints are qualified at a point $x \in X$ if h and g are differentiable at x and if $(\nabla h)_{i \in [\![1, n_{\text{eq}}]\!]}$ is linearly independent and if $\exists v \in X$ such that

1. $\forall i \in [\![1, n_{\text{eq}}]\!], \quad (\nabla h_i(x), v)_X = 0,$
2. $\forall j \in \mathcal{I}(x), \quad (\nabla g_j(x), v)_X < 0.$

A sufficient condition to certify the constraints' qualification is given by Proposition 2.7 [9, 170].

Proposition 2.7. *The constraints are qualified at a point x if, at this point, the gradients of the equality and of the inequality active constraints*

$$\{\nabla h_i(x), i \in [\![1, n_{\text{eq}}]\!]\} \cup \{\nabla g_j(x),\ j \in \mathcal{I}(x)\}$$

are linearly independent.

Necessary optimality conditions

From the definition of the tangential cone are made precise the optimality conditions given by (2.2.8). To explicit their formulation, we consider the Lagrangian function,

$$\mathcal{L} \begin{cases} X \times \mathbb{R}^{n_{\text{eq}}} \times \mathbb{R}_+^{n_{\text{ineq}}} & \to & \mathbb{R} \\ (x; \lambda, \mu) & \mapsto & \mathcal{L}(x; \lambda, \mu) = f(x) + \lambda^T h(x) + \mu^T g(x). \end{cases} \tag{2.4.4}$$

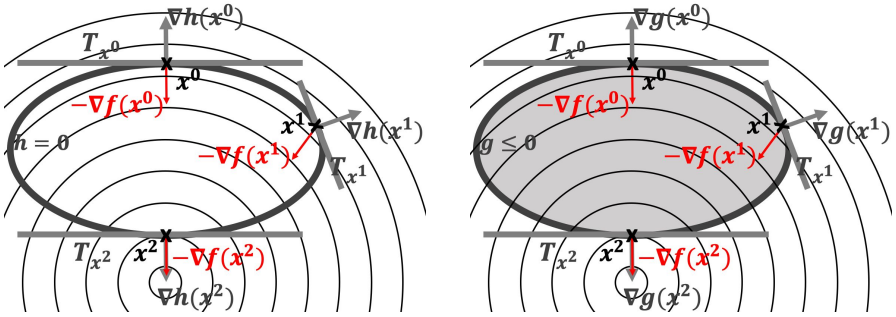
Proposition 2.8. *Let $x \in X$ admissible ($x \in K$) and assume that f, h and g differentiable at x, with the constraints qualified. If x^* is a local minimum of f on K, then, there exists $\lambda^* \in \mathbb{R}^{n_{\text{eq}}}$ and $\mu^* \in \mathbb{R}_+^{n_{\text{ineq}}}$ such that*

$$\begin{cases} \nabla_x \mathcal{L}(x^*; \lambda^*, \mu^*) = \nabla_x f(x^*) + (\lambda^*)^T \nabla_x h(x^*) + (\mu^*)^T \nabla_x g(x^*) = 0 & \\ h_i(x^*) = 0 & \forall i \in [\![1, n_{\text{eq}}]\!], \\ \left(\mu_j^*\right)^T g_j(x^*) = 0 & \forall j \in [\![1, n_{\text{ineq}}]\!], \\ \mu_j^* \geq 0 & \forall j \in [\![1, n_{\text{ineq}}]\!]. \end{cases} \tag{2.4.5}$$

If the problem is convex (h affine, g convex and f convex), this condition is also sufficient.

The proof of this theorem is given in [10, 30].

Let's give some intuition on this condition. Assume that the domain X is a two dimensional space and that there is only one constraint. In the equality case, if the gradient of the objective function is not aligned with the gradient of the constraint, this means that it is still possible to find a direction such



(a) Equality constraint: x^0 and x^2 satisfies the necessary conditions

(b) Inequality constraint: x^2 satisfies the necessary conditions

Figure 2.2: Necessary conditions illustrated for an equality and an inequality constraints, in a two dimensional domain. For equality constraints, collinear gradients of $-f$ and h is the necessary condition whereas in the inequality constraints, not only the gradients must be collinear but the scalar relating one to another must be positive

that the objective function could be decreased without modifying the constraint: the tangential cone and the vector space generated by the objective function gradient are not orthogonal (see Figure 2.2(a)). The same explanation holds in the inequality constraint case with, in addition, the requirement that the multiplier is positive (see Figure 2.2(b)).

Finally, geometrical considerations also bring intuition on the multipliers (see [58]). Indeed, following Figure 2.3, consider an equality (h) and an inequality (g) constraints and $\overline{x}$ a point satisfying the necessary condition (2.4.5) (this point is not necessarily a minimum). Let v be such that $(\nabla h(\overline{x}), v)_X = 0$ and $(\nabla g(\overline{x}), v)_X = \epsilon$, with $\epsilon \geq 0$ small and fixed. This direction corresponds to the modification of the inequality constraint only. Then,

$$(\nabla f(\overline{x}), v)_X = -\mu\epsilon.$$

Thus, the multipliers represent the sensibility of the objective function to the modification of each constraint (the sign of this multiplier is not included in this explanation, see [58]).

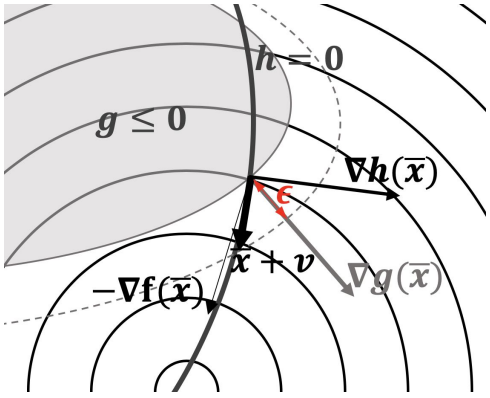


Figure 2.3: Multiplier μ giving the sensibility of the objective function f with respect of variations of the inequality constraint g, see [58]

2.4.2 Lagrangian duality

The optimality necessary conditions introduce new variables λ and μ, as well as a new function, the Lagrangian. This function is of high interest. Indeed, it includes the constraints in an explicit way, transforming the optimization problem (2.1.1) into the study of only one function defined by (2.4.4).

Primal problem

The primal function is defined from the Lagrangian function by:

$$\forall x \in X, \qquad \mathcal{P}(x) = \sup_{\lambda \in \mathbb{R}^{n_{\text{eq}}}, \mu \in \mathbb{R}_+^{n_{\text{ineq}}}} \mathcal{L}(x; \lambda, \mu). \tag{2.4.6}$$

Finding the minimum of this primal function amounts to solving the constrained optimization problem. Indeed, assume that the point $x \notin K$. Then, either there exists $i \in [\![1, n_{\text{eq}}]\!]$ such that $h_i(x) \neq 0$ or $j \in [\![1, n_{\text{ineq}}]\!]$ such that $g_j(x) > 0$ implying, since the objective function f is lower bounded, that the primal function is infinite. The minimization of the primal function is called the primal problem.

Dual problem

Instead of first maximizing the multipliers, the minimization with respect to the optimization variables could be realized, leading to the definition of the dual function.

$$\forall\, (\lambda, \mu) \in \mathbb{R}^{n_{\text{eq}}} \times \mathbb{R}_+^{n_{\text{ineq}}}, \qquad \mathcal{D}(\lambda, \mu) = \inf_{x \in X} \mathcal{L}(x; \lambda, \mu). \tag{2.4.7}$$

The dual problem involves finding

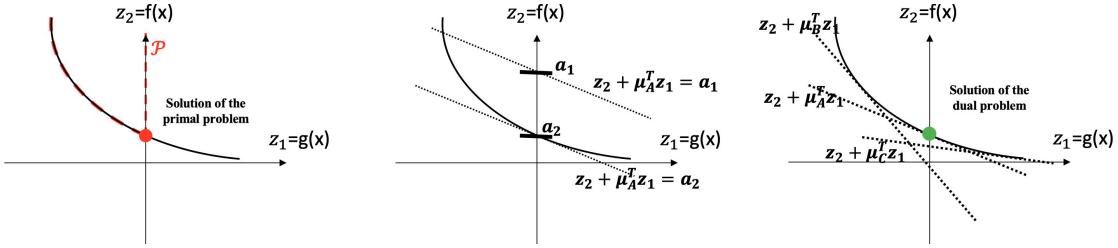
$$(\lambda^*, \mu^*) = \underset{\lambda \in \mathbb{R}^{n_{\text{eq}}}, \mu \in \mathbb{R}_+^{n_{\text{ineq}}}}{\arg\max} \mathcal{D}(\lambda, \mu). \tag{2.4.8}$$

This dual function presents two advantages. First of all, if its computation requires a minimization, this one is unconstrained. Thus, the resolution is possible through simple descent algorithms and might even be explicit. The second advantage is that this new function is concave with respect to the multipliers, simplifying the resolution of the dual problem.

Figures 2.4, 2.5 and 2.6 illustrate the resolution of each problem in three different cases. In each of these figures, the optimization problem is

$$\min f(x), \quad \text{such that} \quad g(x) \leq 0,$$

with $g(x) : X \to \mathbb{R}$. In the graphs, the abscissa z_1 represents the constraint with $z_1 = g(x)$ and the ordinate is $z_2 = f(x)$. In each Figure, the first image shows the resolution of the primal problem. On the second one, the dual function is computed: the objective is to minimize, $\mu \in \mathbb{R}_+$ being fixed, the quantity $z_2 + \mu z_1$ with (z_1, z_2) on the curve. The objective is thus to find the straight line of slope $-\mu$ crossing the curve and minimizing the value of the intercept at the origin ($\min_{a \in \mathbb{R}^+} a = z_2 + \mu z_1$, (z_1, z_2) on the curve). The maximization of the dual function then entails modifying the slope $-\mu$ (recall that $\mu \geq 0$) so that this minimal value of the intercept at the origin is the highest. This is illustrated by the third image. In Figures 2.4 and 2.5, the solutions to the primal and dual problems correspond. Note that in Figure 2.5, the optimal multiplier found by the dual problem is 0, acknowledging the fact that the constraint is not active at the minimum. In Figure 2.6 however, one can notice the difference between the final values of the primal and dual problems.

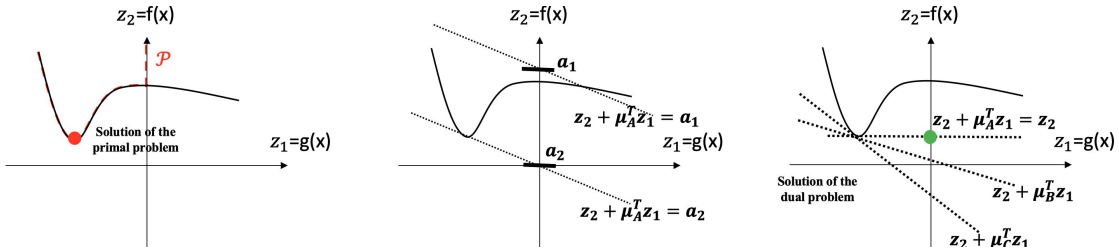


(a) Primal problem

(b) Characterization of the dual function

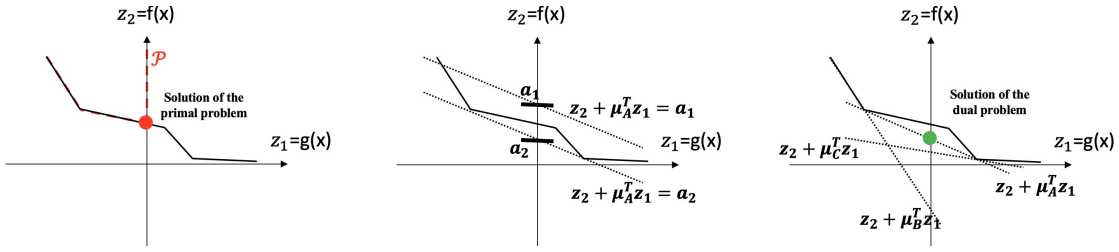
(c) Minimization of the dual function

Figure 2.4: Comparison of the primal and dual problems, $z_1 = g(x)$ and $z_2 = f(x)$. The solutions to each problem correspond.



(a) Primal problem (b) Characterization of the dual function (c) Minimization of the dual function

Figure 2.5: Comparison of the primal and dual problems, $z_1 = g(x)$ and $z_2 = f(x)$. The solutions to each problem correspond.



(a) Primal problem (b) Characterization of the dual function (c) Minimization of the dual function

Figure 2.6: Comparison of the primal and dual problems, $z_1 = g(x)$ and $z_2 = f(x)$. The solutions to each problem DO NOT correspond.

Duality principle

Because they have different characteristics and since the dual problem seems easier than the primal one, it would be convenient if they shared the same solutions. In the general case, the only result, given by Proposition 2.9 is that the value of the dual problem is smaller than the value of the primal one (see Figures 2.4 and 2.5 in case of equality and 2.6 for strict inequality).

Proposition 2.9. $\forall x \in X$,

$$\sup_{\lambda \in \mathbb{R}^{n_{\text{eq}}}, \mu \in \mathbb{R}_+^{n_{\text{ineq}}}} \left(\inf_{x \in X} \mathcal{L}(x; \lambda, \mu) \right) \leq \inf_{x \in X} \left(\sup_{\lambda \in \mathbb{R}^{n_{\text{eq}}}, \mu \in \mathbb{R}_+^{n_{\text{ineq}}}} \mathcal{L}(x; \lambda, \mu) \right). \tag{2.4.9}$$

Under conditions, it might exist points $(\overline{x}; \overline{\lambda}, \overline{\mu})$ at which (2.4.9) is an equality: the saddle-points, defined by Definition 2.7.

Definition 2.7. A point $(\overline{x}, \overline{\lambda}, \overline{\mu}) \in X \times \mathbb{R}^{n_{\text{eq}}} \times \mathbb{R}_+^{n_{\text{ineq}}}$ is a saddle-point if

$$\forall (\lambda, \mu) \in \mathbb{R}^{n_{\text{eq}}} \times \mathbb{R}_+^{n_{\text{ineq}}}, \quad \mathcal{L}(\overline{x}; \lambda, \mu) \leq \mathcal{L}(\overline{x}, \overline{\lambda}, \overline{\mu}) \leq \mathcal{L}(x; \overline{\lambda}, \overline{\mu}), \quad \forall x \in X. \tag{2.4.10}$$

Proposition 2.10. *The point $(\overline{x}; \overline{\lambda}, \overline{\mu})$ is a saddle-point of the Lagrangian function if and only if*

$$\sup_{\lambda \in \mathbb{R}^{n_{\text{eq}}}, \mu \in \mathbb{R}_+^{n_{\text{ineq}}}} \left(\inf_{x \in X} \mathcal{L}(x; \lambda, \mu) \right) = \inf_{x \in X} \left(\sup_{\lambda \in \mathbb{R}^{n_{\text{eq}}}, \mu \in \mathbb{R}_+^{n_{\text{ineq}}}} \mathcal{L}(x; \lambda, \mu) \right). \tag{2.4.11}$$

This implies that $\overline{x}$ is a solution of the primal problem and $(\overline{\lambda}, \overline{\mu})$ solution of the dual problem and that $(\overline{x}; \overline{\lambda}, \overline{\mu})$ satisfies the conditions given by Proposition 2.8.

Without any convexity assumptions, Proposition 2.10 states that the first coordinates of saddle-points constitute a subset of the minima to the primal problem. Since no information is given on the existence of these saddle-points, this subset can be empty. This is very clear in Figure 2.6: there exists a minimum

which is not a saddle-point.

In case of an admissible space reduced to convex inequality constraints only and of a convex objective function, the matching between saddle-points and minima is exact. Because the duality problem is often easier to solve, this context has led to develop optimization algorithms based on the dual. Convergence studies on some of them presented below have been mostly conducted in this specific context.

Yet, these algorithms are also used for non convex problems. Even if ensuring that the solution solves the dual problem is hard, they still generate sequences iteratively improving the guesses. Two of the main ones are now given.

2.4.3 Lagrangian algorithm

In the dual problem, the constraints have been removed at the expense of additional variables, the Lagrange multipliers, and a new maximization problem to solve.

The most natural algorithm coming out from this dual formulation is Uzawa's (Algorithm 2.3). It consists in maximizing the dual function by successive updates. This algorithm can be applied using two gradient descent algorithms: a projected one maximizing the multipliers and, in an inner loop, a second one minimizing the Lagrangian with respect to the primal variable x to compute the dual function. With $s_\lambda \in \mathbb{R}_+^{n_{\text{eq}}}$ and $s_\mu \in \mathbb{R}_+^{n_{\text{ineq}}}$ the steps related to the Lagrange multipliers, the algorithm finally is Algorithm 2.3. The choice of the steps is not made precise. As for the computation of the dual function, Algorithm 2.2 can be applied.

initialize λ^0, μ^0
initialize x^0
while *the stopping criterion is not reached* **do**
  starting from x^k, compute x^{k+1} as the minimizer of the function $x \mapsto \mathcal{L}(x; \lambda^{k+1}, \mu^{k+1})$
  computation of the steps s_λ^k and s_μ^k
  $\forall i \in [\![1, n_{\text{eq}}]\!], \quad \lambda_i^{k+1} = \lambda_i^k + s_{\lambda,i}^k h_i(x^{k+1})$
  $\forall j \in [\![1, n_{\text{ineq}}]\!], \quad \mu_j^{k+1} = \mathbb{P}_{\mathbb{R}^+}\left(\mu_j^k + s_{\mu,j}^k g_j(x^{k+1})\right)$
end

Algorithm 2.3: Uzawa algorithm

This first algorithm remains quite expensive in terms of computations. Indeed, it requires a full minimization of the unconstrained Lagrangian problem for each update of the multipliers. Simplified versions exist in which the dual function is approximated: instead of a full optimization related to the variable x, a fixed number of iterations is realized. In case the minimization over x is reduced to only one update iteration (presented in Algorithm 2.4), the algorithm is called Arrow-Hurwicz. The step management is kept general but a line search could be adapted.

initialize λ^0, μ^0
initialize x^0
while *the stopping criterion is not reached* **do**
  compute the step s^k
  compute $x^{k+1} = x^k - s^k \nabla\left(f + \lambda^k h + \mu^k g\right)(x^k)$
  computation of the steps s_λ^k and s_μ^k
  $\forall i \in [\![1, n_{\text{eq}}]\!], \quad \lambda_i^{k+1} = \lambda_i^k + s_{\lambda,i}^k h_i(x^{k+1})$
  $\forall j \in [\![1, n_{\text{ineq}}]\!], \quad \mu_j^{k+1} = \mathbb{P}_{\mathbb{R}^+}\left(\mu_j^k + s_{\mu,j}^k g_j(x^{k+1})\right)$
end

Algorithm 2.4: Alternate directions algorithm

The convergence of both Uzawa and Arrow-Hurwicz algorithms requires conditions that are not fulfilled in this work. Yet, they appear to a have good numerical behaviors with limited computational costs.

Remark 2.2. In Algorithm 2.3, each guess x does not necessarily belong to the admissible space. Indeed, if Proposition 2.10 states that the constraints are satisfied at convergence (in case the resolution of the

dual problem indeed leads to a saddle-point), it does not ensure that they are for all the minimizing sequence.

Remark 2.3. Note that in case of equality constraints only, we could also consider a new unconstrained problem involving the function $f+\lambda^T h$ with λ^T fixed. If easy, this method does not ensure the satisfaction of the constraints at convergence.

2.4.4 Penalization algorithm - Augmented Lagrangian method

In order to reinforce the importance of the constraints, a penalization π can be added to the objective function f leading to a modified cost $\tilde{f} = f + \pi$. In the Augmented Lagrangian method (AL method), the penalization corresponds to the square norm of the constraints.

Equality constraints

First consider that the problem only involves equality constraints. The penalization chosen is $\pi : x \mapsto \frac{c}{2}\|h(x)\|^2$, with c fixed penalization coefficient, leading to (2.4.12). Under conditions on the penalizing coefficients c, this new optimization problem has the same minimum than the non penalized problem [170].

$$\min_{x\in K} f_{\text{ALM}}(x) = f(x) + \frac{c}{2}\|h(x)\|^2. \tag{2.4.12}$$

The Lagrangian function of this new problem is

$$\mathcal{L}_{\text{ALM}} : \begin{cases} X \times \mathbb{R}^{n_{\text{eq}}} & \to & \mathbb{R} \\ (x;\lambda) & \mapsto & f(x) + \lambda^T h(x) + \frac{c}{2}\|h(x)\|^2 \end{cases} \tag{2.4.13}$$

and the derivatives with respect to the primal and dual variables are

$$\begin{cases} \nabla_x \mathcal{L}_{\text{ALM}}(x;\lambda) = \nabla f(x) + (\lambda + c\cdot h(x))\,\nabla h(x), \\ \nabla_\lambda \mathcal{L}_{\text{ALM}}(x;\lambda) = h(x). \end{cases} \tag{2.4.14}$$

The AL algorithm finally involves optimizing the penalized Lagrangian function $\mathcal{L}_{\text{ALM}}$ updating the multiplier at each iteration following [152]

$$\lambda^{k+1} = \lambda^k + c\cdot h(x^k).$$

Inequality constraints

Consider a problem which only involves inequality constraints. In order to get back to equality constraints, slack variables are introduced [152]. Let $x \in X$. For all active constraints ($j \in \mathcal{I}(x)$),

$$g_j(x) - s_j = 0, \quad s_j \le 0.$$

The new optimization problem consists in

$$\min_{x\in X, s\in\mathbb{R}^{|\mathcal{I}(x)|}} f(x), \quad \text{such that} \quad \forall j \in \mathcal{I}(x),\ g_j(x) - s_j = 0,\ s_j \le 0.$$

Considering the Augmented Lagrangian method under equality constraints and optimizing with respect to s gives $s_j = \min\left(g_j(x) + \frac{\mu_j}{c}, 0\right)$, $j \in \mathcal{I}(x)$. The new Lagrangian function on which working is given by $\mathcal{L}_{\text{ALM}}(x;\mu) = f(x) + \psi_{\text{ALM}}(x;\mu;c)$ with

$$\psi_{\text{ALM}} : (x;\mu;c) \mapsto \begin{cases} -\frac{1}{2}\frac{\mu^2}{c} & g(x) + \frac{\mu}{c} \le 0 \\ \mu g(x) + \frac{c}{2} g(x)^2 & \text{else.} \end{cases} \tag{2.4.15}$$

The update of the multiplier is in this case slightly different. Indeed, it corresponds to $\mu^{k+1} = \max(\mu^k + c\cdot g(x^k), 0)$ which is not the exact derivative of $\mathcal{L}_{\text{ALM}}$ with respect to μ.

Note that, in this new Lagrangian function, only the active constraints must be taken into account. Each derivative computation requires the additional preliminary step to compute the active set.

Projected Augmented Lagrangian method

Assume that the admissible space K is defined by equality and inequality constraints as well as box constraints :

$$K = \{x \in X : h(x) = 0,\, g(x) \leq 0\} \cup K^{box}, \tag{2.4.16}$$

and $\mathbb{P}_{K^{box}}$ the orthogonal projection on the box space K^{box}. Algorithm 2.5 gives the effective projected Augmented Lagrangian method used in this work.

initialize $\lambda^0 \in \mathbb{R}^{n_{\text{eq}}}$, $\mu^0 \in \mathbb{R}_+^{n_{\text{ineq}}}$
initialize x^0 and C_s^0
set $k = 0$
while *the stopping criterion is not reached* **do**
  compute the step s^k with respect to the coefficient C_s^k
  compute $x^{k+1} = \mathbb{P}_{K^{box}}\left(x^k - s^k \nabla \left(f + \lambda^k h + \mu^k g\right)(x^k)\right)$
  if $\mathcal{L}_{\text{ALM}}\left(x^{k+1}; \lambda^k, \mu^k\right) \leq tol \mathcal{L}_{\text{ALM}}\left(x^k; \lambda^k, \mu^k\right)$ **then**
    iteration accepted
    $\forall i \in [\![1, n_{\text{eq}}]\!], \quad \lambda_i^{k+1} = \lambda_i^k + c \cdot h_i(x^k)$
    $\forall j \in [\![1, n_{\text{ineq}}]\!], \quad \mu_j^{k+1} = \mathbb{P}_{\mathbb{R}^+}\left(\mu_j^k + c \cdot g_j(x^k)\right)$
    $C_s^{k+1} = \min\left(\eta_{\text{acc}}^s C_s^k, C_s^0\right)$
  end
  else
    iteration rejected
    $C_s^{k+1} = \eta_{\text{ref}}^s C_s^k$
  end
end

Algorithm 2.5: Projected Augmented Lagrangian algorithm

2.4.5 Null space gradient flow

This second algorithm favors the constraints' decrease with an exact matching of the Lagrange multipliers at each iteration. In this subsection, we describe its application to equality constraints. The explanations are fully based on [82] in which this method is expanded to inequality constraints.

Intuition on the method

We set $X = \mathbb{R}^N$ $(N \in \mathbb{N}^*)$ and we consider $K = \{x \in X : h(x) = 0\}$. The approach entails iterative updates of the variable x following

$$x^{k+1} = x^k - \Delta t \left(\alpha_f \xi_f^k + \alpha_h \xi_h^k\right),$$

with $\Delta t \in \mathbb{R}_+$ a step, $\alpha_f, \alpha_h \in \mathbb{R}_+$ arbitrary parameters and $\xi_f, \xi_h \in X$ two directions such that ξ_f decreases the objective function without impacting the constraints and ξ_h decreases the violation of the constraints.

The choice is

$$\begin{cases} \xi_f^k = \left(I_d - \nabla h(x^k)\left(\nabla h(x^k)^T \nabla h(x^k)\right)^{-1} \nabla h(x^k)^T\right) \nabla f(x^k), \\ \xi_h^k = \nabla h(x^k)\left(\nabla h(x^k)^T \nabla h(x^k)\right)^{-1} h(x^k). \end{cases}$$

Two points must be highlighted:

- the direction ξ_f is designed to leave the constraints unmodified. Indeed, for any point $x \in X$, it satisfies $\nabla h(x)^T \xi_f(x) = 0$. It thus belongs to the tangent space at x to the manifold $\{y \in X : h(y) = h(x)\}$.
- unlike the Lagrange algorithms previously presented, the Lagrange multiplier is not sought after by a gradient descent. At each iteration, it is perfectly adapted so that the constraints decrease at the rate $(1 - \alpha_h)$.

Algorithm

Every implementation details are given in [81, 82] supplemented by the corresponding python library. Note that box constraints cannot be dealt with an orthogonal projection and must be added as classic inequality constraints to fulfill the decrease requirements. Yet, for simplicity reasons, this "projection approach" is nevertheless chosen here.

Algorithm 2.6 describes the numerical process. It involves the merit function M defined at iteration k by

$$M^k(x) = \alpha_f \left(f(x) + \lambda^*(x^k)^T h(x)\right) + \frac{\alpha_h}{2} h(x)^T \left(\nabla h(x^k) \nabla h(x^k)^T\right)^{-1} h(x), \tag{2.4.17}$$

where $\lambda^*(x^k) = \underset{s\in\mathbb{R}}{\arg\min}\, \lambda \in \mathbb{R}^{n_{\text{ineq}}} \|\nabla f(x^k) + \nabla h(x^k)^T \lambda\|$.

for $k \in [\![0, N_{it}]\!]$ **do**
  compute the directions ξ_f^k and ξ_h^k
  for $l \in [\![1, N_{trials}]\!]$ **do**
    update the variable: $\tilde{x}^{k+1} = x^k - \frac{\Delta t}{2^{l-1}} \left(\alpha_f \xi_f^k + \alpha_h \xi_h^k\right)$
    project the result: $x^{k+1} = \mathbb{P}_K(\tilde{x}^{k+1})$
    if $M^k(x^{k+1}) < M^k(x^k)$ **then**
      break
    end
  end
end

Algorithm 2.6: Projected null space gradient flow algorithm

Formalization and extension to inequality constraints

For further explanations, formalization and convergence proofs, the reader is referred to [82]. If this algorithm already existed for equality constraints, [82] proposes an extension to inequality constraints. It involves the resolution of the dual problem to characterize the constraints which might get violated at the following iteration.

2.5 Unconstrained second order algorithms

In case of smoother functions, the Hessian can be computed allowing second order algorithms. We consider in this section those adapted to unconstrained problems.

2.5.1 Newton descent algorithm

In the unconstrained context and assuming the function f twice differentiable, the optimality condition is $\nabla f(x) = 0$. Thus, applying a root search on the gradient (Algorithm 2.7) should allow the determination of the points satisfying the necessary optimality condition [10, 170].

choose the initialization point x_0
while *the stopping criterion is not reached* **do**
  compute $x^{k+1} = x^k - H[f](x^k)^{-1} \nabla f(x^k)$
end

Algorithm 2.7: Newton root finding algorithm

This method can be very efficient. Indeed, it ensures a quadratic convergence as long as the initialization is not too far from the solution [170]. However, it requires regularity and computations to get a Hessian and to invert it. To remedy this second problem, alternative algorithms have been derived from Newton's, very often by approximating the Hessian. This approximation choice leads to different algorithms (among them quasi Newton, BFGS or DFP).

2.5.2 Algorithms adapted to least-square problems

The least square problems constitute a specific class for which Newton type algorithms are very useful. Indeed, the specific formulation of the objective function eases the computation of the gradient and the approximation of the Hessian [21, 107, 170].

Consider now that the objective is to minimize a least-square function f defined on $\mathbb{R}^M$, with $\forall v \in \mathbb{R}^M$, $F(v) = (F_i(v))_{0\leq i\leq N}$ and

$$f(v) = \sum_{i=0}^{N} F_i^2(v) = F(v)\cdot F(v). \tag{2.5.1}$$

The gradient and Hessian of this objective function are:

$$\nabla f(v) = \left(\partial_{v_j} f(v)\right)_{0\leq j\leq M} = \left(\sum_{i=0}^{N} 2F_i(v)\partial_{v_j}F_i(v)\right)_{0\leq j\leq M} = 2J[F]^T F,$$

$$\forall (j_1,j_2)\in [\![1,M]\!]^2, \qquad [H[f](v)]_{j_1,j_2} = 2\left[J[F]^T J[F]\right]_{j_1,j_2} + 2\sum_{i=0}^{N} F_i(v)\partial_{v_{j_1}}\partial_{v_{j_2}}F_i(v),$$

where $J[F]$ is the Jacobian matrix so that

$$\forall j\in [\![1,M]\!], \forall i\in [\![1,N]\!], \quad J[F]_{i,j} = \partial_{v_j}F_i.$$

Following a Newton algorithm, the descent direction d_v^k is chosen so that $\nabla f(v^k) + H[f](v^k)\cdot d_v^k = 0$. Approximating the Hessian by its first term $J[F]^T J[F]$, this leads to choosing a Gauss Newton direction

$$J[F]^T J[F] d_v^k = -J[F]^T F.$$

Remark 2.4. This approach actually amounts to looking for the iterate x^{k+1} as the minimizer of the least square problem applied to the linearization of F:

$$\min_y \|F(x^k) + J[F](x^k)\left(y - x^k\right)\|. \tag{2.5.2}$$

This algorithm presents one main drawback: far from the solution, the matrix $J[F]^T J[F]$ does not correctly approximate the Hessian and might be ill-conditioned. In order to solve this problem, a Levenberg-Marquardt algorithm is preferred [21, 107, 170] and the descent direction chosen is, with $\mu\in\mathbb{R}$ fixed by the user,

$$\left(J[F]^T J[F] + \mu I_d\right) d_v^k = -J[F]^T F. \tag{2.5.3}$$

The intuition given by the ill-conditioned issue can be supplemented by a trust region interpretation. Indeed, a Levenberg-Maquardt iteration amounts to solving (2.5.2) while constraining the variable y to remain in a trust domain, the parameter μ corresponding to this constraint's multiplier [170].

2.6 NON DIFFERENTIABLE OPTIMIZATION

In first and second order algorithms, the descent choices are based on the derivatives, which are not always calculable. Consider for example box constraints. They could be dealt with using differentiable inequality constraints but also as the indicator function of a convex space. This second option leads to a non differentiable function. Yet, the projection solution can be used. If involving non differentiable functions leads usually to complicated resolutions, their convexity simplifies their management. While all the theories presented above do not involve convexity, we consider in this section the following optimization problem

$$\min_{x\in K} f(x) = f_{\mathrm{D}}(x) + f_{\mathrm{ND}}(x), \quad K\subset X. \tag{2.6.1}$$

The function f_{D} is *differentiable with a Lipschitz gradient of constant L* and f_{ND} is *convex non differentiable*. The study of non differentiable optimization is here limited to the case of f_{ND} convex.

2.6.1 Optimality conditions regarding non differentiable convex functions

It is essential to quantify the sensibility of each function to small perturbations. Definition 2.8 first introduces the domain of a function.

Definition 2.8. Let $f : X \to \mathbb{R} \cup \{\infty\}$. Its domain $dom(f)$ is defined as

$$dom(f) = \{x \in X : f(x) < \infty\}. \tag{2.6.2}$$

Subgradient

The gradient being unavailable, the more general notion of subgradient is intended to represent the variations of f_{ND} at a point [58, 170].

Definition 2.9. Let $f_{\mathrm{ND}} : X \to \mathbb{R} \cup \{\infty\}$ be a convex function and $x_0 \in dom(f_{\mathrm{ND}})$. A vector $\eta \in X$ is a subgradient of f_{ND} at point x_0 if

$$\forall x \in dom(f_{\mathrm{ND}}), \quad f_{\mathrm{ND}}(x) \geq f_{\mathrm{ND}}(x_0) + (\eta, x - x_0)_X. \tag{2.6.3}$$

The subdifferential of f_{ND} at point x_0 is constituted of every subgradients of f_{ND} at point x_0 and is written $\partial f_{\mathrm{ND}}(x_0)$.

In one dimension, the subgradient consists in all the straight lines going through the point $(x_0, f_{\mathrm{ND}}(x_0))$ which do not cross the graph of f_{ND}. The straight line is replaced by an hyperplane in higher dimensions. To illustrate this, consider the Hilbert space $\mathbb{R}$ and the indicator function of $\mathbb{R}^+$

$$\chi_{\geq 0} : x \in \mathbb{R} \mapsto \begin{cases} 0 & x \geq 0 \\ \infty & \text{else.} \end{cases} \tag{2.6.4}$$

The domain of this function is $dom(\chi_{\geq 0}) = \mathbb{R}^+$. Consider $\eta \in \partial\chi_{\geq 0}(0)$. Then, $\forall x \in \mathbb{R}^+$, $\chi_{\geq 0}(x) \geq \chi_{\geq 0}(0) + \eta x$ and thus, $\eta \leq 0$. This leads to $\partial\chi_{\geq 0}(0) = \mathbb{R}^-$ (Figure 2.7). A second example is given, considering the Hilbert space $\mathbb{R}$ and the absolute value function

$$va : x \in \mathbb{R} \mapsto |x| \in \mathbb{R}^+. \tag{2.6.5}$$

Set $\eta \in \partial va(0)$. Then, $\forall x \in \mathbb{R}$, $|x| \geq |0| + \eta x$ and thus, $\eta \in [-1, 1]$ (Figure 2.8).

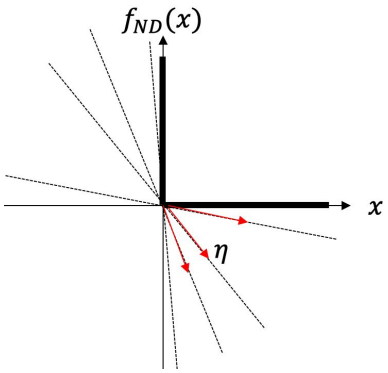


Figure 2.7: Subgradients at $x = 0$ of $\chi_{\geq 0}$

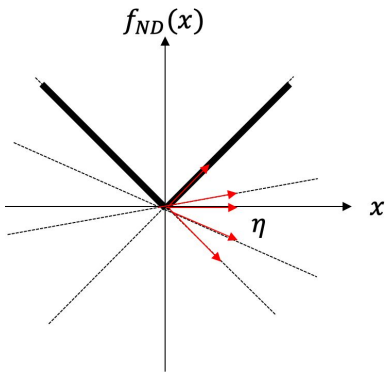


Figure 2.8: Subgradients at $x = 0$ of va

A second interpretation links this notion with the classic gradient. Indeed, the subdifferential of a function f_{ND} at point x gathers all the "steepest directions" at x: extending the range of f_{ND} to $+\infty$, any infinitesimal movement in these directions increases the function. It is worth noticing that, in case f_{ND} is differentiable at a point x_0, the subdifferential only contains the gradient: $\partial f_{\mathrm{ND}}(x_0) = \{\nabla f_{\mathrm{ND}}(x_0)\}$.

Optimality condition

Consider the unconstrained problem

$$\min_{x \in X} f_{\mathrm{ND}}(x).$$

The corresponding optimality condition involves the subgradient.

Proposition 2.11. *Let $f_{\mathrm{ND}} : X \to \mathbb{R} \cup \{\infty\}$ be a convex function. Then, $x^* \in X$ is a global minimum of f_{ND} if and only if $0 \in \partial f_{\mathrm{ND}}(x^*)$.*

PROOF.
Assume that the condition is satisfied. Then, $\forall x \in X$, $f_{\mathrm{ND}}(x) \geq f_{\mathrm{ND}}(x^*)$ which gives the minimum property. The reciprocal property comes from the minimum definition. □

2.6.2 Proximal operator

A second tool to deal with non differentiable convex functions is the proximal operator, defined as follows.

Definition 2.10. Let $f_{\mathrm{ND}} : X \to \mathbb{R} \cup \{\infty\}$ and let $\lambda_{\mathrm{P}} \in \mathbb{R}^+$. The proximal operator $\mathrm{prox}_{\lambda_{\mathrm{P}} f_{\mathrm{ND}}}$ is

$$\forall x \in X, \quad \mathrm{prox}_{\lambda_{\mathrm{P}} f_{\mathrm{ND}}}(x) = \underset{v \in X}{\arg\min}\, f_{\mathrm{ND}}(v) + \frac{1}{2\lambda_{\mathrm{P}}} \|v - x\|_X^2. \tag{2.6.6}$$

The role of this operator is to optimize f_{ND} while controlling the distance between two consecutive iterates. Indeed, if the function f_{ND} is minimized by the proximal operator computation, the second term prevents the proximal operator value to get far from the point at which it is computed. Among the different interpretations that can be given, a few are detailed in the following. For further information, the reader can refer to [157].

Projection

A very simple interpretation is illustrated by Figure 2.9 from [157]. This Figure represents the level set of a function (thin black lines) as well as its domain (thick black line). Each blue dot point in the domain is moved so that the function decreases whereas the blue dot points out of the domain are brought back at the boundary. The role of this operator is thus to minimize the function f_{ND} while staying within its domain $dom(f_{\mathrm{ND}})$.

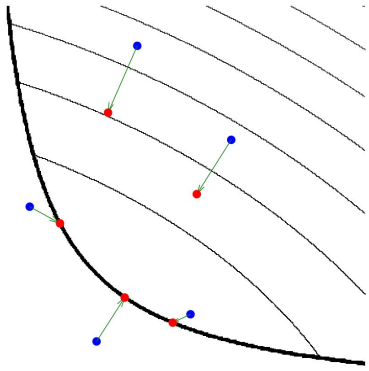


Figure 2.9: Illustration of the proximal operator effects [157]

This first interpretation can be supplemented by considering the indicator function of $\mathbb{R}^+$, $\chi_{\geq 0}$. Let's compute the proximal operator of χ. Let $x \in dom(\chi_{\geq 0}) = \mathbb{R}^+$, $\lambda_{\mathrm{P}} \in \mathbb{R}^+$.

$$\mathrm{prox}_{\lambda_{\mathrm{P}} \chi_{\geq 0}}(x) = \underset{v \in \mathbb{R}}{\arg\min}\, \chi_{\geq 0}(v) + \frac{1}{2\lambda_{\mathrm{P}}} \|x - v\|_{\mathbb{R}}^2 = \underset{v \in \mathbb{R}^+}{\arg\min}\, \|x - v\|^2 = \mathbb{P}_{\mathbb{R}^+}(x). \tag{2.6.7}$$

In the context of an indicator function, the proximal operator exactly corresponds to an orthogonal projection. The proximal operator can then be seen as a sort of projection of the point it is computed at onto a "space of decreasing f_{ND}".

Resolvent of the subdifferential operator

The proximal operator, following Proposition 2.12, can also be seen as the resolvent of the subdifferential operator.

Proposition 2.12. *Let $f_{\mathrm{ND}} : X \to \mathbb{R} \cup \{\infty\}$ be a convex function and $\lambda_{\mathrm{P}} \in \mathbb{R}^+$. Then, $prox_{\lambda_{\mathrm{P}} f_{\mathrm{ND}}} = (I_d + \partial f_{\mathrm{ND}})^{-1}$.*

PROOF.
Let $x \in X$. Then,

$$z = (I_d + \partial f_{\mathrm{ND}})^{-1}(x) \iff x \in \{z + \partial f_{\mathrm{ND}}(z)\} \iff 0 \in \tfrac{z-x}{\lambda_{\mathrm{P}}} + \partial f_{\mathrm{ND}}(z)$$
$$\iff 0 \in \partial\left(f_{\mathrm{ND}}(z) + \tfrac{1}{2\lambda_{\mathrm{P}}}\|x - z\|_X^2\right),$$

which, according to the optimality conditions given by Proposition 2.11 gives the result. □

This characterization leads, in a differentiable context, to $\mathrm{prox}_{\lambda_{\mathrm{P}} f_{\mathrm{ND}}}(x) = (I_d + \nabla f_{\mathrm{ND}})^{-1}(x) \approx x - \lambda_{\mathrm{P}} \nabla f_{\mathrm{ND}}(x) + o(\lambda_{\mathrm{P}})$. Finding the proximal operator can thus be related to gradient descent. Yet, the scheme is implicit and the direction chosen is the gradient at the updated point x^{k+1} instead of x^k as in the classic gradient descent.

Fixed point

To conclude this brief introduction to this very powerful notion, a last property states the equivalence between satisfying the optimality conditions and being a fixed point of the proximal operator.

Proposition 2.13. *Let $f_{\mathrm{ND}} : X \to \mathbb{R} \cup \{\infty\}$ be a convex function. Then,*

$$0 \in \partial f_{\mathrm{ND}}(x) \iff x = prox_{f_{\mathrm{ND}}}(x). \tag{2.6.8}$$

PROOF.
See [157]. □

2.6.3 Proximal algorithm

The proximal operator is a useful tool to deal with non differentiable convex functions: if it requires solving a minimization problem, the convexity property eases this process and depending on the function, the proximal operator can even be explicitly computed. Consider here the unconstrained minimization problem

$$\min_{x \in X} f_{\mathrm{ND}}(x).$$

Algorithm 2.8 is designed to minimize this problem. Convergence properties exist under conditions on the step λ_{P} (see [157] for further details).

```
choose the initialization x_0
while the convergence criteria is not reached do
    compute the step λ_{P,k}
    compute x_{k+1} = prox_{λ_{P,k} f_ND}(x_k)
end
```

Algorithm 2.8: Proximal algorithm

Following the mentioned interpretations, this algorithm can be read in terms of fixed point or in terms of gradient descent while controlling the distance between two iteration points [157]. A last interpretation, coming from gradient flows, is given. Assume that the function f_{ND} is differentiable and consider the equation

$$\frac{dx}{dt}(t) = -\nabla f_{\mathrm{ND}}(x(t)). \tag{2.6.9}$$

In order to numerically solve this equation, time discretization schemes are used. We consider here two different choices:

- **Euler explicit scheme**: leading to a gradient descent algorithm

$$\frac{x^{k+1} - x^k}{\Delta t} = -\nabla f_{\mathrm{ND}}(x^k) \implies x^{k+1} = x^k - \delta t \nabla f_{\mathrm{ND}}(x^k). \tag{2.6.10}$$

- **Euler implicit scheme**: leading to the proximal algorithm

$$\frac{x^{k+1} - x^k}{\Delta t} = -\nabla f_{\mathrm{ND}}(x^{k+1}) \implies x^{k+1} = (I_d + \Delta t \nabla f_{\mathrm{ND}})^{-1}(x^k). \tag{2.6.11}$$

Remark 2.5. Note that the Augmented Lagrangian method can be considered as a proximal algorithm on the dual [18]. For each iteration k,

$$(\lambda_k, \mu_k) \in \underset{\lambda \in \mathbb{R}^{n_{\text{eq}}}, \mu \in \mathbb{R}_+^{n_{\text{ineq}}}}{\arg\max} \mathcal{D}(\lambda, \mu) - \frac{1}{2c}\|\lambda - \lambda^k\|^2 - \frac{1}{2c}\|\mu - \mu^k\|^2.$$

2.6.4 Proximal gradient algorithm

The objective of this section is to minimize a function $f = f_{\text{D}} + f_{\text{ND}}$, with f_{D} differentiable non convex, with a Lipschitz gradient of constant L and f_{ND} non differentiable convex. The proximal gradient algorithm (Algorithm 2.9), combining the gradient and the proximal algorithms, ensures the convergence providing restrictions on the step $\lambda_{\text{P,k}} > 0$. In numerical applications, a line search is performed and the step $\lambda_{\text{P,K}}$ is chosen accordingly to the gradient step (Subsection 2.2.2). Note that in this algorithm, the gradient and proximal steps match.

choose the initialization x_0
while *the convergence criteria is not reached* **do**
  compute the step $\lambda_{\text{P,k}}$
  compute $x_{k+1,aux} = x^k - \lambda_{\text{P,k}} \nabla f_{\text{D}}(x_k)$
  compute $x_{k+1} = \text{prox}_{\lambda_{\text{P,k}} f_{\text{ND}}}(x_{k+1,aux})$
end

Algorithm 2.9: Proximal gradient algorithm

Proposition 2.14. *Let $(x_k)_{k\in\mathbb{N}}$ be the sequence generated by the algorithm. Then,*

- $\sum_{k=0}^{\infty} \|x^{k+1} - x^k\|^2 < \infty$,
- *any accumulation point of (x^k) is a stationary point of f,*
- *there exists $\overline{f}$ such that $\lim_{k\in\infty} f(x^k) = \overline{f}$ and, writing Ω the set of accumulation points of (x^k), $f = \overline{f}$ on Ω.*

PROOF.
The proof can be found in [195]. □

Note that if the function f_{ND} corresponds to an indicator function of a convex set, the algorithm simply amounts to a projected gradient algorithm.

2.6.5 Combining the Augmented Lagrangian and proximal gradient methods

We now focus on the full problem

$$\min_{x\in K} f(x) = f_{\text{D}}(x) + f(x), \quad K = \{v \in X,\, h(v) = 0\}, \tag{2.6.12}$$

where the equality constraint h is differentiable with Lipschitz gradient but might be non convex. The constraints are dealt with an Augmented Lagrangian method, involving the Lagrangian function:

$$\mathcal{L}(x, \lambda; c) = f_{\text{ND}}(x) + \underbrace{f(x) + \lambda^T \cdot h(x) + \frac{c}{2}\|h(x)\|^2}_{f_{\text{D}}(x)}. \tag{2.6.13}$$

In the classic Augmented Lagrangian algorithm, gradient steps on the primal and dual variables were alternating. In this new algorithm, the same method is used, replacing the primal variable update by a gradient-proximal step. The algorithm corresponds to the algorithms presented in Section 2.4.4 with the computation of the dual problem realized using a proximal gradient method: the update $x^{k+1} = x^k - s^k \nabla \left(f_{\text{D}} + \lambda^k h\right)$ (or $x^{k+1} = \mathbb{P}_{K^{box}}\left(x^k - s^k \nabla \left(f_{\text{D}} + \lambda^k h\right)\right)$ in case of box constraints) is now replaced by

$$x_{k+1} = \text{prox}_{s^k f_{\text{ND}}}\left(x^k - s^k \nabla \left(f_{\text{D}} + \lambda^k h\right)(x_k)\right).$$

A line search is used to manage the step. Note that if the non differentiable convex function is reduced to the indicator of a box, the combined algorithm exactly corresponds to Algorithm 2.5.

2.7 Conclusion

The choice of an optimization algorithm is deeply related to the properties of the objective function and constraints involved in the problem. In this chapter the algorithms are built upon optimality conditions. Depending on the regularity of each of the functions and on their convexity, these optimality conditions can be specified. Characterization of the minima are derived as necessary (and even sufficient depending on the convexity) conditions. The algorithms are then designed to look for the points satisfying these conditions.

The objective of this work is to optimize the path. Because the kinematics and the physics must be controlled, objective functions and constraints are designed accordingly. Yet, to keep the computational cost low, a simplified model of the phenomena is built in Chapter 4 and used then. To remain close to significant physical values, this model is calibrated with existing data, by minimizing a least square function. The adaptation of the Levenberg-Marquardt algorithm presented in Section 2.5.2 is detailed in Chapter 4. From this model, the path optimization itself can be conducted. If the functions involved are chosen differentiable, second derivatives may not exist or be highly difficult to compute. Thus, path optimization is dealt with using constrained first order algorithms (Section 2.4) and especially with the Augmented Lagrangian method. The null space gradient flow algorithm is also used in Chapters 6 and 7 for comparison but has not yet been implemented for the projects described in the other chapters. In Chapter 8, the power along the path is added as an optimization variable. In order to avoid degenerated results with rapid oscillations of this second variable, its total variation is considered as a constraint. Corresponding to a L^1-norm on the gradient of the variable, this function is convex but not differentiable. A combination between the Augmented Lagrangian method and the proximal gradient algorithm is thus used. Finally, for the concurrent optimization explained in Chapter 9, the Augmented Lagrangian method is once again applied.

This study of optimization algorithm is not exhaustive and several other possibilities exist. Among the different perspectives this work calls for, the test of alternative optimization methods to design the path would be interesting, as well as a wider use of the null space gradient flow algorithm. It would also be of great interest to adapt this latter to projection on box constraints and ensure the convergence of such a mixed algorithm (see Section 2.4.5).

CHAPTER 3

SHAPE OPTIMIZATION

Contents

3.1 Introduction **67**
3.2 Shape differentiation **67**
3.2.1 Shape differentiability 68
3.2.2 Usual derivatives 69
3.2.3 Case of a function depending on a PDE 71
3.2.4 From the derivative to the descent direction 72
3.3 Shape optimization of a domain **73**
3.3.1 Level set method 73
3.3.2 Application: minimization of the volume and compliance 74
3.4 Conclusion **77**

3.1 Introduction

This chapter focuses on structural optimization which is a specific case of the general minimization problems presented in Chapter 2. The optimization variable is now the structure which must be designed to decrease a cost f and to belong to an admissible subset $K \subset Z$, where Z is a Banach space. The constraints and objective function usually depend on the structure's behavior given as the solution to a partial differential equation. This type of "PDE-constrained optimization problems" arises in various applications (see [15] for more details).

Structural optimization is often divided into three categories. In *parametric optimization*, the structure is represented by a finite number of specific parameters (thickness of a membrane, elasticity coefficients or even a parametrization of the shape boundary) which are determined during the optimization process. In *shape optimization*, the structure's shape itself is modified using infinite dimensional models. In *topology optimization*, the sensitivity of the problem to the topology is added to classic shape variations.

Shape variations can be computed through different techniques. In boundary variation methods, the focus is on the structure's geometry itself. Its boundary is advected with respect to the sensitivity of the problem to its variations [9, 15, 97]. These techniques, based on Hadamard's variation method, include Lagrangian methods with re-meshing processes as well as level set methods. In homogenization methods [8, 32], the shape is optimized through a density variable, defined on the whole working domain. The sensitivity of this function with respect to the optimization problem is computed and the density is modified. Among the topology optimization methods resulting from this technique, the most famous is the SIMP method, in which the density is first optimized and then filtered in order to get a clear black and white final structure. A complete review of this domain is provided by [15].

This chapter presents shape optimization with a focus on sensitivity computations and numerical issues. In particular, existence and uniqueness issues are not mentioned and further details can be found in [10, 97]. In Section 3.2, we detail the notion of shape variation and give classic derivative formulations. Since not only the boundary of the part but also the path are seen as structures, these formulations are first given for general curves and then adapted to the boundaries of bounded sets. This section finally explains the transformation of these derivatives into gradients. In Section 3.3, we focus on parts optimization, and especially on the level set method for shape representation. A numerical test case illustrates these concepts on a classic example of shape optimization for elastic materials.

3.2 Shape differentiation

First introduced by Hadamard [95], shape differentiation theory has been widely developed [9, 15, 97]. It is often applied to open bounded set optimization, in order to measure the sensitivity of the problem

with respect to infinitesimal boundary variations. This theory can of course be expanded to more general shapes, and especially to one dimensional curves in a two dimensional domain. In this work, we consider two different types of structures: on the one hand the part to build which can be represented by an open bounded set, and on the other hand, in the steady context, the path itself represented by a simple curve.

In this section, shape differentiation theory is detailed for both applications. The adaptation of Fréchet-differentiability is first presented along with general settings. Then, the computation of classic derivatives is given as well as differentiation of solutions to partial differential equations. Finally, the regularization process that transforms the derivative into a gradient is detailed.

3.2.1 Shape differentiability

We shall use in this section the analysis of Murat and Simon [9, 15, 97, 149]. The curves considered in this chapter belong to the set $\mathcal{G}$ given by Definition 3.1, with $D \subset \mathbb{R}^2$ a bounded working domain.

Definition 3.1. Let $\mathcal{G}$ be the set made of all the $\mathcal{C}^2$ oriented curves $\Gamma \subset D$, with tangent τ, starting at point A and ending at point B. The normal is defined such that, $\forall x \in \Gamma$, the couple $(\tau(x), n(x))$ is a direct orthonormal basis. The (mean) curvature is given at each point by $\kappa(x) = \operatorname{div} n(x)$ where the normal has been extended to a neighborhood of Γ.

The set $\mathcal{G}$ contains in particular the boundaries of $\mathcal{C}^2$ open sets included in D. Indeed, let Ω be such a domain. The unit vector n is defined on the boundary $\partial\Omega$ as the normal pointing outwards. From this normal, a tangent vector can be defined at each boundary point so that (τ, n) is an orthonormal basis. Finally, there exists a point A at which the normal and tangent are continuous. This point can be set as starting point and end point of the domain boundary which thus belongs to $\mathcal{G}$ (Figure 3.1).

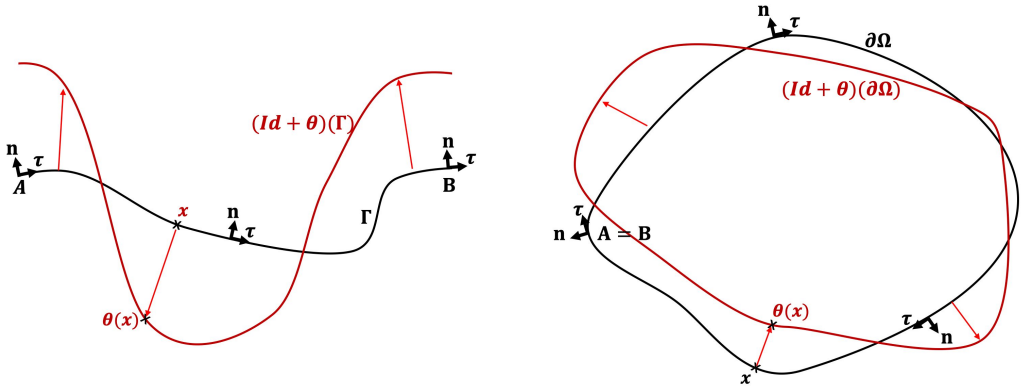
Consider a smooth reference curve $\Gamma_0 \in \mathcal{G}$. Any admissible curve is assumed to be related to the reference one through a perturbation θ such that (Figure 3.1)

$$\Gamma = \{x + \theta(x) \text{ such that } x \in \Gamma_0\}.$$

The curve's deformation is limited by the vector field's regularity. To keep the $\mathcal{C}^2$ property of the curves, we choose here $\theta \in \mathcal{C}^2\left(\overline{D}, \mathbb{R}^2\right)$. Note that since the working domain D is bounded, each element belonging to $\mathcal{C}^2\left(\overline{D}, \mathbb{R}^2\right)$ is also bounded.

Remark 3.1. Choosing $\mathcal{C}^2$ curves provides the existence of the curvature κ. Yet, part of the theory developed below can be adapted to domains with less regularity. For example, the set $\mathcal{G}$ could be extended to piecewise $\mathcal{C}^2$ curves [9, 15, 97].

Remark 3.2. Non-optimizable zones can be included in the definition of general curves. The corresponding restrictions are applied to the vector field set, leaving the theory unchanged.



(a) Curve $(Id + \theta)(\Gamma)$ obtained from the reference curve Γ

(b) Domain $(Id + \theta)(\Omega)$ obtained from the reference domain Ω

Figure 3.1: Deformation of two elements of $\mathcal{G}$: (a)Γ and (b)$\partial\Omega$

A notion of differentiability with respect to the curve Γ is derived [9, 15, 97]:

Definition 3.2. A functional $J : \mathbb{R}^2 \to \mathbb{R}$ is said to be shape differentiable at $\Gamma_0 \subset \mathcal{G}$ if the application $\theta \to J\left(\left(Id+\theta\right)\left(\Gamma_0\right)\right)$ is Fréchet-differentiable at 0 in the Banach space $\mathcal{C}^2\left(\overline{D}, \mathbb{R}^2\right)$, i.e.

$$J\left(\left(Id+\theta\right)\left(\Gamma_0\right)\right) = J(\Gamma_0) + DJ(\Gamma_0)(\theta) + o(\theta) \quad \text{with} \lim_{\theta \to 0} \frac{|o(\theta)|}{\|\theta\|_{\mathcal{C}^2\left(\overline{D}, \mathbb{R}^2\right)}} = 0, \tag{3.2.1}$$

where $DJ(\Gamma_0)$ is a continuous linear form on $\mathcal{C}^2\left(\overline{D}, \mathbb{R}^2\right)$ and $\|.\|_{\mathcal{C}^2\left(\overline{D}, \mathbb{R}^2\right)} = \sum_{|\alpha| \leq 2} \|\partial^\alpha .\|_{L^\infty(\overline{D})}$.

3.2.2 Usual derivatives

The goal is now to compute shape derivatives. Before getting any further, we give the notions of tangential gradient and tangential divergence along the curve [15, 97].

Definition 3.3. Let $\Gamma \in \mathcal{G}$.

- Let $f \in \mathcal{C}^1\left(\Gamma, \mathbb{R}\right)$ be a real-valued function. Its *tangential gradient* $\nabla_\tau f : \Gamma \to \mathbb{R}^2$ is defined by

$$\nabla_\tau f = \nabla \tilde{f} - \nabla_n \tilde{f} = \nabla \tilde{f} - \partial_n \tilde{f} n \, (= \partial_\tau f \tau),$$

  with $\tilde{f}$ any extension of f to an open neighborhood of Γ (the definition is independent from the extension). Note that since f takes values in $\mathbb{R}$, $\nabla_\tau f$, $\nabla \tilde{f}$ and $\nabla_n \tilde{f}$ are vectors whereas $\partial_n \tilde{f}$ and $\partial_\tau f$ are scalars.

- Let $W \in \mathcal{C}^1\left(\Gamma, \mathbb{R}^2\right)$ be a vector field. Its tangential divergence $\mathrm{div}_\tau W : \Gamma \to \mathbb{R}$ is defined by

$$\mathrm{div}_\tau W = \mathrm{div}\tilde{W} - \nabla \tilde{W} n \cdot n,$$

  where $\tilde{W}$ is any extension of W to an open neighborhood of Γ.

Finally, Proposition 3.1 gives the formula of integration by part along a curve (adapted from [97, 149]).

Proposition 3.1. [97, 149] *Let $\Gamma \in \mathcal{G}$. Let $f \in \mathcal{C}^1(\Gamma, \mathbb{R})$ and $\theta \in \mathcal{C}^2\left(\overline{D}, \mathbb{R}^2\right)$. Then*

$$\int_\Gamma f \left(div\theta - (\nabla \theta n) \cdot n\right) ds = \int_\Gamma \left(\kappa f \theta \cdot n - \partial_\tau f \tau \cdot \theta\right) ds + f(B)\theta(B) \cdot \tau(B) - f(A)\theta(A) \cdot \tau(A). \tag{3.2.2}$$

Remark 3.3. Note that in (3.2.2), the knowledge of f out of the curve Γ is not required. In case f is defined and known on a neighborhood of the curve Γ, the tangential gradient decomposition holds:

$$\nabla_\tau = \nabla - \nabla_n = \nabla - \left(\partial_n .\right) n$$

leading to the classic integration by part formula:

$$\int_\Gamma f \left(\mathrm{div}\theta - (\nabla \theta n) \cdot n\right) + \nabla f \cdot \theta ds = \int_\Gamma \left(\kappa f \theta \cdot n + \partial_n f n \cdot \theta\right) ds + f(B)\theta(B) \cdot \tau(B) - f(A)\theta(A)\tau(A). \tag{3.2.3}$$

Derivatives in the general curves set $\mathcal{G}$

From Proposition 3.1, the shape derivatives of classic functions with respect to curves in $\mathcal{G}$ can then be provided.

Proposition 3.2. *Let $\Gamma_0 \in \mathcal{G}$. Let $g \in W^{2,1}(D, \mathbb{R})$ and $h \in \mathcal{C}^1(\Gamma_0, \mathbb{R})$. Then, the functions*

- $J_1(\Gamma) = J_2\left(\left(Id+\theta\right)\left(\Gamma_0\right)\right) = \displaystyle\int_\Gamma g(s) ds,$
- $J_2(\Gamma) = J_1\left(\left(Id+\theta\right)\left(\Gamma_0\right)\right) = \displaystyle\int_\Gamma h \circ \left(Id+\theta\right)^{-1}(s) ds,$
- $J_3(\Gamma) = J_1\left(\left(Id+\theta\right)\left(\Gamma_0\right)\right) = \displaystyle\int_\Gamma g(s) h \circ \left(Id+\theta\right)^{-1}(s) ds$

are shape differentiable at Γ_0 (i.e. $\theta = 0$) and $\forall \theta \in \mathcal{C}^2\left(\overline{D}, \mathbb{R}^2\right)$,

- $DJ_1(\Gamma_0)(\theta) = \displaystyle\int_{\Gamma_0} \left(g(s)\kappa(s) + \partial_n g(s)\right)\theta(s)\cdot n(s)ds + g(B)\theta(B)\cdot\tau(B) - g(A)\theta(A)\cdot\tau(A),$
- $DJ_2(\Gamma_0)(\theta) = \displaystyle\int_{\Gamma_0} \left(h(s)\kappa(s)\theta(s)\cdot n(s) - \partial_\tau h(s)\theta(s)\cdot\tau(s)\right)ds + h(B)\theta(B)\cdot\tau(B) - h(A)\theta(A)\cdot\tau(A),$
- $DJ_3(\Gamma_0)(\theta) = \displaystyle\int_{\Gamma_0} \left(g(s)h(s)\kappa(s) + h(s)\partial_n g(s)\right)\theta(s)\cdot n(s) - g(s)\partial_\tau h(s)\theta(s)\cdot\tau(s)ds$
$+h(B)g(B)\theta(B)\cdot\tau(B) - h(A)g(A)\theta(A)\cdot\tau(A).$

PROOF.
To begin with, Lemma 3.1 states the change of variable formula for a surfacic integral.

Lemma 3.1. [9, 97] *Let $\Gamma_0 \in \mathcal{G}$, $\theta \in \mathcal{C}^2\left(\overline{D},\mathbb{R}^2\right)$ be such that $(Id+\theta)$ is a diffeomorphism. Let $f \in L^1\left(\left(Id+\theta\right)(\Gamma_0)\right)$. Then, $f\circ(Id+\theta) \in L^1(\Gamma_0)$ and*

$$\int_{(Id+\theta)(\Gamma_0)} f ds = \int_{\Gamma_0} f\circ(Id+\theta)\left|\det\left(Id+\nabla\theta\right)\right|\left\|\left(\left(Id+\nabla\theta\right)^{-1}\right)^T n\right\|_{\mathbb{R}^2} ds. \tag{3.2.4}$$

We then determine the formula for the function J_2. Let $\theta \in \mathcal{C}^2\left(\overline{D},\mathbb{R}^2\right)$ and $\Gamma_0 \in \mathcal{G}$.

- Let $h \in \mathcal{C}^1(\Gamma_0,\mathbb{R})$. Then, Lemma 3.1 holds and
$$J_2(\Gamma) = \int_{\Gamma_0} h\left|\det\left(Id+\nabla\theta\right)\right|\left\|\left(\left(Id+\nabla\theta\right)^{-1}\right)^t n\right\|_{\mathbb{R}^2} ds.$$
Moreover, following [9, 15, 97],
$$\begin{cases} \left\|\left(\left(Id+\nabla\theta\right)^{-1}\right)^T n\right\|_{\mathbb{R}^2} = 1-\left(\nabla\theta\right)^T n\cdot n + o(\theta), \quad \lim\limits_{\theta\to 0}\dfrac{\|o(\theta)\|_{L^\infty(\Gamma)}}{\|\theta\|_{\mathcal{C}^2\left(\overline{D},\mathbb{R}^2\right)}} = 0, \\ \det\left(Id+\nabla\theta\right) = 1+\mathrm{div}\theta + o(\theta), \quad \lim\limits_{\theta\to 0}\dfrac{\|o(\theta)\|_{L^\infty(\Gamma)}}{\|\theta\|_{\mathcal{C}^2\left(\overline{D},\mathbb{R}^2\right)}} = 0. \end{cases}$$
Then,
$$J_2\left(\Gamma\right) = \int_{\Gamma_0} h ds + \int_{\Gamma_0} h\left(\mathrm{div}(\theta) - \left(\nabla\theta n\right)\cdot n\right)ds + o(\theta).$$
Proposition 3.1 finally leads to the result.
- Let $g \in W^{2,1}\left(D,\mathbb{R}\right)$. Then,
$$J_1(\Gamma) = \int_{\Gamma_0} g\circ(Id+\theta)\left|\det\left(Id+\nabla\theta\right)\right|\left\|\left(\left(Id+\nabla\theta\right)^{-1}\right)^t n\right\|_{\mathbb{R}^2} ds,$$
and $g\circ(Id+\theta) = g + \nabla g\cdot\theta + o(\theta)$, with $\lim\limits_{\theta\to 0}\dfrac{\|o(\theta)\|_{L^\infty(\Gamma)}}{\|\theta\|_{\mathcal{C}^1(\mathbb{R}^2,\mathbb{R}^2)}} = 0$. Thus,
$$J_1\left(\Gamma\right) = \int_{\Gamma_0} g ds + \int_{\Gamma_0} \nabla g\cdot\theta + g\left(\mathrm{div}(\theta) - \left(\nabla\theta n\right)\cdot n\right)ds + o(\theta).$$
Since g is defined on a neighborhood of Γ_0, (3.2.3) applies and gives the result.
- The final formula simply derives from a combination of the two first proofs.

□

Remark 3.4. In Proposition 3.2, function J_1 corresponds to the case usually found in literature. Yet, especially in Chapter 8, we use some functions defined on the curve Γ only. These functions must be transported by the vector field θ, resulting in the derivatives DJ_2 and DJ_3 that involve the tangential gradient ∂_τ.

In all these cases and under some regularity conditions, it exists v_n, $v_\tau \in \mathcal{C}^0\left(\Gamma_0,\mathbb{R}\right)$ such that the derivative is given $\forall\theta \in \mathcal{C}^2\left(\overline{D},\mathbb{R}^2\right)$ by the general formulation

$$DJ(\Gamma_0)(\theta) = \int_{\Gamma_0}\left(v_n\theta\cdot n + v_\tau\theta\cdot\tau\right)ds + v_\tau(B)\theta(B)\cdot\tau(B) + v_\tau(A)\theta(A)\cdot\tau(A). \tag{3.2.5}$$

Open bounded set

We now focus on the particular case of curves corresponding to the boundary to open bounded sets $\partial\Omega$, i.e. curves $\partial\Omega \in \mathcal{G}$ such that the endpoints A and B satisfy $A = B$ (and $\tau(A) = \tau(B)$). In this specific context [9, 97], Proposition 3.1 can be simplified.

Proposition 3.3. *Let Ω be a $\mathcal{C}^2$ an open bounded set, $f \in W^{2,1}(D, \mathbb{R})$ and $\theta \in \mathcal{C}^2\left(\overline{D}, \mathbb{R}^2\right)$. Then*

$$\int_{\partial\Omega} \left(\nabla f \cdot \theta + f\left(div\theta - (\nabla\theta n) \cdot n\right)\right) ds = \int_{\partial\Omega} \left(\partial_n f + \kappa f\right) \theta \cdot n ds. \tag{3.2.6}$$

PROOF.
The function f existing on a neighboring of $\partial\Omega$, (3.2.3) applies. Since the starting point is also the end point and because the tangent is continuous at this point, $f(B)\theta(B) \cdot \tau(B) - f(A)\theta(A) \cdot \tau(A) = 0$ leading to the result. □

Proposition 3.4 states the usual derivatives when the curve is the boundary to an open bounded set.

Proposition 3.4. [9, 15, 97] *Let $\Omega \subset \mathbb{R}^2$ be a $\mathcal{C}^2$ open bounded set. Let $f \in W^{1,1}(D, \mathbb{R})$. The function $J(\Omega) = \int_\Omega f(x)dx$ is differentiable at Ω and, $\forall\theta \in \mathcal{C}^2\left(\overline{D}, \mathbb{R}^2\right)$,*

$$DJ(\Omega)(\theta) = \int_{\partial\Omega} f(s)\theta(s) \cdot n(s) ds. \tag{3.2.7}$$

Let $g \in W^{2,1}(D, \mathbb{R})$. Then, the function $J(\Omega) = \int_{\partial\Omega} g(s)ds$ is differentiable at Ω and, $\forall\theta \in \mathcal{C}^2\left(\overline{D}, \mathbb{R}^2\right)$,

$$DJ(\Omega)(\theta) = \int_{\partial\Omega} \left(\partial_n g(s) + g(s)\kappa(s)\right) \theta(s) \cdot n(s) ds. \tag{3.2.8}$$

In the specific case of curves corresponding to boundaries to open bounded sets, and under some regularity conditions, it exists $v_n, v_\tau \in L^2(\partial\Omega, \mathbb{R})$ such that the derivative is given by the general formulation

$$DJ(\Omega)(\theta) = \int_{\partial\Omega} \left(v_n\theta \cdot n + v_\tau\theta \cdot \tau\right) ds. \tag{3.2.9}$$

Usually, Proposition 3.4 applies and $v_\tau = 0$.

3.2.3 Case of a function depending on a PDE

To evaluate the structure, shape optimization often involves solving a partial differential equation. Since it affects the optimization functions (objective and constraints), this solution must be taken into account while differentiating,. We present here a formal but easy strategy: the method of Céa [9, 15, 46].

Consider the variational formulation of a partial differential equation: find $u \in H^1(D)$ such that,

$$F(\Gamma, u, \phi) = 0, \qquad \forall\phi \in H^1(D). \tag{3.2.10}$$

Consider the optimization problem

$$\min_\Gamma f(\Gamma, u), \quad \text{such that} \quad \forall\phi \in H^1(D),\ F(\Gamma, u, \phi) = 0. \tag{3.2.11}$$

Getting the shape derivative of this problem is not straightforward because of the dependence of the variable u to the shape Γ. The method of Céa [9, 15, 17, 46] consists in introducing a Lagrangian function

$$\mathcal{L} : \begin{cases} \mathcal{G} \times H^1(D) \times H^1(D) & \to & \mathbb{R} \\ (\Gamma, v, q) & \mapsto & f(\Gamma, v) + F(\Gamma, v, q). \end{cases} \tag{3.2.12}$$

Then, for u solution of the PDE, we know that $\forall q \in H^1(D)$, $\mathcal{L}(\Gamma, u, q) = f(\Gamma, u)$. Thus

$$Df(\Gamma, u)(\theta) = D\mathcal{L}(\Gamma, u, q)(\theta) = \partial_\Gamma\mathcal{L}(\Gamma, u, q)(\theta) + \partial_v\mathcal{L}(\Gamma, u, q)\left(\partial_\theta u(\Gamma)(\theta)\right)) \tag{3.2.13}$$

Choosing p such that, $\forall\phi \in H^1(D)$, $\partial_v\mathcal{L}(\Gamma, u, p)(\phi) = 0$ spares the computation of the derivative of the solution u with respect to the shape. The function p is called the *adjoint* to the problem. To compute the derivative, we finally need to determine this adjoint and to partially differentiate the Lagrangian $\mathcal{L}$ with respect to the shape variable Γ.

Remark 3.5. The method of Céa can be understood as a saddle-point computation [9, 15].

This method is very simple and thus convenient. Yet, it must be applied with care: indeed, it is only formal and assumes the differentiability of the PDE solution with respect to the shape. Else, (3.2.13) could not be satisfied and another process should then be used, involving the differentiation of the PDE solution with respect to the curve Γ. Since the method of Céa can be used in each of the optimization problems described in this work, PDE differentiation is not detailed here. The interested reader can refer to [15] for very clear and detailed explanations.

Remark 3.6. The notion of adjoint introduced here can be related to the object used in optimal control [9, 15].

3.2.4 From the derivative to the descent direction

In both the Augmented Lagrangian and null space gradient flow methods presented in Chapter 2, the update direction is defined based on the gradients. The shape derivative of a function J must thus be transformed into a shape gradient J'. To apply Riesz representation theorem, a Hilbert structure $(H, (,)_H)$ must be chosen so that $H \subset \mathcal{C}^2\left(\overline{D}, \mathbb{R}^2\right)$, and for any $\Gamma \in \mathcal{G}$, $J'(\Gamma)$ is chosen as

$$\forall \theta \in H, \quad DJ(\Gamma)(\theta) = (J'(\Gamma), \theta)_H. \tag{3.2.14}$$

We analyze here three Hilbert structures. More information on this topic can be found in [15, 42, 67]. From Propositions 3.2 and 3.4 and under regularity assumptions, the derivative of a differentiable function J can be characterized by $v = (v_\tau, v_n) \in \mathcal{C}^0(\overline{D}, \mathbb{R})$ following

$$DJ(\Gamma)(\theta) = \int_\Gamma (v_n \theta \cdot n + v_\tau \theta \cdot \tau)\, ds + v(B)\theta(B) \cdot \tau(B) + v(A)\theta(A) \cdot \tau(A). \tag{3.2.15}$$

The first Hilbert space is $L^2(\Gamma, \mathbb{R}^2)$. This is obviously a naive idea since $L^2(\Gamma, \mathbb{R}^2) \not\subset \mathcal{C}^2\left(\overline{D}, \mathbb{R}^2\right)$. Very formally (but nevertheless often used in numerical applications), without considering any regularity issues, we could then choose $J' \in L^2(\Gamma, \mathbb{R}^2)$ such that

$$\forall \theta \in L^2(\Gamma, \mathbb{R}^2), \qquad DJ(\Gamma)(\theta) = (J', \theta)_{L^2(\Gamma, \mathbb{R}^2)} = \int_\Gamma J' \cdot \theta ds, \tag{3.2.16}$$

and thus

$$J'(s) = v_\tau \tau + v_n n, \ s \in \Gamma, \quad J'(A) = v_\tau(A)\tau(A), \quad J'(B) = v_\tau(B)\tau(B).$$

However, the Hilbert space $L^2(\Gamma, \mathbb{R}^2)$ does not provide enough regularity to ensure that the gradient J' is defined at the endpoints, nor that $DJ(\Gamma)$ is defined on $L^2(\Gamma, \mathbb{R}^2)$.
Another possibility, corresponding to the Laplace-Beltrami operator, consists in choosing the Hilbert space $H^1(\Gamma, \mathbb{R}^2)$ and thus

$$\forall \theta \in H^1(\Gamma, \mathbb{R}^2), \quad DJ(\Gamma)(\theta) = (J', \theta)_{H^1(\Gamma, \mathbb{R}^2)} = \int_\Gamma \nu_\Gamma^2 \nabla_\tau J' \cdot \nabla_\tau \theta + J' \cdot \theta ds, \tag{3.2.17}$$

with $\nu_\Gamma > 0$. This Hilbert space is still not included in $\mathcal{C}^2\left(\overline{D}, \mathbb{R}^2\right)$. However, the derivative of J can now be applied to functions $\theta \in H^1(\Gamma, \mathbb{R}^2)$ since it provides the existence of θ at the curve endpoints. Choosing a Hilbert space $H^k(\Gamma, \mathbb{R}^2)$ with $k > 1$ drastically increases the computational cost of the gradient determination [15] and is not considered in this work. The Laplace-Beltrami operator involves differentiation along the path. The gradient J' is decomposed as $J' = J'_\tau \tau + J'_n n$ with, $\forall W = W_\tau \tau + W_n n$, $W_\tau, W_n \in H^1(\Gamma, \mathbb{R})$,

$$\begin{aligned} DJ(\Gamma)(W) &= \int_\Gamma (v_n W_n + v_\tau W_\tau)\, ds + v(B)W_\tau(B) + v(A)W_\tau(A), \\ &= \int_\Gamma \left[\nu_\Gamma^2 \left(\nabla_\tau J'_\tau \cdot \nabla_\tau W_\tau + \nabla_\tau J'_n \cdot \nabla_\tau W_n\right) + J'_\tau W_\tau + J'_n W_n\right] ds \end{aligned} \tag{3.2.18}$$

Solving this equation finally amounts to solving two minimization problems and to determine $J'_\tau, J'_n \in H^1(\Gamma, \mathbb{R})$ such that

$$\begin{cases} J'_\tau = \underset{W\in H^1(\Gamma,\mathbb{R})}{\arg\min} \displaystyle\int_\Gamma \left(\frac{\nu_\Gamma^2}{2} (\partial_\tau W)^2 + \frac{1}{2} W^2 - v_\tau W \right) ds - v(A)W(A) - v(B)W(B), \\ \\ J'_n = \underset{W\in H^1(\Gamma,\mathbb{R})}{\arg\min} \displaystyle\int_\Gamma \left(\frac{\nu_\Gamma^2}{2} (\partial_\tau W)^2 + \frac{1}{2} W^2 - v_n W \right) ds. \end{cases} \tag{3.2.19}$$

When the set of curves is reduced to the boundary of open bounded sets Ω, no pointwise value is involved in the shape derivative (see (3.2.9)) and the scalar products on $L^2(\partial\Omega, \mathbb{R}^2)$ or $H^1(\partial\Omega, \mathbb{R}^2)$ can be used. A third choice consists in choosing the scalar product corresponding to $H^1(D, \mathbb{R}^2)$. This choice regularizes the problem and actually is an extension process. Indeed, this scalar product defines the gradient J' on the whole working domain D. The computation of this gradient amounts to solving, $\forall W \in H^1(D, \mathbb{R}^2)$,

$$\begin{aligned} \int_D \left(\nu_\Gamma^2 \nabla J' : \nabla W + J' \cdot W \right) dx &= DJ(\Gamma)(W) \\ &= \int_{\partial\Omega} (v_n W \cdot n + v_\tau W \cdot \tau)\, ds. \end{aligned} \tag{3.2.20}$$

Under the assumptions that each integrand involved in the optimization problem is defined on a neighborhood to the boundary $\partial\Omega$ to the open bounded set Ω (assumption used in Proposition 3.4), the component v_τ vanishes and the gradient J' is reduced to $J' = J'_n n$ with $\forall Q \in H^1(D, \mathbb{R})$,

$$\int_D \left(\nu_\Omega^2 \nabla (J'_n) \cdot \nabla Q + J'_n Q \right) dx = \int_{\partial\Omega} v_n Q ds. \tag{3.2.21}$$

These regularization equations allow to compute the shape gradient, which must then be discretized to get the numerical direction. This discretization depends on the optimization problem and will be fully detailed in Chapters 6, 7, 8, 9.

Remark 3.7. Note that the Hilbert space $H^1(D, \mathbb{R}^2)$ cannot be used for any curve $\Gamma \in \mathcal{G}$ since any function in $H^1(D, \mathbb{R}^2)$ is not necessarily defined at point A and B.

3.3 Shape optimization of a domain

The main tools used to compute shape derivatives have been reviewed in the previous section. We now focus on the specific problem of optimizing an open bounded set. To this end, we first detail domain representation, which is a key ingredient of shape optimization. We focus here on the level set strategy. Other existing representations are further introduced in Section 5.2.1 and [9, 15]. To end this chapter, an example of the compliance optimization is presented.

3.3.1 Level set method

Among the many different techniques used to optimize the boundary of an open set Ω, we choose to focus on the level set method which does not require re-meshing and easily handles topology modifications. Introduced by [156] for tracking surfaces (see [15, 17] for its application in structural optimization), it relies upon a function ψ defined on the whole working domain D such that

$$\begin{cases} \psi(x) = 0 & \iff & x \in \partial\Omega \cap D \\ \psi(x) < 0 & \iff & x \in \Omega \\ \psi(x) > 0 & \iff & x \in D \setminus \overline{\Omega}. \end{cases} \tag{3.3.1}$$

Considering the optimization process as a time evolution process, where each iteration is a time step, the level set function ψ now depends on two variables: $x \in D$ and t a pseudo-time representing the iteration count. At each iteration, the function follows the domain $\Omega(t)$ and thus, on any constant level set $\psi(t, x(t)) = C$, one has

$$\forall t, \quad d_t \psi(t, x(t)) = 0,$$

leading to

$$\partial_t \psi(t,x) + \dot{x} \cdot \nabla \psi(t,x) = 0, \quad \forall t, \forall x \in D. \tag{3.3.2}$$

On the boundary $\partial\Omega(t)$ (and in the whole domain D if the gradient chosen results from a regularization-extension scalar product), the velocity $\dot{x}(t)$ is given by the descent direction θ computed by shape differentiation:

$$\dot{x}(t) = \theta\left(t, x(t)\right), \quad \forall t, \forall x \in \partial\Omega(t). \tag{3.3.3}$$

Extending the normal to the boundary $\partial\Omega$ to the whole working domain D by $n(x) = \frac{\nabla\psi(x)}{|\nabla\psi(x)|}$ and writing $d_\Omega^n(t,x) = \theta(t,x) \cdot n(t,x)$ the update direction on the normal (and $d_\Omega = d_\Omega^n(t,x)n(t,x)$ the update direction), the update of the level set is finally given y the Hamilton Jacobi equation

$$\partial_t \psi(t,x) + d_\Omega^n(t,x)|\nabla\psi(t,x)| = 0, \quad \forall t, \forall x \in D. \tag{3.3.4}$$

In numerical applications, the processes described in [41, 181] are used. Along the iterations, the advection equation tends to flatten the level set function values thus "blurring" the results. A re-distanciation algorithm is thus applied based on the signed distance equation [9, 15, 63, 141]. In numerical applications, this function is provided by Freefem++[96].

3.3.2 Application: minimization of the volume and compliance

In order to illustrate the notions introduced in this chapter, we consider the following optimization problem.

Problem setting

Consider a design domain $D \subset \mathbb{R}^2$ and a space of admissible shapes $\mathcal{U}$, open bounded subsets of D with Lipschitz boundary. Let $\Omega \in \mathcal{U}$. Its boundary, $\partial\Omega$, with exterior unit normal vector, n, is composed of three disjoints parts (Fig. 3.2): $\partial\Omega_{\mathrm{D}}$ on which displacement is imposed (Dirichlet boundary condition), $\partial\Omega_{\mathrm{N}}$ on which any load could be applied, and $\partial\Omega_{\mathrm{F}}$ which is traction-free. Only the boundary $\partial\Omega_{\mathrm{F}}$ is optimizable and the boundary $\partial\Omega_{\mathrm{D}}$ is not reduced to the empty set ($\partial\Omega_{\mathrm{D}} \neq \emptyset$).

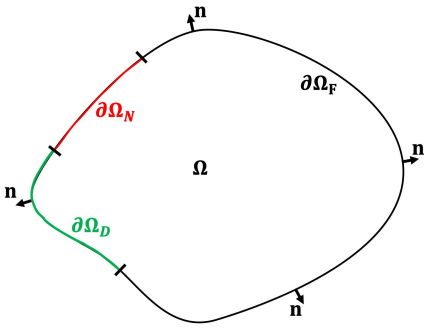


Figure 3.2: Open set Ω

The solid (and corresponding shape Ω) is comprised of an elastic material with a Hooke's tensor A relating the elastic stress and strain. Recall that for any symmetric matrix ξ, A is defined by:

$$A\xi = 2\mu_{\mathrm{e}}\xi + \lambda_{\mathrm{e}}\mathrm{Tr}(\xi)Id, \tag{3.3.5}$$

where λ_{e} and μ_{e} are the Lamé coefficients and Id is the identity tensor in two dimensions. In particular, with linearized elasticity, the elastic stress σ^e, elastic strain ϵ, and the elastic displacement u are defined such that:

$$\epsilon(u) = \frac{1}{2}\left(\nabla u + \nabla u^T\right), \qquad \sigma^e = A\epsilon(u). \tag{3.3.6}$$

The surface loading applied to the Neumann boundary, $\partial\Omega_{\mathrm{N}}$, is denoted $g \in L^2\left(\partial\Omega_{\mathrm{N}}\right)$. The elastic

displacement $u \in H^1(\Omega, \mathbb{R}^2)$ is the solution of the partial differential equation given by eq. (3.3.7):

$$\begin{cases} -\text{div}\left(A\epsilon(u)\right) = 0 & \text{in } \Omega \\ A\epsilon(u).n = g & \text{on } \partial\Omega_{\text{N}} \\ A\epsilon(u).n = 0 & \text{on } \partial\Omega_{\text{F}} \\ u = 0 & \text{on } \partial\Omega_{\text{D}}. \end{cases} \tag{3.3.7}$$

Setting $H^1_D(\Omega, \mathbb{R}^2) = \{v \in H^1(\Omega, \mathbb{R}^2), \text{ such that } v = 0 \text{ on } \partial\Omega_{\text{D}}\}$ the set of functions in $H^1(\Omega, \mathbb{R}^2)$ cancelling on $\partial\Omega_{\text{D}}$, the elastic variational problem states that, $\forall \varphi \in H^1_D(\Omega, \mathbb{R}^2)$,

$$\int_\Omega A\epsilon(u) : e(\varphi) dx - \int_{\partial\Omega_{\text{N}}} g\varphi ds = 0. \tag{3.3.8}$$

It is well known that the elastic problem, (3.3.7), is well-posed if $\partial\Omega_{\text{D}} \neq \emptyset$, and that the equation admits a unique weak solution $u \in H_1(\Omega, \mathbb{R}^2)$ [78].

The objective function is the normalized shape compliance

$$f(\Omega) = \frac{1}{C^0_{\text{ply}}} \int_\Omega A\epsilon(u) : \epsilon(u) dx = \frac{C_{\text{ply}}(\Omega)}{C^0_{\text{ply}}},$$

with $C^0_{\text{ply}} = C_{\text{ply}}(\Omega^0)$ the initial compliance. A volume constraint is added as

$$\frac{1}{V^0} \int_\Omega dx = \frac{V}{V^0} \leq \frac{V_{max}}{V^0},$$

with V_{max} fixed by the user and $V^0 = \int_{\Omega^0} dx$ the initial volume. An Augmented Lagrangian method is applied, with c_V the penalization, l_V the multiplier and the penalized Lagrangian function

$$\mathcal{L}_{\text{ALM}}(\Omega, l_V, c_V) = \frac{C_{\text{ply}}}{C^0_{\text{ply}}} + \psi_{\text{ALM}}\left(\frac{V - V_{max}}{Vol^0}, l_V, c_V\right), \tag{3.3.9}$$

with ψ_{ALM} defined by (2.4.15).

Differentiation

Using the method of Céa (see Section 3.2.3, [17, 46]), a Lagrangian function is introduced such that, $\forall (\Omega, v, q) \in \mathcal{U} \times H^1_D(D, \mathbb{R}^2) \times H^1_D(D, \mathbb{R}^2)$,

$$\mathcal{L} = \mathcal{L}_{\text{ALM}}(\Omega, l_V, c_V) + \int_\Omega A\epsilon(v) : \epsilon(q) dx - \int_{\partial\Omega_{\text{N}}} g \cdot q ds. \tag{3.3.10}$$

The differentiation of the Lagrangian with respect to v and evaluated at (Ω, u, q) leads to, $\forall \phi \in H^1_D(\Omega, \mathbb{R}^2)$,

$$\int_\Omega \left(\frac{2}{C^0_{\text{ply}}} A\epsilon(u) + A\epsilon(q)\right) : \epsilon(\phi) dx = 0.$$

Choosing $p = -\frac{2}{C^0_{\text{ply}}} u$ gives the required adjoint property. The derivative of the objective function with respect to the shape is then (following Proposition 3.4), $\forall \theta \in \mathcal{C}^2\left(\overline{D}, \mathbb{R}^2\right)$ such tha $\theta \cdot n = 0$ on $\partial\Omega_{\text{D}} \cup \partial\Omega_{\text{N}}$,

$$Df(\Omega)(\theta) = \int_{\partial\Omega_{\text{F}}} \left(\frac{1}{V^0} \psi'_{\text{ALM}}\left(\frac{V - V_{max}}{V^0}, l_V, c_V\right) - \frac{1}{C^0_{\text{ply}}} A\epsilon(u) : A\epsilon(u)\right) \theta \cdot n ds. \tag{3.3.11}$$

A $H^1(D)-$regularization is finally applied (with $\nu_\Omega > 0$) and the gradient chosen satisfies $f'(\Omega) = -Qn$ with $Q \in H^1(D, \mathbb{R})$ such that $Q = 0$ on $\partial\Omega_{\text{N}} \cup \partial\Omega_{\text{D}}$ and, $\forall W \in H^1(D, \mathbb{R})$ such that $W = 0$ on $\partial\Omega_{\text{N}} \cup \partial\Omega_{\text{D}}$,

$$\int_D \left(\nu_\Omega^2 \nabla Q \cdot \nabla W + QW\right) dx = Df(\Omega)(Wn). \tag{3.3.12}$$

Remark 3.8. The vector field chosen here could be less regular than $\mathcal{C}^2\left(\overline{D}, \mathbb{R}^2\right)$. Indeed, the curvature is not required.

Resolution algorithm

To solve the algorithm, a classic Augmented Lagrangian method (see Chapter 2, Section 2.4.4) is applied. To update the shape at each iteration k, an advection step s_Ω^k is defined as

$$s_\Omega^k = C_{s,\Omega}^k \frac{\Delta x}{\|d_\Omega^k\|_{L^\infty}}. \tag{3.3.13}$$

with Δx the characteristic element mesh size. The coefficient $C_{s,\Omega}$ is initialized to $C_{s,0} = 5$ and modified at each iteration through

$$C_{s,\Omega}^{k+1} = \max\left(\eta_{\text{acc}}^s C_{s,\Omega}^k, C_{s,0}\right) \tag{3.3.14}$$

if the iteration is accepted and

$$C_{s,\Omega}^{k+1} = \eta_{\text{ref}}^s C_{s,\Omega}^k \tag{3.3.15}$$

if rejected. The algorithm is finally recapped by Algorithm 3.1:

```
Choose the initialization Ω_0 and the corresponding level set function ψ
Solve the elastic problem and compute the objective function
Solve the adjoint problem and compute the derivative
for N iterations do
    Compute the gradient step corresponding to C_{s,k}Δx
    Update the level set function and thus the shape, Ω^{k+1} and ψ^{k+1}
    Compute the new objective function
    if f(Ω^{k+1}) ≤ tol f(Ω^k) then
        Accept the iteration
        Update the multiplier l_V^{k+1} = max(l_V^k + c_V (V^{k+1} − V_max)/V^0, 0)
        Solve the adjoint problem and compute the derivative
        C_s^{k+1} = max(η_acc^s C_s^k, C_{s,0})
    end
    else
        Iteration refused
        C_s^{k+1} = η_ref^s C_s^k
    end
end
```

Algorithm 3.1: Algorithm for classic shape optimization problems

Numerical results

We consider the classic cantilever problem. The working domain is a square with a characteristic length of size $1.4mm$, discretized into 12.800 triangular elements (Figure 3.3). The loading is applied on the middle of the right side (centered segment of size $0.28mm$). The left side constitutes the Dirichlet boundary, Γ_D. The relevant material properties and design parameters (non-dimensional) are the following: Lamé coefficients $\lambda_e = 1$ and $\mu_e = 8$, and loading $g = (0, -5)$. The regularization coefficient is set to $\nu_\Omega = 5\Delta x$ (see (3.2.21)).

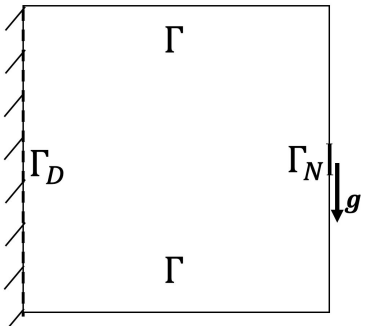


Figure 3.3: Cantilever test case

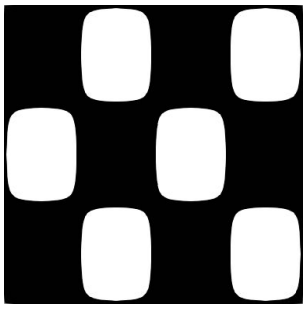
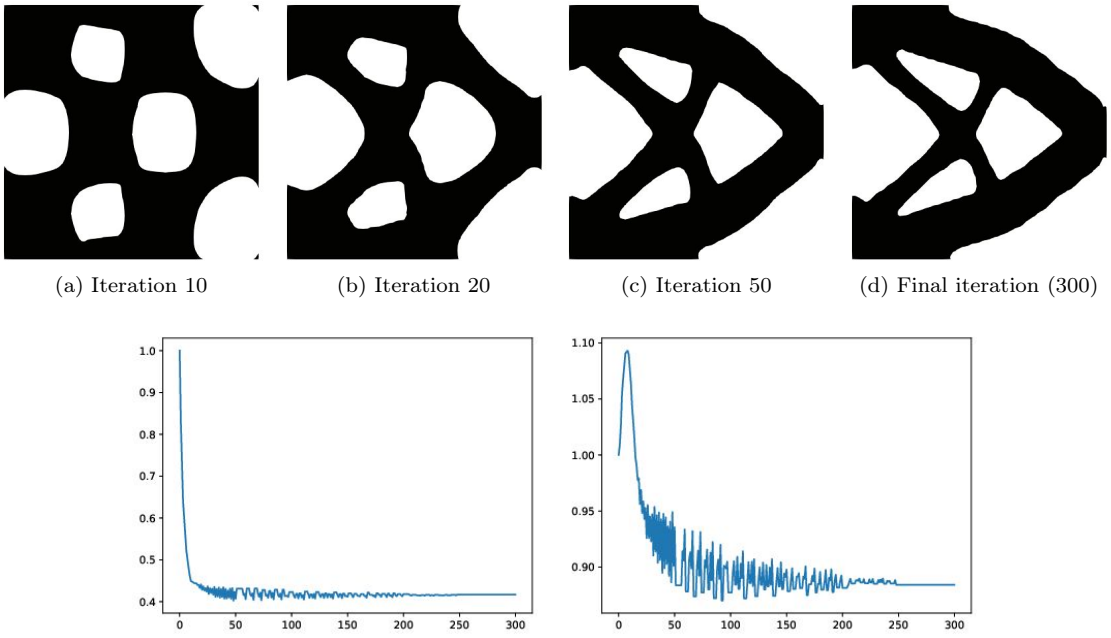


Figure 3.4: Initial shape, $V^0 = 1.15e-06$, $C_{\text{ply}}^0 = 8.13\,10^{-7}$

The optimization process is run with a tolerance *tol* (see Algorithm 3.1) initialized at 1.6 and multiplied by 0.9 every 50 iterations. The initial step coefficient is $C_{s,0} = 5$ and the update coefficients are $\eta^s_{\text{acc}} = 1.2$ and $\eta^s_{\text{ref}} = 0.6$. A first test case is run with $V_{max} = 0.8V^0$. Figure 3.5 presents the evolution of the shape as well as the compliance and volume with respect to the iterations. The evolution of the shape shows that the solid part is re-organized to strengthen some specific zones and allow for mass reduction. After 300 iterations, the optimization has converged and both the volume and the compliance have reached final values.



(a) Iteration 10 (b) Iteration 20 (c) Iteration 50 (d) Final iteration (300)

(e) Evolution of $C_{\text{ply}}/C^0_{\text{ply}}$ (f) Evolution of V/V^0

Figure 3.5: Shape, compliance and volume evolution with respect to the iterations, for $V_{max} = 0.8V^0$

From this optimization problem, three different shapes are generated and are used as complex geometry test cases for path optimization (Chapter 6, 7, 8). They correspond to optimizations with $V_{max} = 1.3V^0$, $V_{max} = 0.95V^0$ and $V_{max} = 0.85V^0$ and are respectively shown in Figures 3.6, 3.7 and 3.8.

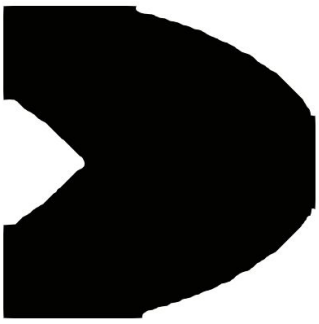


Figure 3.6: Resulting shape
$V_{max} = 1.3V^0$
$V/V^0 = 1.32203$

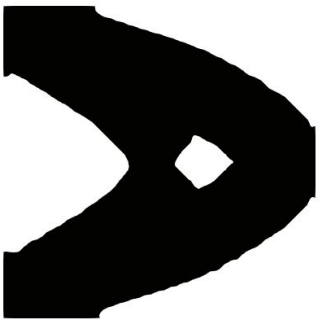


Figure 3.7: Resulting shape
$V_{max} = V^0$
$V/V^0 = 1.02973$

Figure 3.8: Resulting shape
$V_{max} = 0.95V^0$
$V/V^0 = 0.982655$

3.4 Conclusion

This chapter reviewed the shape optimization tools that will be used in the remainder of this thesis. These general tools are applied to two main problems. In Chapters 6 and 8 these tools are first used to optimize the path itself. Then, in Chapter 9, these tools are applied to domain's optimization, in order to design the part to be built.

CHAPTER 4

PROBLEM STATEMENT: MODEL AND OBJECTIVES

Contents

4.1 Introduction . . . 79
4.2 Transient model presentation: hypothesis, description and optimization problem . . 79
4.2.1 Hypothesis leading to a three dimensional model . . . 80
4.2.2 Two dimensional model . . . 81
4.2.3 Scanning path optimization problem . . . 82
4.3 Model calibration . . . 84
4.3.1 Settings of the calibration problem . . . 84
4.3.2 Adaptation to Levenberg Marquardt algorithm . . . 85
4.3.3 Target data . . . 87
4.3.4 Numerical results . . . 88
4.4 Steady model presentation . . . 91
4.4.1 Model derivation . . . 91
4.4.2 Calibration of the steady state model . . . 93
4.5 Conclusion . . . 95

4.1 Introduction

The Powder Bed Fusion (PBF) process presented in Chapter 1 is far too complex and computationally demanding to be combined to any of the optimization algorithms detailed in Chapter 2. Assumptions are thus required. Yet, scanning path optimization for PBF processes has not been studied much, letting freedom in the frame design and very few insights on the pertinence of each choice. This chapter describes the different modeling assumptions that we have used in this work and sets the framework of the different optimization problems we have considered.

In Section 4.2, we state the chosen simplifications. This leads to a first macroscopic three dimensional model focusing on thermal aspects only. To accelerate further the simulation, we have reduced this model to two dimensions and referred to in the following as the *transient model*. This Section finally sets the optimization problem. In Section 4.3, we calibrate this model to better fit the real process. Using a Levenberg-Marquardt algorithm, the coefficients involved in the temperature simulation are modified. Thanks to this process, the optimization results from the following chapters have a physical interpretation. Finally, Section 4.4 proposed a *steady state model*. To accelerate further the computations, the time dependence is neglected: the scanning path becomes a hot thread set down on the metallic powder. Even if this steady state model is obviously a toy model, it gives hints on the algorithmic issues, allowing for facilitated resolutions. Moreover, assuming that the energy source velocity tends to infinity (an assumption that could be reasonably made in EBM technology, given that the source is quite fast), it still enables physical interpretation.

4.2 Transient model presentation: hypothesis, description and optimization problem

The objective of this work is to optimize the scanning path in order to improve the quality of the final part, while keeping a relevant manufacturing time. Because optimization requires several simulations and sensitivity computations, this model must carefully balance the accuracy of physical results and the economy of the computational costs. A full simulation of the PBF process is hence prohibited and a simplified model must be chosen.

4.2.1 Hypothesis leading to a three dimensional model

Following Chapter 1, a macroscopic approach is chosen, involving powder and solid only, with material characteristics remaining independent from the temperature. This approach still provides the control of the manufacturing time and the thermo-mechanical effects, especially thermal expansion and residual stresses. Computing the latter involves plasticity models. These are hard to solve and their full resolution is most frequently avoided in optimization in aid of alternate approaches [16, 36, 51]. For example, the scanning path optimization work, developed in [51] focuses on the inherent strain method allowing for a residual stresses' quantification without involving the process's time dependency: indeed the inherent strain method developed in [124] and on which [51] is based considers a steady state model. On the contrary in this work, the time dependence is kept whereas the process efficiency is characterized through temperature considerations only. Indeed, residual stresses are mainly due to the spatial temperature gradient and to thermal expansion (see Chapter 1 and [68, 140]). The focus is thus on the thermal problem and on the control of the temperatures occurring during the building process. This choice is obviously a huge approximation. Yet, it also simplifies tremendously the physical resolution and, since very few references related to scanning path optimization exist, offers a relevant first step to the algorithm development. Expanding the results to inherent strain methods or full thermo-mechanical considerations is part of the perspectives.

In a thermal context, the phenomena involved in the building process are the source absorption by the top layer, conduction within the object, convection and radiation. Once a few layers have been built, the effects between conduction on the one hand and convection and radiation on the other hand get unbalanced [186]. In this model, for linearity reasons, convection and radiation are ignored. Yet, the model calibration process presented in Section 4.3 indirectly involve them in the resolution. Adding them to the model would require advanced computational methods and is part of the perspectives.

These first assumptions lead to a three dimensional model. Let (e_X, e_Y, e_Z) be an orthonormal basis of $\mathbb{R}^3$, and let e_Z be the building direction. The model focuses on the scanning of the top layer in the plane $\{z = 0\}$ (Figure 4.1). The basis plate is in the plane $\{z = -H\}$ (the part building has already been started). We set D the horizontal square cross section of the build chamber (an open bounded set in $\mathbb{R}^2$) and $\tilde{D}$ the three dimensional working domain such that $\tilde{D} = D \times [-H, 0]$.

At time $t_0 = 0$, the source $\tilde{q}$ is switched on for the building of the top layer. At $t = t_{\mathrm{F}} > 0$, once the path has been traveled along, the source is switched off. Note that the computation can be carried on for $t_{\mathrm{Final}} > t_{\mathrm{F}}$, with $\tilde{q}(t > t_{\mathrm{F}}) = 0$ in order to take into account the cooling down before a new layer of powder is coated. Let $\rho, c_p, \lambda \in \mathbb{R}$ be the material density, heat capacity and conductivity. Further modeling them and especially considering them time and space dependent is part of the perspectives.
Before switching on the source, the part is assumed to be at constant temperature y_{ini}, which is also the base plate temperature (a Dirichlet boundary condition is thus set on $D \times \{z = -H\}$). This approximation is related to the relaxation time allowed between each layer's building. Finally, the three dimensional working domain is assumed to be surrounded by adiabatic powder, leading to the setting of Neumann boundary conditions on $\partial D \times [-H, 0]$. These settings result in (4.2.1), where y is the temperature.

$$\begin{cases} \rho c_p \partial_t y(t,x) - \nabla \cdot (\lambda \nabla y(t,x)) = 0, & (t,x) \in (0, t_{\mathrm{F}}) \times \tilde{D}, \\ \lambda \partial_n y(t,x) = \tilde{q}(t,x), & (t,x) \in (0, t_{\mathrm{F}}) \times (D \times \{0\}), \\ \lambda \partial_n y(t,x) = 0, & (t,x) \in (0, t_{\mathrm{F}}) \times (\partial D \times [-H, 0]), \\ y(t,x) = y_{\mathrm{ini}}, & (t,x) \in (0, t_{\mathrm{F}}) \times (D \times \{-H\}), \\ y(0,x) = y_{\mathrm{ini}}, & x \in \tilde{D}, \end{cases} \tag{4.2.1}$$

In this model, the distinction between powder and solid appears only in the temperature history of the building process: the material goes from powder to solid at a point x if there exists a time t at which the temperature at x goes beyond a change of phase temperature y_ϕ. On the top layer, the solid part at each time t is characterized by $\{x \in D \times \{0\} : \text{such that } y(t,x) \geq y_\phi\}$.

Following most macroscopic models [68, 139], the energy source is modeled as a Gaussian beam, given by (4.2.2), with $\tilde{P} > 0$ the source power, A an absorption coefficient and $r > 0$, the parameter related

to the focus of the beam:

$$\tilde{q}(t,x) = \frac{A\tilde{P}}{\pi r^2} \exp\left(-\frac{(x-u(t))^2}{r^2}\right), \quad 0 \le t \le t_{\mathrm{F}}. \tag{4.2.2}$$

The source center u is defined as the solution of the trajectory ordinary differential equation:

$$\begin{cases} \dot{u}(t) = V\tau(t), \quad 0 \le t \le t_{\mathrm{F}}, \\ u(0) = \tilde{u}, \end{cases} \tag{4.2.3}$$

with a constant velocity $V > 0$ and a tangent unit vector $\tau(t)$ to the path Γ. The initial condition $\tilde{u}$ is the starting point of the path. The family of points $u(t)$ for $0 \le t \le t_{\mathrm{F}}$ defines the scanning path Γ. Usually, the solution $u(t)$ of (4.2.3) is defined in the space $C^1([0,t_{\mathrm{F}}],\mathbb{R}^2)$ or in the space of absolutely continuous functions $AC\left([0,t_{\mathrm{F}}],\mathbb{R}^2\right)$, depending on the smoothness of the path Γ [10] (recall that an absolutely continuous function is a continuous function which is the primitive of a function in $L^1(0,t_{\mathrm{F}})$). Since the right hand side of (4.2.3) does not depend on u, it is enough in the sequel to consider solutions u which belong to the simpler Hilbert space $H^1([0,t_{\mathrm{F}};\mathbb{R}^2)$ which is a subspace of the continuous functions. This induces that the source $q \in \mathcal{C}^0\left([0,t_{\mathrm{F}}],\mathcal{C}^\infty(D)\right)$. Finally choosing $y_{\mathrm{ini}} \in L^2(D,\mathbb{R})$, there exists a unique solution y to the problem (4.2.1) and $y \in \mathcal{C}^0\left([0,t_{\mathrm{F}}],L^2(\tilde{D})\right) \cap L^2\left([0,t_{\mathrm{F}}],H^1(\tilde{D})\right)$ [10].

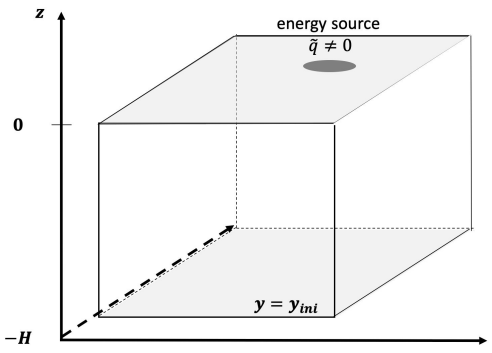


Figure 4.1: Three dimensional model

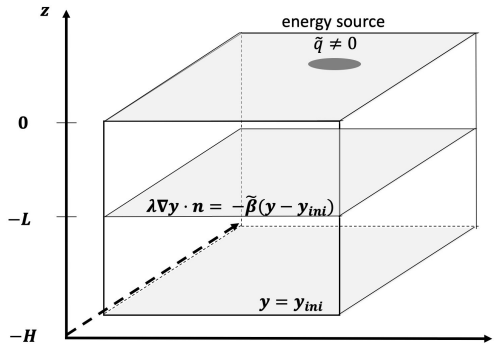


Figure 4.2: Towards a two dimensional model through Fourier boundary condition

4.2.2 Two dimensional model

In PBF processes, the path actually belongs to the two dimensional plane $\{z = 0\}$. Reducing the working domain dimensions would greatly fasten the computations and ease the optimization. Yet, such a modification involves new modeling assumptions.

The focus is now on the plane $\{z = 0\}$. The new working domain is the top layer D. The energy source, previously applied on the top layer surface, is now "volumetric" in two dimensions, whereas the working domain's boundary ∂D is chosen adiabatic (Neumann boundary condition). As for the conduction phenomenon, it is now modeled by a classic two-dimensional heat equation supplemented by an absorption term representing the vertical conduction. The introduction of this additional term is achieved in two steps.
The first one consists in truncating the working domain to focus on the last layer (Figure 4.2). We consider a new working domain $\tilde{D}_{trunc}$ with thickness L ($0 < L < H$ and $\tilde{D}_{trunc} = D \times (-L, 0)$). Recalling that the physical parameters ρ, c and λ are constant with respect to time and to the vertical axis, a Fourier boundary condition, with a transmission coefficient $\tilde{\beta}$ is set on the bottom boundary such that, $\forall (t,x) \in (0,t_{\mathrm{F}}) \times (D \times \{z = -L\})$,

$$\lambda \partial_n y = -\tilde{\beta}\left(y(t,x) - y_{\mathrm{ini}}\right).$$

This models the heat loss by conduction from the domain $\tilde{D}_{trunc}$ to the domain $\tilde{D} \setminus \tilde{D}_{trunc}$. The coefficient $\tilde{\beta}$ measures this heat transmission and must be related to the conduction at the surface and

to a characteristic thickness ΔZ. We choose $\tilde{\beta} = \frac{\lambda}{\Delta Z}$, leading to the modified heat equation (4.2.4)

$$\begin{cases} \rho c_p \partial_t y(t,x) - \nabla \cdot (\lambda \nabla y(t,x)) = 0, & (t,x) \in (0,t_{\mathrm{F}}) \times \tilde{D}_{trunc}, \\ \lambda \partial_n y(t,x) = \tilde{q}(t,x), & (t,x) \in (0,t_{\mathrm{F}}) \times (D \times \{z=0\}), \\ \lambda \partial_n y(t,x) = 0, & (t,x) \in (0,t_{\mathrm{F}}) \times (\partial D \times [-L,0]), \\ \lambda \partial_n y(t,x) = -\tilde{\beta}\,(y(t,x) - y_{\mathrm{ini}}), & (t,x) \in (0,t_{\mathrm{F}}) \times (\partial D \times \{z=-L\}), \\ y(0,x) = y_{\mathrm{ini}}, & x \in \tilde{D}_{trunc}. \end{cases} \tag{4.2.4}$$

In a second step, we average 4.2.4 in the vertical direction in order to deduce a two dimensional model. Whereas we had $x = (X,Y,Z)$ and ∇ the 3-d gradient operator in 4.2.1, we now consider $x' = (X,Y)$ and ∇' a plane gradient operator. The source term, previously applied on the top layer, is now a volumetric source of heat on the surface D. An integration along the (Oz)-axis, between $(-L)$ and 0 gives:

$$\begin{aligned} \int_{-L}^{0} \rho c_p \partial_t y - \nabla \cdot (\lambda \nabla y)\, dZ &= \rho c_p \partial_t \left(\int_{-L}^{0} y dZ \right) - \nabla' \cdot \left(\lambda \nabla' \left(\int_{-L}^{0} y dZ \right) \right) - \int_{-L}^{0} \lambda \partial_Z^2 y dZ \\ &= \rho c_p L \partial_t \tilde{y} - L \nabla' \cdot (\lambda \nabla' \tilde{y}) - \lambda \left[\partial_Z y \right]_{-L}^{0}, \\ &= \rho c_p L \partial_t \tilde{y} - L \nabla' \cdot (\lambda \nabla' \tilde{y}) - \tilde{q} + \tilde{\beta}\,(y(-L) - y_{\mathrm{ini}}). \end{aligned} \tag{4.2.5}$$

with $\tilde{y} = \dfrac{1}{L} \displaystyle\int_{-L}^{0} y dz$ the temperature averaged along the vertical axis. Finally approximating $y(-L)$ by $\tilde{y}$ and, dividing the equation by L gives (4.2.6):

$$\begin{cases} \rho c_p \partial_t y(t,x') - \nabla' \cdot (\lambda \nabla' y(t,x')) + \dfrac{\tilde{\beta}}{L}\,(y(t,x') - y_{\mathrm{ini}}) = \dfrac{\tilde{q}(t,x')}{L} & \text{in } (0,t_{\mathrm{F}}) \times D, \\ \lambda \partial_n y(t,x') = 0 & \text{on } (0,t_{\mathrm{F}}) \times \partial D, \\ y(0,x') = y_{\mathrm{ini}}(x') & \text{in } D. \end{cases} \tag{4.2.6}$$

To ease the notations, in the following we set $\beta = \frac{\tilde{\beta}}{L}$, $q = \frac{\tilde{q}}{L}$, $x = (X,Y)$ a point in two dimensions and ∇ the plane gradient operator.

In addition to the approximation of the thermal model, this lost of dimension only authorizes the control of the temperature on the top layer. Information on the other layers is not available and the remelting process must be controlled by the top temperatures only.

4.2.3 Scanning path optimization problem

The heat equation used in the following is finally given by

$$\begin{cases} \rho c_p \partial_t y(t,x) - \nabla \cdot (\lambda \nabla y(t,x)) + \beta\,(y(t,x) - y_{\mathrm{ini}}) = q(t,x) & \text{in } (0,t_{\mathrm{F}}) \times D, \\ \lambda \partial_n y(t,x) = 0 & \text{on } (0,t_{\mathrm{F}}) \times \partial D, \\ y(0,x) = y_{\mathrm{ini}}(x) & \text{in } D, \end{cases} \tag{4.2.7}$$

with $\beta = \dfrac{\lambda}{\Delta Z \cdot L}$, $P = \frac{A\tilde{P}}{L\pi r^2}$ and $\forall (t,x) \in [0,t_{\mathrm{F}}] \times D$,

$$q(x,t) = P \exp\left(-\frac{(x - u(t))^2}{r^2} \right). \tag{4.2.8}$$

Note that, as mentioned in the three dimensional model, $u \in H^1([0,t_{\mathrm{F}}], D)$, $q \in \mathcal{C}^0\left([0,t_{\mathrm{F}}], \mathcal{C}^\infty(D)\right)$ and $y \in L^2\left([0,t_{\mathrm{F}}], H^1(D)\right) \cap \mathcal{C}^0\left([0,t_{\mathrm{F}}], L^2(D)\right)$. This physical model is the basis of a path optimization problem. The working domain D is split into two different zones: D_S part of the layer that must be built during the scanning, and $D \setminus D_S$ part of the layer that must remain powder (see Figure 4.3).

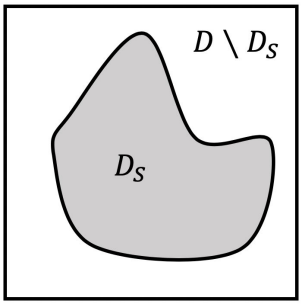


Figure 4.3: Working domain D composed of the part to built D_S and the exterior $D \setminus D_S$

- Constraint on the solid region. Assume that $D_S \subset D$ is the part of the layer that must solidify (see Figure 4.3). Then, $\forall x \in D_S$, there must be a time $t \in (0, t_{\mathrm{F}})$ such that the temperature is above the phase change (or fusion) temperature y_ϕ, namely $y(t,x) > y_\phi$.
  Therefore, the following constraint, with the notation $(.)^+ = \max(0,.)$,

$$C_\phi(t_{\mathrm{F}}, y) = \int_{D_S} \left[\left(y_\phi - \max_{t \in (0,t_{\mathrm{F}})} y(t,x) \right)^+ \right]^2 dx,$$

  has to vanish to ensure that the desired solid region is built with a given path Γ. In the following, we need to differentiate this function with respect to the final time t_{F} and to the temperature y. Since the maximum function in time is not differentiable, it is approximated by a $L^{\mathfrak{p}}$-norm in time. The effective version of the constraint is given by:

$$C_\phi(t_{\mathrm{F}}, y) = \int_{D_S} \left[(y_\phi - N_{\mathfrak{p}}(t_{\mathrm{F}}, y)(x))^+ \right]^2 dx, \quad N_{\mathfrak{p}}(t_{\mathrm{F}}, y)(x) = \left(\frac{1}{t_{\mathrm{F}}} \int_0^{t_{\mathrm{F}}} |y(t,x)]^{\mathfrak{p}} dt \right)^{1/\mathfrak{p}}. \quad (4.2.9)$$

  The integer $\mathfrak{p} \in \mathbb{N}$, $\mathfrak{p} > 1$ has a real impact on the constraint and the higher $\mathfrak{p}$ is chosen, the more accurate the norm $N_{\mathfrak{p}}$ will approximate the maximum in time. However, this will also make the variations of this same norm steepest, complicating the optimization process. In numerical applications, this coefficient is fixed to $\mathfrak{p} = 64$.

- Constraint on the maximal temperature. We limit the temperature in the whole working domain D:
  - in the region $D \setminus D_S$, the powder must not melt and the temperature has to remain under $y_{\mathrm{M,D\setminus D_S}} \leq y_\phi$. We could fix $y_{\mathrm{M,D\setminus D_S}} = y_\phi$. However, to urge the temperature to remain under the limit, we usually choose $y_{\mathrm{M,D\setminus D_S}} < y_\phi$.
  - in the region D_S, the powder must solidify. Yet, thermal stresses must be avoided. Recall that thermal stresses are typically computed like $\sigma_{th} = C\,(y - y_{\mathrm{ini}})\,Id$ with C a material parameter related to thermal expansion, y_{ini} the initial temperature and Id the identity matrix. Hence, to minimize the thermal stresses induced by the source, one can impose a maximum temperature $y_{\mathrm{M,D_S}} > y_\phi$ in the region D_S. Since this constraint comes from modeling simplifications, we fix the temperature $y_{\mathrm{M,D_S}}$ arbitrarily. The impact of this temperature on the optimized path in the steady state context is assessed in Section 6.5.3 (Chapter 6).

  These limited temperature conditions must be satisfied at any time and any point in the domain. This leads to two distinct constraints

$$\begin{cases} C_{\mathrm{M,D_S}}(t_{\mathrm{F}}, y) & = \dfrac{1}{t_{\mathrm{F}}} \displaystyle\int_D \int_0^{t_{\mathrm{F}}} \left[(y(t,x) - y_{\mathrm{M,D_S}})^+ \right]^2 \mathbb{1}_{D_S}(x) dt dx \\ & = \dfrac{1}{t_{\mathrm{F}}} \displaystyle\int_{D_S} \int_0^{t_{\mathrm{F}}} \left[(y(t,x) - y_{\mathrm{M,D_S}})^+ \right]^2 dt dx, \\ C_{\mathrm{M,D\setminus D_S}}(t_{\mathrm{F}}, y) & = \dfrac{1}{t_{\mathrm{F}}} \displaystyle\int_D \int_0^{t_{\mathrm{F}}} \left[(y(t,x) - y_{\mathrm{M,D\setminus D_S}})^+ \right]^2 \mathbb{1}_{D\setminus D_S}(x) dt dx \\ & = \dfrac{1}{t_{\mathrm{F}}} \displaystyle\int_{D\setminus D_S} \int_0^{t_{\mathrm{F}}} \left[(y(t,x) - y_{\mathrm{M,D\setminus D_S}})^+ \right]^2 dt dx. \end{cases} \quad (4.2.10)$$

Both constraints can be gathered in a unique one by defining the space dependent maximum temperature y_{M} as

$$y_{\mathrm{M}}(x) = \begin{cases} y_{\mathrm{M,D_S}} & x \in D_S, \\ y_{\mathrm{M,D\setminus D_S}} & x \in D \setminus D_S. \end{cases} \tag{4.2.11}$$

The resulting constraint is then:

$$C_{\mathrm{M}}(t_{\mathrm{F}}, y) = \frac{1}{t_{\mathrm{F}}} \int_D \int_0^{t_{\mathrm{F}}} \left[(y(t,x) - y_{\mathrm{M}}(x))^+ \right]^2 dxdt. \tag{4.2.12}$$

- The objective consists in minimizing the execution time t_{F} or equivalently the path length L_{F} (since the velocity V is constant):

$$t_{\mathrm{F}} = \frac{L_{\mathrm{F}}}{V}\,.$$

Recall that in the real process, the velocity is not constant and especially curvature dependent. This model should be supplemented by a kinematic model or constraints on the curvature. These features are part of the perspectives.

Remark 4.1. The function $z^+ = \max(0, z)$ is not differentiable at 0. However, its square $z \to (z^+)^2$ is indeed differentiable at 0 and is thus amenable to gradient-based optimization. The square function has been chosen here but it could have been any smooth increasing function f, positive on $\mathbb{R}^+$ such that $f(0) = f'(0) = 0$. Other choices are possible for replacing a pointwise constraint with an integral one.

Remark 4.2. In this model, the phase change is instantaneous. This is inaccurate since, in reality, the temperature must stand above the change of phase temperature for a small amount of time δt_ϕ, which would need to be modeled. This issue could be addressed in a simpler way by stating an effective change of state temperature $\tilde{y_\phi}$ such that $\tilde{y_\phi} > y_\phi$. Thus, since the temperature varies continuously in time, it would remain above y_ϕ a bit longer.

The following optimization problem finally holds:

$$\min_{\Gamma} J(\Gamma) = t_{\mathrm{F}} \quad \text{such that} \quad \begin{cases} C_\phi = C_{\mathrm{M}} = 0, \\ y \text{ solution of (4.2.7) depending on } \Gamma \text{ through the function } u(t) \\ \text{involved in the source } q \text{ by (4.2.8)}. \end{cases} \tag{4.2.13}$$

4.3 Model calibration

The two dimensional model on which the optimization problem is built is based on several approximations and especially a dimension reduction. In order to physically interpret the numerical results, this resulting model is calibrated to better fit the real process.

Calibration is widely used in applied mathematics with the resolution of an inverse problem on the model parameters to fit reference data. To do so, specific algorithms have been developed. Usually modeling the distance to the data through least square criteria, most approaches involve Gauss Newton and Levenberg-Marquardt methods presented in Chapter 2 (Section 2.5.2).

4.3.1 Settings of the calibration problem

The model calibration focuses in this work on five specific parameters: the material coefficients ρc_p (since the density and heat capacity always appear together, they are considered as a unique parameter), the conductivity λ, the source radius r as well as the parameters $\tilde{\beta}$ and L, characterizing the two dimensional approximation. This arbitrary choice comes from the will to keep the source parameters constant, but for the radius. Indeed, this latter being deeply related to the mesh precision, it is essential to adapt it to the working domain discretization chosen in this work (see Remark 4.4).

In order to ease the calibration process, the parameters to fit are scaled so that

$$\rho c_p = \rho c_{p,0} v_{\rho c_p}, \quad \lambda = \lambda_0 v_\lambda, \quad r = r_0 v_r, \quad \tilde{\beta} = \tilde{\beta}_0 v_{\tilde{\beta}}, \quad L = L_0 v_L. \tag{4.3.1}$$

The quantities $\rho c_{p,0}$, λ_0, r_0, $\tilde{\beta}_0$, L_0 are fixed and the optimization parameters are gathered in an optimization vector $v \in \mathbb{R}^5$ so that $v = (v_0, v_1, v_2, v_3, v_4) = \left(v_{\rho c_p}, v_\lambda, v_r, v_{\tilde{\beta}}, v_L\right)$. The heat equation (4.3.2) finally holds.

$$\begin{cases} v_{\rho c_p} \rho c_{p,0} \partial_t y(x,t) - \nabla \left(v_\lambda \lambda_0 \nabla y(t,x)\right) + \dfrac{v_{\tilde{\beta}} \tilde{\beta}_0}{v_L L_0} \left(y(t,x) - y_{\text{ini}}(x)\right) & \\ \qquad = \frac{A\tilde{P}}{\pi (v_r r_0)^2 v_L L_0} \exp\left(-\frac{(x-u(t))^2}{(v_r r_0)^2}\right) \mathbb{1}_{t<t_{\text{F}}}(t) & (t,x) \in [0, t_{\text{Final}}] \times D \\ v_\lambda \lambda_0 \partial_n y(t,x) = 0 & (t,x) \in [0, t_{\text{Final}}] \times \partial D \\ y(0,x) = y_{\text{ini}}(x) & x \in D. \end{cases} \tag{4.3.2}$$

Note that this equation is simulated until t_{Final} whereas the source is switched off for $t > t_{\text{F}}$: the relaxation process is taken into account. In order to fit the target temperature $\widehat{y}$, a least square function f^{CAL} is built as

$$f^{\text{CAL}}(v) = \int_0^{t_{\text{Final}}} \int_D \left(\frac{y - \widehat{y}}{\widehat{y}}\right)^2 (t,x) dx dt. \tag{4.3.3}$$

An Euler implicit resolution scheme is used for the resolution. The time is discretized into N time steps of constant length Δt ($t = (0, ..., t_N)$, $t_N = t_{\text{Final}}$). The path Γ on which travels the energy source center u is approximated by a broken line with constant length segments (more details on the path discretization are provided in Chapter 5), with the segment size $\Delta u = V \Delta t$. The source is thus applied on a sequence of points $(u_i)_{i \in [\![1, N_u]\!]}$ belonging to the path Γ and corresponding to the times $(t_1, ..., t_{N_u})$ (the final time corresponds to $t_{\text{F}} = \Delta t N_u$). For $i \in [\![N_u + 1, N]\!]$, the source is switched off. This leads to a sequence of temperatures $(y_i)_{i \in [\![0,N]\!]}$ so that $y_0 = y_{\text{ini}}$ and, $\forall i \in [\![0, N-1]\!]$,

$$\begin{cases} v_{\rho c_p} \rho c_{p,0} \dfrac{y_{i+1} - y_i}{\Delta t} - \nabla \left(v_\lambda \lambda_0 \nabla y_{i+1}\right) + \dfrac{v_{\tilde{\beta}} \tilde{\beta}_0}{v_L L_0} \left(y_{i+1} - y_{\text{ini}}\right) & \\ \qquad = \frac{A\tilde{P}}{\pi (v_r r_0)^2 v_L L_0} \exp\left(-\frac{(x-u_{i+1})^2}{(v_r r_0)^2}\right) \mathbb{1}_{i+1 \leq N_u} & x \in D \\ v_\lambda \lambda_0 \partial_n y_{i+1} = 0 & x \in \partial D \\ y_0 = y_{\text{ini}} & x \in D. \end{cases} \tag{4.3.4}$$

To the sequence of temperatures $(y_i)_{i \in [\![0,N]\!]}$ corresponds a sequence of target temperatures $(\widehat{y}_i)_{i \in [\![0,N]\!]}$ and the discretized least square function

$$f^{\text{CAL}}(v) = \sum_{i=0}^{N} \Delta t \int_D \left(\frac{y_i - \widehat{y}_i}{\widehat{y}_i}\right)^2 (x) dx. \tag{4.3.5}$$

Note that, if more than one path is used for the calibration, the sum of the least square functions corresponding to each simulation gives the final objective.

Remark 4.3. We recall that the absorption coefficient holds as $\tilde{\beta} = \frac{\lambda}{\Delta Z}$ and optimizing this coefficient actually amounts to optimizing the characteristic length ΔZ.

4.3.2 Adaptation to Levenberg Marquardt algorithm

In order to minimize the least square function, a Levenberg-Marquardt strategy, presented in Chapter 2 (Section 2.5.2) is used. To compute the Jacobian and Hessian approximation, we first set $F^{\text{CAL}}(v) = \left(F_i^{\text{CAL}}(v)\right)_{i \in [\![0,N]\!]}$ with $\forall x \in D$,

$$F_i^{\text{CAL}}(v)(x) = \sqrt{\Delta t} \left(\frac{y_i - \widehat{y}_i}{\widehat{y}_i}\right)(x). \tag{4.3.6}$$

Then, following (4.3.5), the least square function f^{CAL} corresponds to the $L^2(D)^{N+1}$ norm of this vector:

$$f^{\text{CAL}}(v) = \|F^{\text{CAL}}(v)\|^2_{H^1(D)^{N+1}} = \sum_{i=0}^{N+1} \int_D F_i^{\text{CAL}}(v)^2(x) dx.$$

The Jacobien matrix $\mathcal{J}^{\mathrm{CAL}}$ is defined by, $\forall i \in [\![0, N]\!]$ (time steps), $\forall j \in [\![0, 4]\!]$ (number of calibration parameters), $\forall x \in D$

$$J[F^{\mathrm{CAL}}]_{i,j}(v)(x) = \frac{\sqrt{\Delta t}}{\widehat{y}_i(v)(x)} \partial_{v_j} y_i(v)(x). \tag{4.3.7}$$

The gradient of f^{CAL} is then given, $\forall x \in D$, by $\nabla f^{\mathrm{CAL}}(v)(x) = 2\left(J[F^{\mathrm{CAL}}](v)^T \cdot F^{\mathrm{CAL}}(v)\right)$, with, $\forall j \in [\![0, 4]\!]$,

$$\left(J[F^{\mathrm{CAL}}](v)^T \cdot F^{\mathrm{CAL}}(v)\right)_j = \sum_{k=0}^{N} \Delta t \int_D \frac{\partial_{v_j} y_k(v)\,(y_k(v) - \widehat{y}_k(v))}{\widehat{y}_k(v)^2}(x) dx. \tag{4.3.8}$$

Finally, the simplified Hessian $H[F^{\mathrm{CAL}}] = J[F^{\mathrm{CAL}}](v)^T \cdot J[F^{\mathrm{CAL}}](v)$ is defined, $\forall i, j \in [\![0, 4]\!]$, by

$$H[F^{\mathrm{CAL}}]_{i,j} = \sum_{k=0}^{N} \Delta t \int_D \frac{\partial_{v_j} y_k(v) \partial_{v_i} y_k(v)}{\widehat{y}_k(v)^2}(x) dx. \tag{4.3.9}$$

To compute the descent direction, the remaining task is to get the sensitivity of the temperature with respect to the calibration vector v. A simple differentiation of the partial differential equation (4.3.4) gives,

- with respect to $v_{\rho c_p}$:

$$\begin{cases} v_{\rho c_p} \rho c_{p,0} \partial_t \partial_{v_{\rho c_p}} y - \nabla\left(v_\lambda \lambda_0 \nabla \partial_{v_{\rho c_p}} y\right) + \frac{v_{\tilde\beta} \tilde\beta_0}{v_L L_0} \partial_{v_{\rho c_p}} y = -\rho c_{p,0} \partial_t y & \text{in } (0, t_{\mathrm{Final}}) \times D \\ v_\lambda \lambda_0 \partial_n \partial_{v_{\rho c_p}} y = 0 & \text{on } (0, t_{\mathrm{Final}}) \times \partial D \\ \partial_{v_{\rho c_p}} y(t=0) = 0 & \text{in } D. \end{cases}$$

- with respect to v_λ:

$$\begin{cases} v_{\rho c_p} \rho c_{p,0} \partial_t \partial_{v_\lambda} y - \nabla\left(v_\lambda \lambda_0 \nabla \partial_{v_\lambda} y\right) + \dfrac{v_{\tilde\beta} \tilde\beta_0}{v_L L_0} \partial_{v_\lambda} y = \nabla\left(\lambda_0 \nabla y\right) & \text{in } (0, t_{\mathrm{Final}}) \times D \\ v_\lambda \lambda_0 \partial_n \partial_{v_\lambda} y + \lambda_0 \partial_n y = 0 & \text{on } (0, t_{\mathrm{Final}}) \times \partial D \\ \partial_{v_\lambda} y(t=0) = 0 & \text{in } D. \end{cases}$$

- with respect to v_r:

$$\begin{cases} v_{\rho c_p} \rho c_{p,0} \partial_t \partial_{v_r} y - \nabla\left(v_\lambda \lambda_0 \nabla \partial_{v_r} y\right) + \dfrac{v_{\tilde\beta} \tilde\beta_0}{v_L L_0} \partial_{v_r} y \\ \qquad = \begin{cases} \frac{2A\tilde P}{\pi v_L L_0 v_r^2 r_0^2} \exp\left(-\frac{(X-u)^2}{v_r^2 r_0^2}\right)\left[\frac{(X-u)^2}{v_r^3 r_0^2} - \frac{1}{v_r}\right] & t \le t_{\mathrm{F}} \\ 0 & t > t_{\mathrm{F}} \end{cases} & \text{in } (0, t_{\mathrm{Final}}) \times D \\ v_\lambda \lambda_0 \partial_n \partial_{v_r} y = 0 & \text{on } (0, t_{\mathrm{Final}}) \times \partial D \\ \partial_{v_r} y(t=0) = 0 & \text{in } D. \end{cases}$$

- with respect to $v_{\tilde\beta}$:

$$\begin{cases} v_{\rho c_p} \rho c_{p,0} \partial_t \partial_{v_{\tilde\beta}} y - \nabla\left(v_\lambda \lambda_0 \nabla \partial_{v_{\tilde\beta}} y\right) + \dfrac{v_{\tilde\beta} \tilde\beta_0}{v_L L_0} \partial_{v_{\tilde\beta}} y = \dfrac{\tilde\beta_0}{v_L L_0}\left(y_{\mathrm{ini}} - y\right) & \text{in } (0, t_{\mathrm{Final}}) \times D \\ v_\lambda \lambda_0 \partial_n \partial_{v_{\tilde\beta}} y = 0 & \text{on } (0, t_{\mathrm{Final}}) \times \partial D \\ \partial_{v_{\tilde\beta}} y(t=0) = 0 & \text{in } D. \end{cases}$$

- with respect to v_L:

$$\begin{cases} v_{\rho c_p} \rho c_{p,0} \partial_t \partial_{v_L} y - \nabla\left(v_\lambda \lambda_0 \nabla \partial_{v_L} y\right) + \dfrac{v_{\tilde\beta} \tilde\beta_0}{v_L L_0} \partial_{v_L} y \\ \qquad = \frac{v_{\tilde\beta} \tilde\beta_0}{v_L^2 L_0}\left(y - y_{\mathrm{ini}}\right) - \begin{cases} \frac{A\tilde P}{\pi v_L^2 L_0 v_r^2 r_0^2} \exp\left(-\frac{(X-u)^2}{v_r^2 r_0^2}\right) & t \le t_{\mathrm{F}}, \\ 0 & t > t_{\mathrm{F}}, \end{cases} & \text{in } (0, t_{\mathrm{Final}}) \times D \\ v_\lambda \lambda_0 \partial_n \partial_{v_L} y = 0 & \text{on } (0, t_{\mathrm{Final}}) \times \partial D \\ \partial_{v_L} y(t=0) = 0 & \text{in } D. \end{cases}$$

They are then discretized following the same Euler implicit scheme than for the heat equation. Finally, since the coefficients represent physical quantities, each element of the vector v must remain positive. Thus, each iteration is ended by a projection $\mathbb{P}_+$ of the vector v on the positive values. The projected Levenberg-Maquardt algorithm (see Algorithm 4.1) has been chosen accordingly to [107]:

$$\mu_{CAL} = 1, \quad \gamma_{CAL} = 0.99995, \quad \sigma_{CAL} = 10^{-4}, \quad \alpha_{CAL} = 0.9, \quad \epsilon_{CAL} = 10^{-5}, \quad \epsilon_{CAL,2} = 10^{-12}. \tag{4.3.10}$$

Choose an initialization v^0 as well as $\mu_{CAL} > 0$, $\gamma_{CAL}, \sigma_{CAL}, \alpha_{CAL} \in (0,1)$, $\epsilon_{CAL}, \epsilon_{CAL,2} > 0$ and $k := 0$
if $\sqrt{f^{\mathrm{CAL}}(v^k)} = \|F^{\mathrm{CAL}}(v^k)\| \leq \epsilon_{CAL}$ **then**
  STOP
end
else
  Compute the direction d_v^k such that
  $\left(H[F^{\mathrm{CAL}}](v^k) + \mu_{CAL} f^{\mathrm{CAL}}(v^k) I_d\right) d_v^k = -\left(J[F^{\mathrm{CAL}}](v^k)^T \cdot F^{\mathrm{CAL}}(v^k)\right)$
  if $\sqrt{f^{\mathrm{CAL}}\left(\mathbb{P}_+\left(v^k + d_v^k\right)\right)} \leq \gamma_{CAL}\|F^{\mathrm{CAL}}(v^k)\| = \gamma_{CAL}\sqrt{f^{\mathrm{CAL}}(v^k)}$ **then**
    Levenberg-Marquardt iteration accepted
    $v^{k+1} = \mathbb{P}_+\left(v^k + d_v^k\right)$
  end
  else
    Compute a stepsize $t_k = \max\{\alpha_{CAL}^l, l = 0, 1, 2, ..\}$ such that
    $$f^{\mathrm{CAL}}\left(\mathbb{P}_+\left(v^k - t_k \nabla f^{\mathrm{CAL}}(v_k)\right)\right) \leq f^{\mathrm{CAL}}(v^k) + \sigma_{CAL} \nabla f^{\mathrm{CAL}}(v^k)^T \cdot \left(\mathbb{P}_+\left(v^k - t_k \nabla f^{\mathrm{CAL}}(v^k)\right) - v^k\right)$$
    if $t_k \leq \epsilon_{CAL,2}$ **then**
      STOP
    end
    else
      $v^{k+1} = \mathbb{P}_+\left(v^k - t_k \nabla f^{\mathrm{CAL}}(v^k)\right)$
    end
  end
end

Algorithm 4.1: Projected Levenberg Marquardt.

4.3.3 Target data

This calibration work has been initiated thanks to the data provided by Giovanni MAGNO, Centre de Nanosciences et de Nanotechnologies, Université Paris Saclay, laboratory part of the SOFIA project [2]. The final target data have been provided by the code developed by Kamel ETTAIEB, LURPA. This codes uses a convolution method to simulate a three dimensional conduction heat equation. This model had itself been calibrated using ESI codes and is fully presented in [77].

These data are given for a working domain $D_{data} = [0, 1.0\,10^{-3}m] \times [0, 1.0\,10^{-3}m]$ meshed with 10000 quadrangle elements. Four simulation sets are used: two paths combined to two different initial temperatures ($293K$ and $773K$). The source speed is fixed to $V = 1ms^{-1}$ and the computational time step is $\Delta t = 1.0\,10^{-5}s$. The paths are discretized with equidistant points (with a distance $\Delta u = V\Delta t$) and correspond to (Figures 4.4 and 4.5):

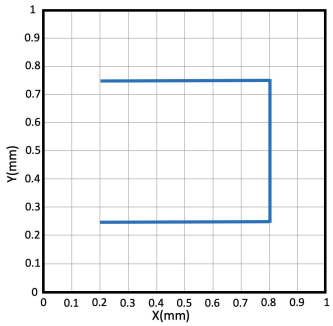
- 2 lines zigzag, starting at $\left(0.2\,10^{-3}m, 0.25\,10^{-3}m\right)$, with horizontal length $0.6\,10^{-3}m$ and vertical length $0.5\,10^{-3}m$.
- 3 lines zigzag, starting at $\left(0.2\,10^{-3}m, 0.17\,10^{-3}m\right)$, with horizontal length $0.6\,10^{-3}m$ and vertical length $0.33\,10^{-3}m$.

The source power is also a Gaussian source (see (4.2.2)). However, the absorption A is coefficiented by 0.4 coming from the data model calibration. This absorption being fixed to $A = 0.3$, we will consider in

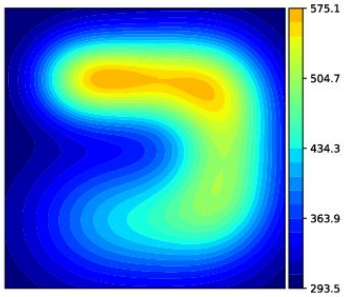
the following $A = 0.3 * 0.4 = 0.12$. The source power is $\tilde{P} = 300W$ and its radius $r = 5.0\,10^{-5}m$. As for the physical coefficients used, they correspond to the Ti6Al4V alloy with $\rho c_p = 4420 * 800 m^{-3} J K^{-1}$ and $\lambda = 15 W m^{-1} K^{-1}$. Note that these values correspond to the powder's.

The data provided gives more information than the N_u target temperatures corresponding to the source travel along the scanning path. Indeed, 100 additional temperatures are given, corresponding to the evolution of the temperature straight after the source is switched off (in each case, $N = N_u + 100$).

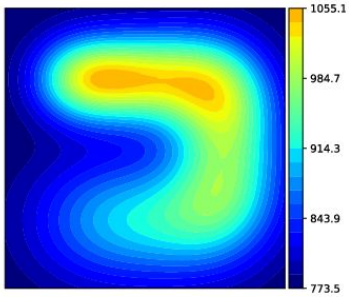
Further introduced in the next chapters, the working domain used in this work is different from the one used to simulate the data. We consider $D = [-0.7, 0.7] \times [-0.7, 0.7]$ given in mm meshed with 12800 isosceles right-angled triangular elements (characteristic mesh size of $02.47\,10^{-5}m$). The data being provided for $D_{\text{data}} = [0, 1.0] \times [0, 1.0]$ given in mm, the calibration mesh is composed of 6498 isosceles right-angled triangular elements (characteristic mesh size of $2.48\,10^{-5}m$, close to the mesh size that will be used in the remain of this work). The target temperatures are interpolated on this new mesh thanks to the algorithms developed by FreeFem++ [96]. The temperature maps after N time steps on the new working domain are given by Figures 4.4 and 4.5.



(a) Two lines zigzag

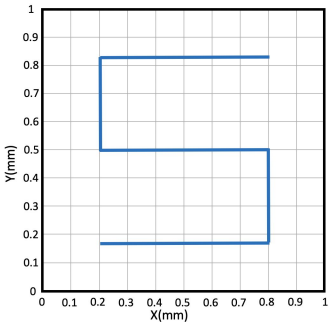


(b) Temperature 100 time steps after switching off the source, $y_{\text{ini}} = 293K$

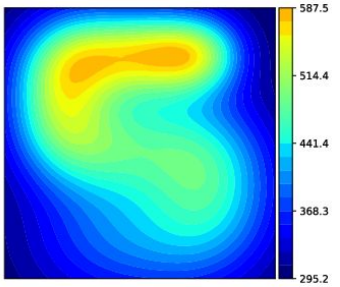


(c) Temperature 100 time steps after switching off the source, $y_{\text{ini}} = 773K$

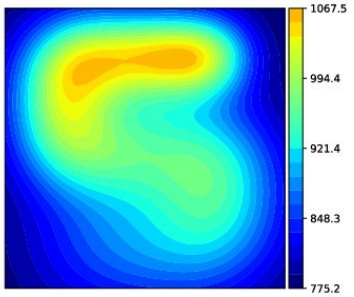
Figure 4.4: Two lines zigzag path and corresponding temperature 100 iterations after switching off the source, with $y_{\text{ini}} = 293K$ (center) and $y_{\text{ini}} = 773K$(right)



(a) Three lines zigzag



(b) Temperature 100 time steps after switching off the source, $y_{\text{ini}} = 293K$



(c) Temperature 100 time steps after switching off the source, $y_{\text{ini}} = 773K$

Figure 4.5: Three lines zigzag path and corresponding temperature 100 iterations after switching off the source, with $y_{\text{ini}} = 293K$ (center) and $y_{\text{ini}} = 773K$(right)

4.3.4 Numerical results

Algorithm 4.1 is run with

$$\begin{aligned} \rho c_{p,0} = 4420 * 800 m^{-3} J K^{-1}, \quad \lambda_0 = 15 W m^{-1} K^{-1}, \quad r_0 = 5.0\,10^{-5} m \\ \tilde{\beta}_0 = \frac{\lambda_0}{\Delta Z} = \frac{15}{0.5\,10^{-3}} W m^{-2} K^{-1}, \quad L_0 = 0.1\,10^{-3} m. \end{aligned} \tag{4.3.11}$$

If the values $\rho c_{p,0}$, λ_0 and r_0 are chosen from the data (values corresponding to the titanium, see the Nomenclature), the coefficients βLZ and L_0 come from successive guesses. As for the initial optimization vector, v, each of its elements is initialized to 1 ($v^0 = (1, 1, 1, 1, 1)$).

Remark 4.4. The source radius is strongly related to the mesh accuracy. Indeed, to be well captured, the mesh must be thin enough around the source. In our case, the mesh is thinner than the one used to provide the data temperatures. Thus, there should not be any source capture issue. The radius is nevertheless kept as an optimization parameter.

Results

The code is run using a combination of FreeFem++ (FreeFem++ 3.56) for the finite element computations and Python (Python 3.6.7). On a MacBook laptop equipped with 2,3 GHz Intel Core i5 and a RAM of 16GB, a mean Levenberg-Marquardt iteration takes 30s whereas a gradient type one takes 70s. The code is stopped after 60 iterations. Figures 4.6 and 4.7 give the temperature 100 iterations after the source's switching off for the initial and final parameters for both the 2 lines zigzag and 3 lines zigzag paths with $y_{\text{ini}} = 293K$. The calibration process mainly increases the absorption. This is very clear looking the evolution of $v_{\tilde{\beta}}$ with respect to the iterations. In the temperature maps, this comes out by the decrease of the global temperature during the calibration. The results corresponding to $y_{\text{ini}} = 793K$ show the same evolution tendency. Figure 4.8 shows the evolution of the optimization variable elements $v_{\rho c_p}, v_\lambda, v_r, v_{\tilde{\beta}}, v_L$ and the evolution of the least square function f^{CAL}.
The final coefficients are:

$$v_{\rho c_p} = 0.865692, \quad v_\lambda = 0.838116, \quad v_r = 1.062719, \quad v_{\tilde{\beta}} = 4.293009, \quad v_L = 0.675627.$$

leading to the effective coefficients:

$$\rho c_p = 3061086.91 m^{-3} J K^{-1}, \quad \lambda = 12.57 W m^{-1} K^{-1}, \quad r = 5.31\,10^{-5} m$$
$$\tilde{\beta} = \frac{\lambda}{\Delta Z} = \frac{12.57}{9.76\,10^{-5}} W m^{-2} K^{-1}, \quad L = 6.76\,10^{-5} m.$$

To quantify these calibration results in terms of mean temperature difference to the data, we introduce $\Delta(t,x) = \left(\frac{y(t,x) - \widehat{y}(x)}{\widehat{y}(x)}\right)$ the pointwise difference as wall as its "mean value" over space and time:

$$\overline{\Delta} = \frac{1}{N|D|} \sum_{i=0}^{N} \int_D |\Delta(t_i, x)|.$$

This value went from 0.0498 to 0.0217 during this calibration process. The results are here quoted several significant figures. This choice only consists to avoid accuracy losses from the calibration process to further numerical tests, and is in no way dictated by the model's accuracy which obviously remains low.

Artificial parameters $\tilde{\beta}$ and L optimization

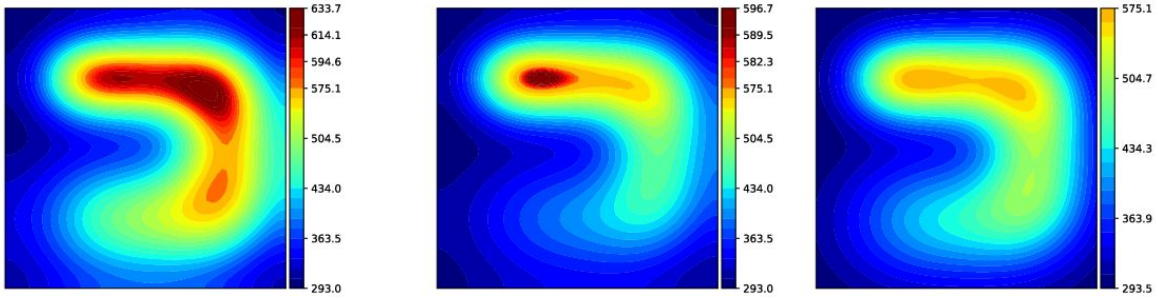
To keep the parameters ρc_p, λ, r unchanged, the calibration can be run to optimize only the artificial coefficients $\tilde{\beta}$ and L. Using the previous initialization and running 60 iterations, the final values are

$$v_{\tilde{\beta}} = 4.286761, \quad v_L = 0.585071,$$

leading to the effective parameters

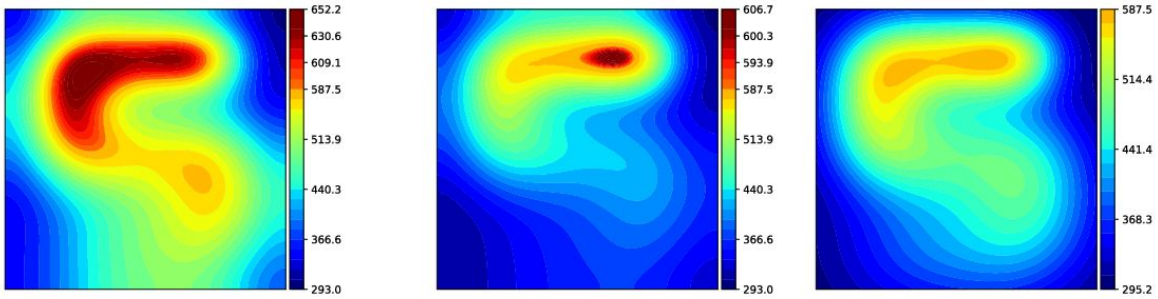
$$\tilde{\beta} = \frac{\lambda}{\Delta Z} = \frac{15}{1.16638\,10^{-4}} W m^{-2} \cdot K^{-1}, \quad L = 5.85071\,10^{-5} m,$$

and to a final mean temperature difference $\overline{\Delta} = 0.0220$. The temperature obtained 100 iterations after the source's switching off for the final iteration in the different path cases with $y_{\text{ini}} = 293K$ is shown by Figure 4.9 whereas the evolution of the parameters $v_{\tilde{\beta}}$ and v_L, of the least square function f^{CAL} are given by Figure 4.10.
It is worth noticing that the final temperatures are very close to the previous results and the final least square function value is almost as small as the previous one. Thus, without any modification of the material nor the process parameters, two lengths L and ΔZ (ΔZ being related to the absorption parameter $\tilde{\beta}$) can be found making the two dimensional model a correct approximation of the three dimensional one.



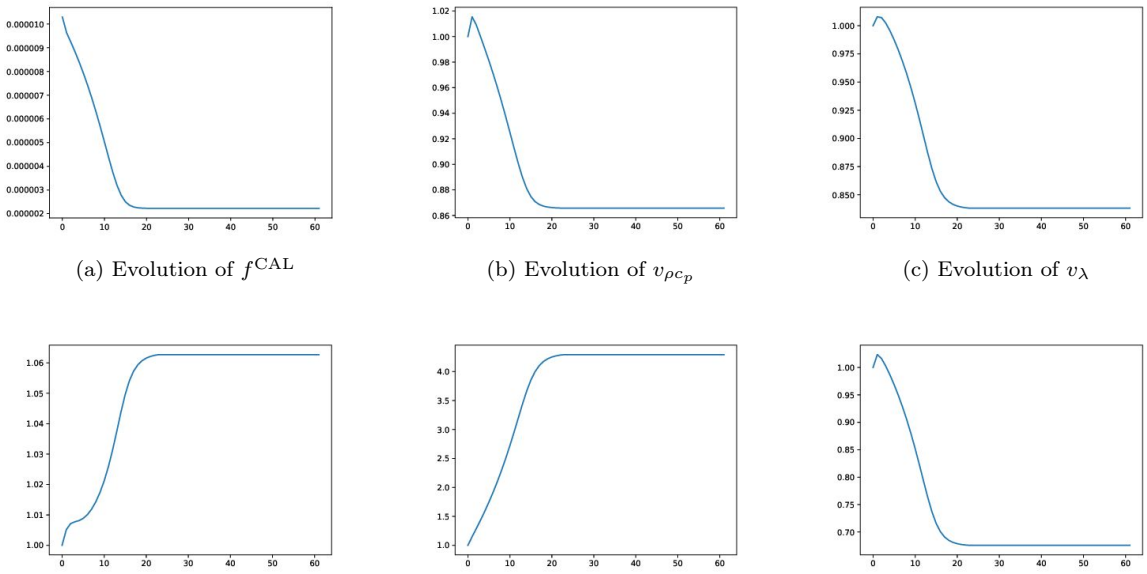
(a) Computed with the initial vector v^0 (b) Computed with the final vector v (c) Target

Figure 4.6: Temperature, 100 iterations after the source switching off, 2 lines zigzag and $y_{\text{ini}} = 293K$



(a) Computed with the initial vector v^0 (b) Computed with the final vector v (c) Target

Figure 4.7: Temperature, 100 iterations after the source switching off, 3 lines zigzag and $y_{\text{ini}} = 293K$



(a) Evolution of f^{CAL} (b) Evolution of $v_{\rho c_p}$ (c) Evolution of v_λ

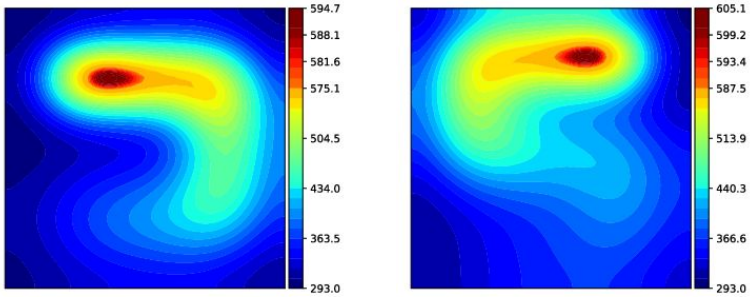
(d) Evolution of v_r (e) Evolution of $v_{\tilde{\beta}}$ (f) Evolution of v_L

Figure 4.8: Evolution of the least square function f^{CAL} as well as the optimization variable v with respect to the iterations

Parameters choice for the numerical applications

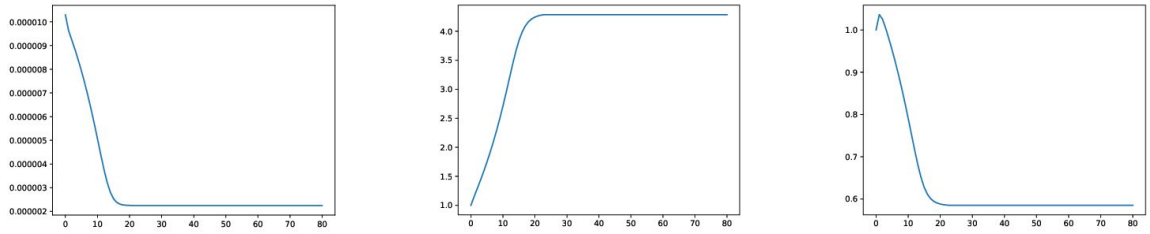
We finally choose this second set of parameters with $\tilde{\beta} = \frac{\lambda}{\Delta Z}$ and

$$\Delta Z = 1.16638\,10^{-4}m, \quad L = 5.85071\,10^{-5}m. \tag{4.3.12}$$



(a) 2 lines zigzag, $y_{\text{ini}} = 293K$ (b) 3 lines zigzag, $y_{\text{ini}} = 293K$

Figure 4.9: Temperature, 100 iterations after the source switching off, with the optimized artificial parameters $\tilde{\beta}$ and L



(a) Evolution of f^{CAL} (b) Evolution of $v_{\tilde{\beta}}$ (c) Evolution of v_L

Figure 4.10: Evolution of the least square function f^{CAL} as well as the optimization variables $\tilde{\beta}$ and L with respect to the iterations

As for the source values, the source radius is fixed to $5.0\,10^{-5}m$ whereas the source power $\tilde{P}$ depends on the test cases.
The model's influence on the path optimization results will be conducted in the steady state context in Chapter 6.

4.4 Steady model presentation

To further simplify the computations, a new model is derived in which each point instantaneously reaches its maximum value when the source is switched on. This leads to steady state considerations, neglecting any time dependence. Instead of optimizing the path with respect to time, we now consider that the optimization variable is a hot thread to be dropped on the working domain. This model is of course not realistic and must be considered as a toy model. However it gives, at a very low computational cost, insights and intuition on the time dependent problem, especially in case of processes allowing high scanning velocity such as EB-PBF. Indeed, this new model can be seen as a transient model with an infinite scanning velocity.

4.4.1 Model derivation

From the time-dependent to the steady state model

To derive the steady state model from the previous one, the heat equation is first averaged in time. Consider the implicit Euler scheme corresponding to (4.3.4), with the time step Δt constant (and N_u time steps, the final computation time being now the source's switching off time). Averaging this equation in time and dividing it by ρc_p leads to

$$\begin{cases} \dfrac{1}{t_{\mathrm{F}}}\displaystyle\sum_{i=0}^{N_u}\left(y_{i+1}-y_i-\nabla\left(\lambda\frac{\Delta t}{\rho c_p}\nabla y_{i+1}\right)+\frac{\tilde{\beta}}{L}\frac{\Delta t}{\rho c_p}\left(y_{i+1}-y_{\mathrm{ini}}\right)\right) \\ \qquad\qquad = \dfrac{\Delta t}{\rho c_p}\dfrac{A\tilde{P}}{\pi}\dfrac{1}{Lt_{\mathrm{F}}}\displaystyle\sum_{i=0}^{N_u}\frac{1}{r^2}\exp\left(-\frac{(x-u_{i+1})^2}{r^2}\right) & \text{in } D, \\ \dfrac{\Delta t}{\rho c_p}\lambda\partial_n y_{i+1}=0 & \text{on } \partial D. \end{cases}$$

Let $\overline{y}$ be the average temperature in time and assume that

$$\overline{y}=\frac{1}{t_{\mathrm{F}}}\sum_{i=0}^{N_u-1}y_{i+1}=\frac{1}{t_{\mathrm{F}}}\sum_{i=0}^{N_u-1}y_i. \tag{4.4.1}$$

Then, $\dfrac{1}{t_{\mathrm{F}}}\displaystyle\sum_{i=0}^{N_u-1}y_{i+1}-y_i=\overline{y}-\overline{y}=0$ and, dividing the heat equation by $\dfrac{\Delta t}{\rho c_p}$ leads to,

$$\begin{cases} -\nabla\left(\lambda\nabla\overline{y}\right)+\dfrac{\tilde{\beta}}{L}\left(\overline{y}-y_{\mathrm{ini}}\right)=\dfrac{A\tilde{P}}{\pi}\dfrac{1}{Lt_{\mathrm{F}}}\displaystyle\sum_{i=0}^{N_u-1}\frac{1}{r^2}\exp\left(-\frac{(x-u_{i+1})^2}{r^2}\right) & \text{in } D, \\ \lambda\partial_n\overline{y}=0 & \text{on } \partial D. \end{cases} \tag{4.4.2}$$

The second approximation involves modifying the source. Instead of considering a Gaussian distribution, the source is assumed to be concentrated on the application point. A new heat equation holds, where χ_u is the Dirac function related to the point u:

$$\begin{cases} -\nabla\left(\lambda\nabla\overline{y}\right)+\dfrac{\tilde{\beta}}{L}\left(\overline{y}-y_{\mathrm{ini}}\right)=A\tilde{P}\dfrac{1}{Lt_{\mathrm{F}}}\displaystyle\sum_{i=0}^{N_u-1}\int_D\frac{1}{\pi}\frac{1}{r^2}\exp\left(-\frac{(x-u_{i+1})^2}{r^2}\right)dx \\ \qquad\qquad = A\tilde{P}\dfrac{1}{Lt_{\mathrm{F}}}\displaystyle\sum_{i=0}^{N_u-1}\chi_{u_{i+1}} & \text{in } D, \\ \lambda\partial_n\overline{y}=0 & \text{on } \partial D. \end{cases} \tag{4.4.3}$$

With Γ the scanning path, the final assumption is then that $\chi_\Gamma=\displaystyle\sum_{i=0}^{N_u-1}\chi_{u_{i+1}}$, where χ_Γ is the lineic Dirac mass along the path Γ such that, $\forall\phi\in H^1(D)$, $\int_D\chi_\Gamma\phi ds=\int_\Gamma\phi ds$. This leads to the steady state heat equation (4.4.4).

$$\begin{cases} -\nabla\left(\lambda\nabla\overline{y}\right)+\dfrac{\tilde{\beta}}{L}\left(\overline{y}-y_{\mathrm{ini}}\right)=\dfrac{A\tilde{P}}{Lt_{\mathrm{F}}}\chi_\Gamma & \text{in } D, \\ \lambda\partial_n\overline{y}=0 & \text{on } \partial D. \end{cases} \tag{4.4.4}$$

To simplify the writting, we set

$$\overline{\beta}=\frac{\tilde{\beta}}{L},\quad \overline{P}=\frac{A\tilde{P}}{Lt_{\mathrm{F}}} \tag{4.4.5}$$

Steady path optimization problem

In the following, the bar notation is given up (keeping in mind that the coefficients involved in the steady heat equation are different from the coefficients involved in the transient heat equation) and the steady state problem is finally given by:

$$\begin{cases} -\nabla\left(\lambda\nabla y\right)+\beta\left(y-y_{\mathrm{ini}}\right)=P\chi_\Gamma & \text{in } D, \\ \lambda\partial_n y=0 & \text{on } \partial D, \end{cases} \tag{4.4.6}$$

with λ the conductivity in the steady state context. Following [10], this leads to $y\in H^1(D)$ uniquely defined. Equivalently to (4.4.6), the temperature $y\in H^1(D)$ is the solution of the variational formulation:

$\forall \phi \in H^1(D)$,

$$\int_D \left(\lambda \nabla y \cdot \nabla \phi + \beta \left(y - y_{\text{ini}}\right) \phi\right) dx - \int_\Gamma P \phi ds = 0. \tag{4.4.7}$$

The corresponding steady state objective function and constraints are:

- Constraint on the solid region: the phase-change constraint becomes $\forall x \in D_S,\ y(x) \geq y_\phi$ resulting in

$$C_\phi(y) = \int_{D_S} \left[\left(y_\phi - y(x)\right)^+\right]^2 dx. \tag{4.4.8}$$

- Constraint on the maximum temperature: the constraint becomes $\forall x \in D,\ y(x) \leq y_{\text{M}}(x)$, resulting in

$$C_{\text{M}}(y) = \int_D \left[\left(y(x) - y_{\text{M}}(x)\right)^+\right]^2 dx. \tag{4.4.9}$$

  Like in the transient case, this constraint can be split into two different constraints: the constraint $C_{\text{M,D}_\text{S}}$ related to the domain D_S and $C_{\text{M,D}\setminus\text{D}_\text{S}}$ for the domain $D \setminus D_S$.

- Objective of minimizing the path length: the final time does not mean much while considering the steady state. The length of the path is considered instead, given by:

$$L_{\text{F}} = \int_\Gamma ds. \tag{4.4.10}$$

  Note that to involve the kinematics, the length could be replaced by the integral of curvatures on the path. This however complicates the resolution and remains part of perspectives.

The optimization problem is finally the same than for the time dependent one given by (4.2.13), but for a slight difference on the objective function: in the steady state case, the objective function considered is directly the path length L_{F} whereas in the unsteady case, the final time t_{F} is chosen. Because the velocity V is constant in a first approach, both choices actually amount to the same.

4.4.2 Calibration of the steady state model

The data provided have been simulated with the same parameters than in Section 4.3 and correspond to the maximum values of the temperature at each point. The computational mesh remains the same whereas the path discretization in this steady state case is fully detailed in Chapter 5.

The objective is now to calibrate the model in order to make the steady state temperature fits the maximum temperature obtained during the time dependent simulations given in Section 4.3. In this calibration, two parameters can be modified: the modified absorption β and the modified power P. This is an arbitrary choice and the conductivity λ could also be considered. Yet to ease the change of material without any new calibration, this parameter is kept unmodified ($\lambda = 15 W m^{-1} K^{-1}$ for the calibration). Like in the transient case, these parameters are normalized and two reference values are introduced (following (4.4.5) with the parameters found in Section 4.3 and (4.4.4), the final time corresponding to the time required to execute all the paths):

$$\overline{\beta}_0 = \frac{15}{1.16638\,10^{-4}} \frac{1}{5.85071\,10^{-5}} W m^{-3} K^{-1}, \quad \overline{P}_0 = \frac{0.3 \cdot 0.4 \cdot 300}{8.32\,10^{-3} \cdot 5.85071\,10^{-5}} W s^{-1} m^{-1}, \tag{4.4.11}$$

Finally the optimization variables are gathered into $v = (v_0, v_1) = \left(v_{\overline{\beta}}, v_{\overline{P}}\right)$, such that

$$\beta = v_{\overline{\beta}} \overline{\beta}_0, \quad P = v_{\overline{P}} \overline{P}_0. \tag{4.4.12}$$

The calibration process is run for the reduced least square function $f^{\text{CAL}} = \int_D \left(\frac{y-\widehat{y}}{\widehat{y}}\right)^2 dx$ (the total objective function is the sum on each path considered of the specific least square function) and the sensitivity of the heat equation with respect to each parameter is

$$\begin{cases} -\nabla \cdot \left(\lambda \nabla \partial_{v_{\overline{\beta}}} y(x)\right) + v_{\overline{\beta}} \overline{\beta}_0 \partial_{v_{\overline{\beta}}} y(x) = \overline{\beta}_0 \left(y_{\text{ini}} - y(x)\right) & \text{in } D, \\ \lambda \partial_n \partial_{v_{\overline{\beta}}} y(x) = 0 & \text{on } \partial D, \end{cases} \tag{4.4.13}$$

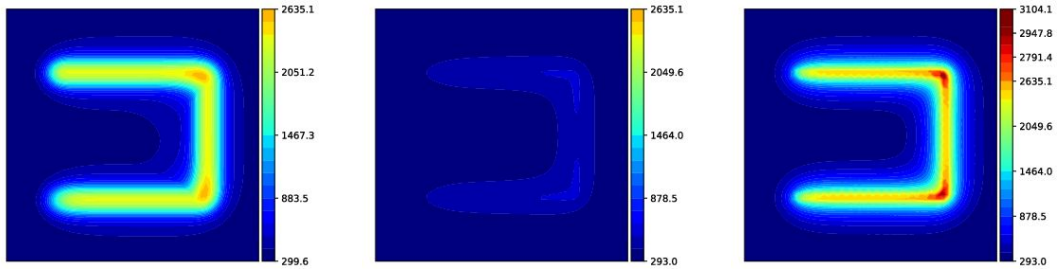
$$\begin{cases} -\nabla \cdot \left(\lambda \nabla \partial_{v_{\overline{P}}} y(x)\right) + v_{\overline{\beta}} \overline{\beta}_0 \partial_{v_{\overline{P}}} y(x) = \overline{P}_0 \chi_\Gamma(x) & \text{in } D, \\ \lambda \partial_n \partial_{v_{\overline{P}}} y(x) = 0 & \text{on } \partial D. \end{cases} \tag{4.4.14}$$

Finally, the algorithm is run for 1000 iterations. Figures 4.11 and 4.12 gives the results. The two first lines show three temperature maps: the target one, the temperature map at calibration initialization and the temperature map after calibration. The first line corresponds to the 2 lines zigzag with $y_{\text{ini}} = 293k$ whereas the second shows the 3 lines zigzag with $y_{\text{ini}} = 293K$. The result tendency is the similar for $y_{\text{ini}} = 793K$. Finally, the objective function and optimization variables are displayed. In the transient context, the time dependency results in preheating the neighborhood of the source. In the steady state model, the time dependency and thus this preheating are removed. As a consequence, the conduction seems lower. To compensate this effect, the power is increased ($v_{\overline{P}} = 22.1$) as well as the absorption coefficient β.
The final coefficients are

$$v_{\overline{\beta}} = 3.084650, \quad v_{\overline{P}} = 22.099501, \tag{4.4.15}$$

leading to

$$\beta = \frac{15}{2.212293\,10^{-9}} W m^{-3} \cdot K^{-1}, \quad P = 300 * 5.447934\,10^6 W s^{-1} \cdot m^{-1}. \tag{4.4.16}$$



(a) Target temperature, 2 lines zigzag, $y_{\text{ini}} = 293K$

(b) Initial temperature, 2 lines zigzag, $y_{\text{ini}} = 293K$

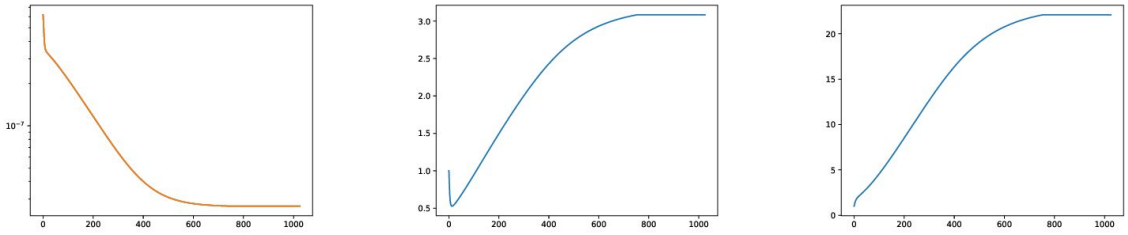
(c) Result, 2 lines zigzag, $y_{\text{ini}} = 293K$

(d) Target temperature, 3 lines zigzag, $y_{\text{ini}} = 293K$

(e) Initial temperature, 3 lines zigzag, $y_{\text{ini}} = 293K$

(f) Result, 3 lines zigzag, $y_{\text{ini}} = 293K$

Figure 4.11: Result from the calibration of the steady state model



(a) Evolution of f^{CAL} (b) Evolution of $v_{\overline{\beta}}$ (c) Evolution of $v_{\overline{P}}$

Figure 4.12: Objective function and variables evolution from the calibration of the steady state model

It is clear that the results before calibration are very different from the target values.

Parameters choice for the steady state numerical applications

The coefficients finally chosen are

$$\beta = \frac{\lambda}{2.212293\,10^{-9}} W m^{-3}, \qquad P = \tilde{P} * 5.447934\,10^{6} W s^{-1} \cdot m^{-1}, \tag{4.4.17}$$

where $\tilde{P}$ is a process choice. As for the conductivity λ, it is related to the material. All the chosen values are given in the Nomenclature.

4.5 Conclusion

In this chapter two different models have been introduced. The transient one aims at modeling the process in two dimensions. The resulting optimization problem is based on thermal considerations only and especially on a temperature control. Resulting from several assumptions, a calibration has been run to enable physical interpretations of the results in the remaining of this work. Two physical cases have been specifically highlighted: the titanium material and the aluminium material, both showing very different conductivities. It is reminded that the model chosen is based on several assumptions which may involve several other problem parameters (related to the source for example) in the calibration process. Improving the model and its calibration is part of the perspectives with, in particular, comparison of the resulting coefficients found for calibrations run on different materials or different heat source properties. Based on the transient model, we have also defined a steady state model that does not involve time dependence. This toy model highly reduces the computational time and leads to a simplified optimization resolution.

CHAPTER 5

PATH DISCRETIZATION

Contents

5.1 Introduction . . . 97
5.2 Working domain and path meshing . . . 98
5.2.1 State of the art . . . 98
5.2.2 Adaptation of a front tracking approach . . . 98
5.3 Path discretization in the steady state context . . . 100
5.3.1 Path description . . . 100
5.3.2 Discretization coherence . . . 100
5.3.3 Discretizing the continuous characteristics of the path . . . 103
5.3.4 Linking the physical mesh and line discretization . . . 104
5.3.5 Differentiation and integration along the line . . . 107
5.4 Path discretization in the transient context - using angles . . . 108
5.4.1 Path description . . . 108
5.4.2 Discretizing the continuous characteristics of the path . . . 109
5.4.3 Linking the physical mesh and the line discretization . . . 109
5.4.4 Differentiation and integration along the line . . . 109
5.5 Path discretization in the transient context - using points . . . 109
5.5.1 Path description . . . 109
5.6 Conclusion . . . 110

5.1 Introduction

To allow for numerical computations, the continuous optimization problem detailed in Chapter 4 must be adapted to a discrete setting. Two main approximations are required: the working domain D must be meshed and the path Γ given by a finite number of parameters. Both choices are crucial in terms of computational cost and accuracy of the results. They are also tightly related: the path carries the source center in the transient context and the source itself in the steady state context. Thus, its discrete representation directly impacts the resulting temperatures that must be controlled. On the other hand, the working domain mesh also impacts these temperatures and thus the path modification.

This "moving interface" issue is well-known in the literature [91, 155]. For example computational fluid mechanics and especially multiphase flow modeling must accurately compute the interface between two flows. The motion of this interface along the simulation led the development of different techniques allowing for accurate and cheap enough computations [136, 155, 171, 183]. This issue is also considered in topology and shape optimization: the interface is the optimization variable itself and is thus moved along the iterations [15, 17]. One can finally mention computer vision and image processing [155].

This chapter presents the discretization choices used in this work. Starting by a brief review of interface motions modeling, Section 5.2 motivates the numerical choices and especially the adoption of a Lagrangian-Eulerian method involving two different meshes: the working domain mesh, called physical mesh, is first developed whereas the path mesh, called path discretization, depends on the context and is detailed in Sections 5.3, 5.4 and 5.5. Section 5.3 expands the path discretization in the steady state context. To keep the representation simple, a broken line is selected, its node points being the optimization variables. Section 5.4 gives a first discretization choice in the transient context. Also based on a broken line, each element has the same length. The control is then on the first point, the number of elements and the angles between each element and the horizontal axis. Finally, Section 5.5 presents a second choice in the transient context, where the broken line is once again defined by its node points.

5.2 WORKING DOMAIN AND PATH MESHING

In this work, the objective is to optimize the source path Γ to control the temperature occurring in a working domain D. While the heat equation is solved in a finite element setting (using FreeFem++ 3.56 [96]), the path must be represented to enable the source computation (4.2.8). We thus focus on a moving interface problem in which the path and working domain discretizations must be chosen.

5.2.1 State of the art

The path Γ is brought to evolve along the optimization process and the representation issue can be compared with free interfaces motions. As previously mentioned, this problematic arises in several domains and is thus widely developed. Based on [102, 136] and references therein, a brief overview is given.

Two main approaches, the Eulerian and the Lagrangian, have been developed to compute the evolution of an interface in time and space.

Lagrangian approaches

In Lagrangian methods, the interface is followed point-wisely. To do so, it is in most cases included into the computational mesh, requiring re-meshing all along the motion (see Figure 5.1). The techniques based on this approach (interface or boundary integrals) provide accurate results. However, the computational costs are high and the topology modifications difficult to handle.

Eulerian approaches

In Eulerian methods, the interface is not point-wisely followed but tracked or captured. The working domain mesh is fixed (or slightly adapted at each interface modification) and an additional variable representing the interface is created. Two main categories of interface variable exist (Figure 5.1):

- interface tracking: this first technique brings back the Lagrangian spirit. Indeed, the interface is fully discretized and, if not modifying the working domain mesh, this discretization follows the interface modifications along the process. Among the several references given in [136, 102], the approach from Tryggvason and al. [183] is specifically of interest in this work.
- interface capturing: in this second technique, the interface is not discretized but reproduced. In marker and cell techniques for example, a marker quantity is introduced and convected in each cell of the working domain mesh. Finding the interface thus amounts to considering this marker function (see SIMP methods in topology optimization [32]). Another example is given by level set methods, that consist in introducing a finite element function related to the working domain mesh, the interface being given by its zero level ([17, 156]). The Marching Cubes method is also widely used and intends to find the interface by characterizing the mesh cells on one side or another [132].

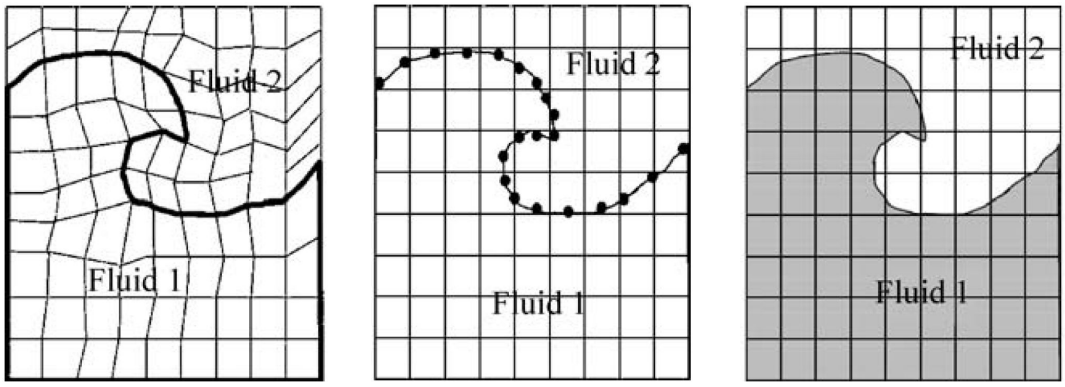
If losing simulation accuracy, both dicretization categories highly reduce the computational costs by sparing re-meshing. Yet, their handlings of the interface topology are distinct. On the one hand, interface capturing naturally allows topology modifications: the markers evolve "on their own" following transports equations and the user only steps in to "capture" the result. On the other hand, in interface tracking, topology modifications imply discretization adaptation and are thus fully controlled.

5.2.2 Adaptation of a front tracking approach

Because of the complexity of Powder Bed Fusion (PBF) processes, the computational costs are a real issue in path optimization. If the several modeling assumptions proposed in Chapter 4 allow for simulations, the numerical representations must be chosen carefully. In particular, Lagrangian approaches are prescribed: re-meshing processes are far too expensive to be included in the optimization algorithms. An Eulerian strategy is thus selected with the introduction of a unique fixed mesh for the working domain (called physical mesh in the following) and a numerical path variable (called discretized path).

Working domain mesh

As mentioned in Chapter 4, the working domain D is chosen in this work to be a square domain of size $1.4mm \times 1.4mm$. The corresponding **fixed** mesh (called **physical mesh** in the following) is structured



(a) Lagrangian approach (b) Eulerian approach - front tracking (c) Eulerian approach - front capturing

Figure 5.1: Main categories in free interface representation for numerical computations [102]

and composed of 12800 squared isosceles triangular elements (mean element size of $2.47\text{e}{-5}m$, Figure 5.2). This working domain is deliberately tiny. Indeed, the metal conductivities mentioned in Chapter 4 are small, resulting in very long scanning paths (around $20mm$ for the considered working domain in the case of the titanium). Considering a small working domain reduces the path possibilities and thus the number of local minima, easing the optimization process. Applying the developed algorithms or testing the obtained results on larger domains is part of the perspectives. This approach could also help designing new scanning patterns.

On this mesh, the different partial differential equations are computed using $P1$ and $P0$ finite elements. These finite elements are created using Freefem++ 3.56 [96]: the mesh being fixed, the classic rigidity matrices involved in the different variational problems are computed and exported. The optimization algorithm is then run into Python (3.6.0), which great flexibility in arrays management and objects creation facilitates the coding. This software mix, developed by Florian Feppon [81], in an additional argument pleading for Eulerian techniques: re-meshing would have required multiple communications between Freefem++ and Python, slowing down the computational process. Besides its convenience, this software mix has also brought slight inaccuracies in the finite element formulations. Indeed, the $P1$-functions are stored in Python using one dimensional arrays that contain the value at each node. The multiplication of $P1-$functions is then approximated by an element by element computation in Python. These inaccuracies have been quantified in the steady state context and appeared to be small enough to be neglected in this work.

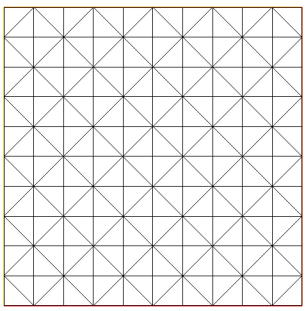


Figure 5.2: Triangular elements meshing elements

Path variable

In addition to the non re-meshing process, path optimization requires a full control of the path topology that must remain an oriented curve. Moreover, in the transient context, the source is applied at points along the path. Both reasons lead to choosing front-tracking methods.

The path mesh, called **path discretization** in the following, depends on the context. Indeed, considering the steady state or transient cases leads to different source representations and thus to different modelings. In any case, three issues must be drawn to attention [183]. A first issue, common in front tracking methods, is related to the evolution of the path discretization's accuracy along the optimization process. Indeed, at each iteration, the path discretization nodes are advected, with potential modification of each element's length. To keep a relevant representation, the discretization must be adapted at each iteration [102, 136, 183]. The resulting line is re-discretized to control each segment's size of the discretized path, removing and adding points to ensure each segment length to be in a chosen range $[d_\text{l}, d_\text{u}]$. In addition to accurate description issues leading to add points, this re-discretization is essential to control the computer memory required by this representation. A second issue corresponds to discretizing functions defined on the path. Some approximation work is hence required to get a correct description of the geometrical functions (normal and curvature for exemple) held by the line. The third point of attention finally consists in the relation between both meshes. Indeed, each carries information that must be shared by the other.

First in the steady state context and then in the transient one with two different discretization choices, a description of the method and of the solutions to the different issues is presented.

Remark 5.1. Choosing front-capturing methods is also possible. Actually, this approach and especially level sets have been used in [51]. Considering them in the thermal steady state context and studying their pertinence in the transient one are part of the perspectives.

5.3 PATH DISCRETIZATION IN THE STEADY STATE CONTEXT

In this section, we focus on the steady state case. The discretization must fulfill the already mentioned requirements. Because there is only little literature on scanning path optimization, the discretization must also remain simple enough to allow for an easy coding and a numerical analysis of the choices taken. In a second time, this model could be made more complex. After setting the parametrization, we detail how to keep its coherence. During the optimization, each discretization element is modified. A redescretization process is defined to ensure the preservation of the description accuracy. We then concentrate on the approximations required to compute the gradients. First, numerical normals and curvatures must be defined. Then, a linking between the path and mesh must be made explicit. Finally, the gradients computation involves differentiating and integrating along the path.

5.3.1 Path description

To satisfy both the Eulerian fixed mesh requirement and the control of the path topology, a representation based on points is chosen. Among the several possibilities, two main ones: the broken lines, corresponding to the actual movement of the source, and the splines, enabling an easier control of the curvature and thus of the kinematics. In this work, the broken line approach has been elected. Yet, this choice remains arbitrary and no comparison between both techniques exist (see [6] for an approach using splines). Such a comparison is part of the perspectives.

In the following, the broken line discretization control points are referred to as "nodes" or simply "points" whereas the segments delimited by two successive points are called "elements" or simply "segments". The path is discretized by N_u nodes, referred to as $(u_i)_{i\in[\![1,N_u]\!]}$, and $= N_u - 1$ elements, referred to as $(S_i)_{i\in[\![1,N_u-1]\!]}$. Each segment S_i is of length $l_i = \|u_{i+1} - u_i\|$ (Figure 5.3). This discretization leads to different technical issues.

5.3.2 Discretization coherence

At each iteration, the path must be re-discretized to control each segment size, allowing as much design freedom as possible without saturating the computer's memory. This step consists in removing and adding points to ensure each segment length to be in the range $[d_\text{l}, d_\text{u}]$. In the steady state context, these bounds are chosen so that $d_\text{u} = 2d_\text{l} = 0.7\Delta x$, with Δx the mean physical mesh element size. The coefficient 0.7 is arbitrarily chosen (for fluid mechanics, the ratio must be around 0.5 [183]) to provide a correct balance between accuracy and computational costs. This coefficient is numerically evaluated in Section 6.5.2 (Chapter 6). A deeper study on its impact on kinematics is also part of the perspectives. Indeed, it is directly related to the points number and thus to the slight path design oscillations.

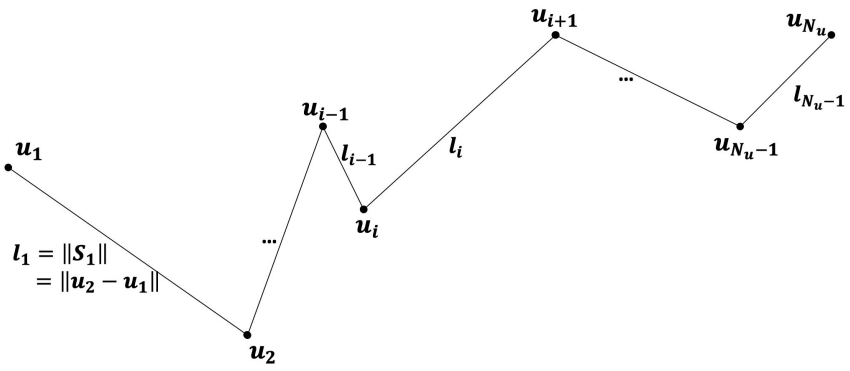


Figure 5.3: Path discretization in the steady state case

Adding process

Consider a segment with length l such that $l > d_{\text{u}}$. The adding process consists in adding equidistant points so that each new element satisfies $l < d_{\text{u}}$. To minimize the number of added elements, this amounts to dividing the segment into $\left[\frac{l}{d_{\text{u}}}\right] + 1$ elements of size $\tilde{l} = \frac{l}{\left[\frac{l}{d_{\text{u}}}\right] + 1}$ (which gives $\frac{d_{\text{u}}}{2} \leq \tilde{l} \leq d_{\text{u}}$).

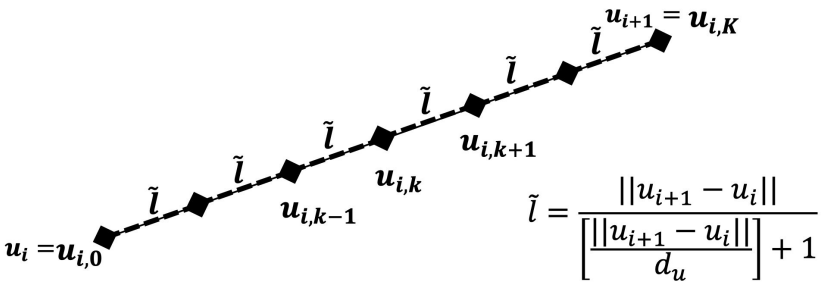


Figure 5.4: Adding process

Remark 5.2. The equidistant approach could be replaced by others such as Legendre methods for example (to increase the accuracy of the approximation [183]).

Deleting process

Consider a segment with length l such that $l < d_{\text{l}}$, not long enough for the discretization. Then, one of the endpoints of the segment is simply removed. The process is the following: starting from the last point of the line u_{N_u} and going backward, the first point u_i ($i \in [\![1, N_u - 1]\!]$) for which $\|u_{N_u} - u_i\| \geq d_{\text{l}}$ is found (Figure 5.5 (a-b)). The line is modified removing the intermediary points and updating the segments' length before running the same process starting from the point u_i (Figure 5.5 (c)). Once the starting point u_1 is reached (Figure 5.5 (d)), the process is stopped (u_1 cannot be deleted) and run again forward, starting from u_1 and stopped when the last point is reached (Figure 5.5 (e-f)). Note that this algorithm always keep at least two points in the path, even if the distance between them is smaller than the threshold (Figure 5.5).

Remark 5.3. Starting the process from the last point and going backward is an arbitrary choice, coming from coding simplifications.

Remark 5.4. This algorithm is used if the path does not carry any physical value. In Chapter 8, each element of the path carries a power value ζ. Removing the elements thus gets more complicated and a second process is given in section 8.3.1 (Chapter 8).

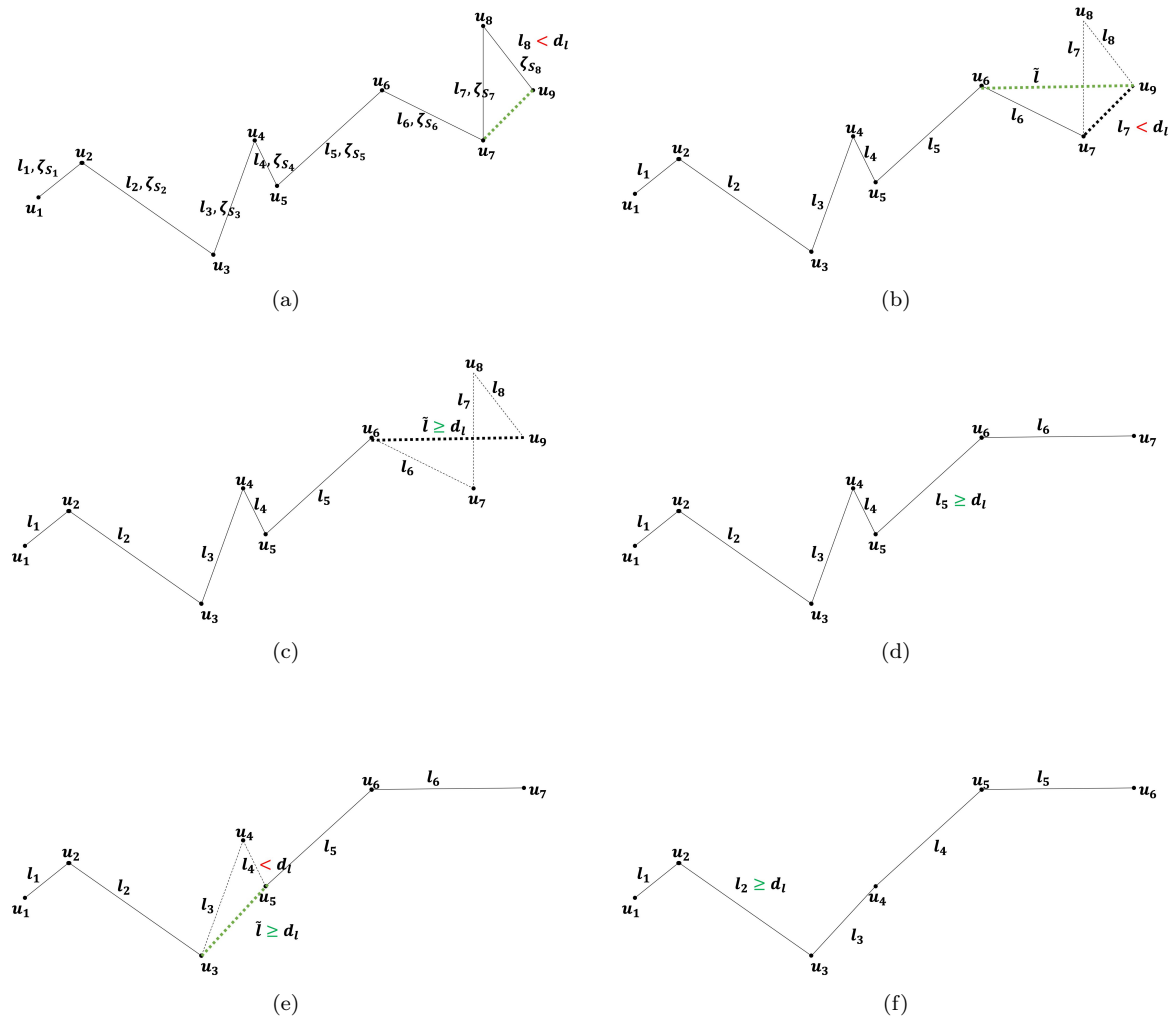


Figure 5.5: Deleting process: (a) Starting from the end point, the last segment is too small. The segment from the last point to u_7 is also too small. (b) The point u_6 is the first point enabling a long enough segment: u_7, u_8 are removed and the point numbering modified. (c) The length l_5 is too small. Starting from u_6, the next point allowing a long enough segment is looked for. (d) The length l_2 and the length between u_1 and u_3 are too small. The point u_2 is removed. The point u_1 is kept. (e) Starting from u_1, the process is run again. (f) Resulting line.

Case of two-points path

The re-discretization process does not remove the first and last points. Thus, paths discretized with only two points can exist. In case the distance between these two points is smaller than the threshold d_l, the path is deleted. This issue does not happen often but in path topology optimization presented in Chapter 8.

Re-discretization algorithm

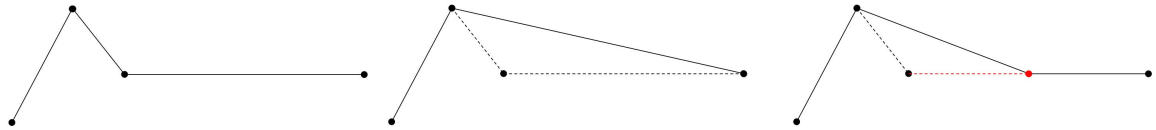
The discretization algorithm is finally the following: to keep as mush freedom as possible, the adding process is first run. Indeed, starting by deleting points could erase inflections (illustrated by Figure 5.6). Then, the deleting process is applied. Finally, to make sure that each element's length is in $[d_\mathrm{l}, d_\mathrm{u}]$ the adding process is run again. The algorithm follows Algorithm 5.1.

Remark 5.5. This rediscretization process could be used to control the curvature. Indeed, Figure 5.6 (b) would be better if the curvature must be reduced. An adapted choice of d_u and d_l could also provide such a control.

Remark 5.6. Note that during this process, the number of nodes vary and N_u in Algorithm 5.1 is not

1 Add points so that $\forall i \in [\![1, N_u]\!]$, $l_i \leq d_{\mathrm{u}}$
2 Delete points so that $\forall i \in [\![1, N_u]\!]$, $l_i \geq d_{\mathrm{l}}$
3 Delete two points lines
4 Add points so that $\forall i \in [\![1, N_u]\!]$, $l_i \leq d_{\mathrm{u}}$

Algorithm 5.1: Rediscretization algorithm



(a) Line before discretization (b) Discretization process starting by the deleting step (c) Discretization process starting by a first adding process

Figure 5.6: Difference in the process in case of an additional first adding step. In (c), the inflection of the path is kept whereas it has disappeared in (b).

constant.

5.3.3 Discretizing the continuous characteristics of the path

During the optimization process, the tangent, normal and curvature of the path at each node point are required, which, considering a broken line, involves approximations. To begin with, the oriented sequence of tangent vectors $(\tau_0, ..., \tau_{N_u-1})$ is modeled with

$$\forall i \in [\![1, N_u - 1]\!], \quad \tau_i = \frac{u_{i+1} - u_i}{l_i}. \tag{5.3.1}$$

Consider a path Γ in the steady state context. Following the broken line representation with N_u nodes, this path is fully described by the oriented sequence of points $(u_1, ..., u_{N_u})$, or equivalently by the starting point u_1, a sequence of length $(l_1, ..., l_{N_u-1})$ and an oriented sequence of tangent vectors $(\tau_0, ..., \tau_{N_u-1})$ with

$$\forall i \in [\![1, N_u - 1]\!], \quad \tau_i = (u_{i+1} - u_i)/l_i.$$

The normal to a segment n_i^{S} is naturally defined as the normalized vector so that the basis $(\tau_i, n_i^{\mathrm{S}})$ is orthonormal and positively oriented. The normal to a point, n_i^u, is then defined by the average between the normals to both neighboring segments, weighted by their length:

$$\begin{cases} n_1^u = n_1^{\mathrm{S}} \\ n_i^u = \dfrac{l_{i-1} n_{i-1}^u + l_i n_i^u}{l_{i-1} + l_i} \quad i \in [\![2, N_u - 1]\!] \\ n_{N_u}^u = n_{N_u-1}^{\mathrm{S}}. \end{cases} \tag{5.3.2}$$

As for the curvature κ, Gauss approach has been chosen [37, 169] (osculating circles and length variations methods turn out to give similar results). Let $\mathfrak{a}_i$ be the angle between the horizontal (e_{X} vector) and the tangent vector τ_i and let ψ_i be the angle between the vectors τ_i and τ_{i+1} (see Figure 5.7). Then, $\forall 2 \in 1, .., N_u - 1$,

$$\psi_i = \mathfrak{a}_i - \mathfrak{a}_{i-1} = \int_{u_{i-1}}^{u_i} \kappa(s) ds.$$

Choosing a linear interpolation for the curvature (for the segment i, $\kappa(t) = (1-t)\kappa_{i-1} + t\kappa_i$ with $(\kappa_i)_{[\![,N_u]\!]}$ the curvature at each point), one gets that, $\forall i \in 1, .., N_u - 1$,

$$\psi_i = l_i \int_0^1 \left((1-t)\kappa_{i-1} + t\kappa_i\right) dt = l_i \frac{\kappa_{i-1} + \kappa_i}{2}$$

Finally, from the closing property of curvature, $\int \kappa dl = \sum_2^{N_u - 1} \psi_i$, one can deduce the curvature $\forall i \in 2, .., N_u - 1$ [169]

$$\kappa_i = \frac{2}{l_i + l_{i-1}} \psi_i. \tag{5.3.3}$$

As for the values at the end points, they are set to 0 and $\kappa_1 = \kappa_{N_u} = 0$.

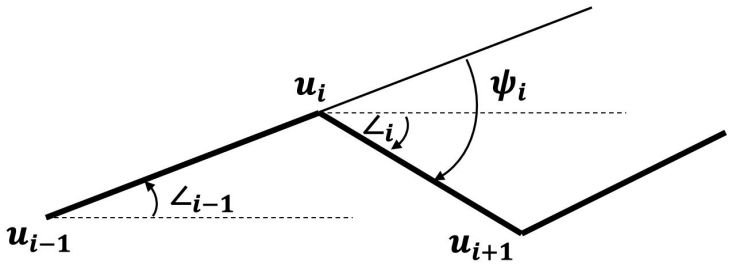


Figure 5.7: Computation of a discrete curvature

5.3.4 Linking the physical mesh and line discretization

Both the physical mesh and the path discretization hold information that must be shared. The path Γ defined by its discretization, carries the heat source. In order to solve the heat equation on the physical mesh, this heat source must be expressed in the physical mesh. On the contrary, many quantities exist on the physical mesh (the temperature or its derivatives for example) and their values along the path may be required [183].

From the path discretization to a $P0$-heat source function

From the path Γ, a function defined on the mesh must be created to solve the finite element formulation of the heat equation. For coding reasons, we chose to build a P0-function such that for each cell of the mesh, the source energy carried by this cell is computed. To do so, each mesh cell crossed by the path is identified and the length included in this cell computed. The value in each cell is finally normalized by the cell area, leading to a $P0-$source function shown by Figure 5.11. Note that if the path follows an edge, both cells involved are identified and half of the corresponding length is attributed to each.

The algorithm is the following: for each segment $S_i = [u_i, u_{i+1}]$, we start from the end point $u_i = u_i^0$, identify the cell $\mathbb{T}_i^0$ to which it belongs (Figure 5.8 (a)), and look for the next intersection u_i^1 between the path and edge or summit of the mesh (Figure 5.8 (b)). The length of the subsegment $[u_i^0, u_i^1]$ is then computed and related to the identified cell. From the intersection, the next cells crossed by the segment are identified, followed by the sought for the next intersection (Figure 5.8 (b)). This process is repeated until the second endpoint of the segment is reached. In case of a subsegment following an edge, both cells involved are identified (Figure 5.8 (e))

The algorithm iteratively realizes two steps: identifying the triangles involved and computing the next intersection. Both process are detailed for a segment $[u_i, u_{i+1}]$, which tangent τ is the unitary vector oriented from u_i towards u_{i+1}.

Identifying the list of triangles $\mathbb{T}_i^k$: consider that the list of triangles is found until a point u_i^k. Three different cases can hold:

- the point u_i^k is in the middle of a triangle $\mathcal{T}$. Then, the next intersection between the segment and an edge or summit of the physical mesh belongs to this same triangle and the list $\mathbb{T}_i^k$ is reduced to $\mathcal{T}$. This situation might happen for $k = 0$ and $u_i^k = u_i$ only.
- the point u_i^k belongs to an edge of the physical mesh (but not to a summit). The segment can go into two different triangles $\mathcal{T}_1$ and $\mathcal{T}_2$ (Figure 5.9 (a-b)). The scalar product between the tangent to the segment and the normal to the edge pointing towards $\mathcal{T}_1$ is realized. If the result is positive, the list $\mathbb{T}_i^k$ is reduced to $\mathcal{T}_1$. If the result is negative, the $\mathbb{T}_i^k$ is reduced to $\mathcal{T}_2$ (Figure 5.9 (a)). In case the scalar product is small (the threshold 5.0e−7 is fixed), the segment follows the edge and both triangles are added to the list (Figure 5.9 (b)). Note that the last situation can occur for $k = 0$ and $u_i^k = u_i$ only. Else, the point u_i^k would be a summit. Note also that for any point but the first one, finding the next triangle is trivial: the segment came from a cell and goes to another. Yet, for the first point, this is not easy and we have chosen to apply the same process in any situation.

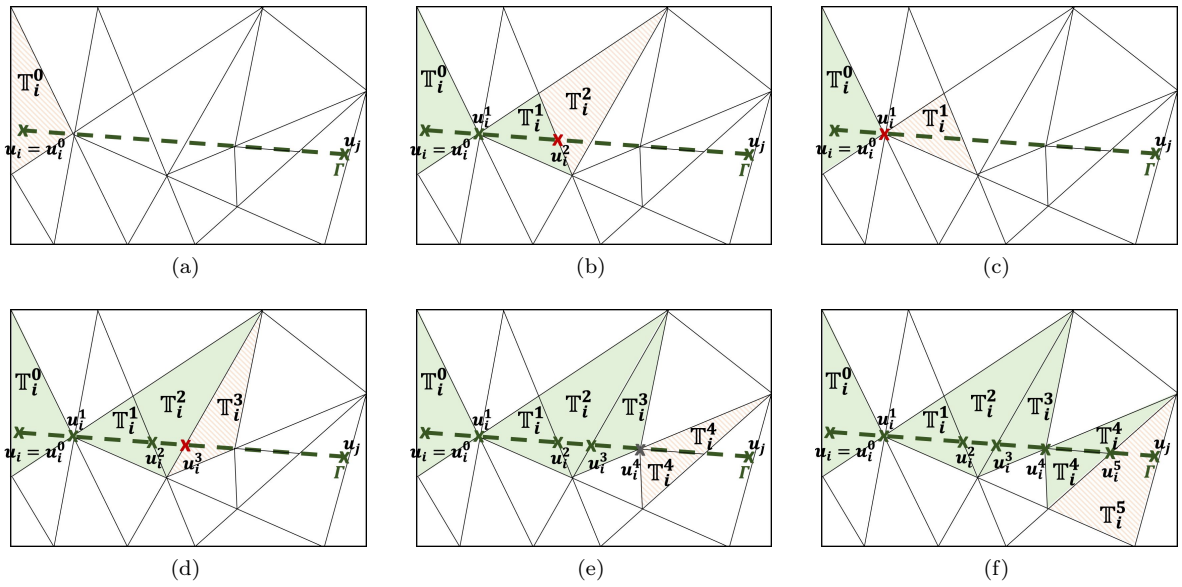
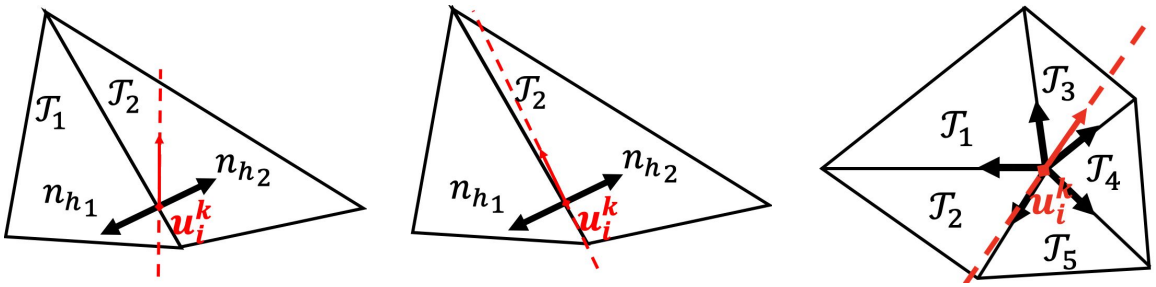


Figure 5.8: Transforming the path Γ into a $P0$-heat source function

- the point u_i^k belongs to a summit of the physical mesh. The segment can go in all the triangles $\mathcal{T}_{i\in[\![1,M]\!]}$ to which this summit belongs. The tangent to each of the edges involved $\tau^{edge}_{i\in[\![1,M]\!]}$, oriented from the summit towards the second endpoint of the edge, is computed. We find the closest edge to the segment, i.e. $i = \underset{i\in[\![1,M]\!]}{\arg\min}\langle \tau, \tau_i^{edge}\rangle$ (Figure 5.9 (c)). This edge belongs to the next triangles involved and the process applied for a point belonging to an edge is then applied.



(a) Point on the edge: the triangle involved is $\mathcal{T}_2$

(b) Point on the edge: to triangles involved

(c) Point on the summit

Figure 5.9: Process to find the triangles involved. (a-b) From a point on the edge, (c) From a point on the summit

Computing the next intersection u_i^{k+1}: consider that point u_i^k as well as the list of triangle into which looking for the next intersection $\mathbb{T}_i^k$ known, and assume that the point u_{i+1} does not belong to any triangle of the list (else this is over). Three different situations can occur:

- if there is more than one element in the triangle list $\mathbb{T}_i^k$, the segment follows an edge and the next intersection if thus one of the edge end point. Determining which one is easy using the direction of the segment tangent.
- if there is only one element $\mathcal{T}$ in the triangle list $\mathbb{T}_i^k$ and if u_i^k is a summit, the next intersection is on the edge of $\mathcal{T}$ that does not include u_i^k as an endpoint.
- if there is only one element in the triangle list $\mathbb{T}_i^k$ and if u_i^k is not at a summit, we need to find the edge to which the next point belongs. Consider that the triangle is characterized by its summits P_1, P_2 and P_3.

Proposition 5.1. *There exists $j \in [\![1,3]\!]$ such that the ray starting from u_j^k with direction τ crosses with the straight line going through this summit and oriented as the opposite edge, i.e $\exists j \in [\![1,3]\!]$, $t_j > 0$, $A_i \in \mathbb{R}^2$ such that,*

$$\begin{cases} u_i^k A_j = t_j d \\ P_j A_j \wedge P_{j+1 \bmod 3} P_{j+2 \bmod 3} = 0, \end{cases}$$

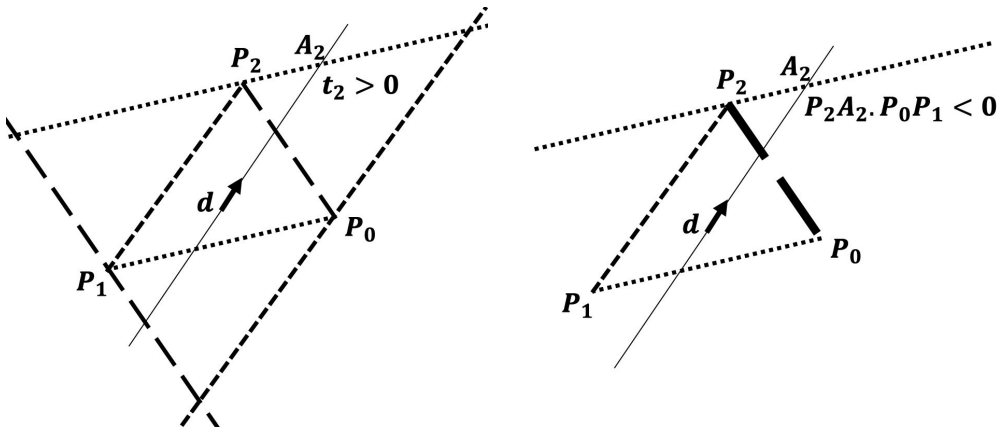
The point P_j belongs to the edge cut by the half straight line $\{u_i^k + x\tau, x \in \mathbb{R}^+\}$.

PROOF.
The easy step to prove is the existence of a point such that t_i is strictly positive. Indeed, the parallel to each side going through the opposite summit define a triangle and any ray starting in the triangle must cut one of the sides of the bigger triangle (Figure 5.10 (a)). Then, the point P_j for which the property holds belongs to the next edge cut. Indeed, the starting point u_i^k is situated between the straight line generated by the edge $P_{j+1 \bmod 3} P_{j+2 \bmod 3}$ and parallel straight line going through P_j. Since t_j is positive, the segment is oriented towards the summit P_j and the next intersection cannot belong to the edge $P_{j+1 \bmod 3} P_{j+2 \bmod 3}$. It thus belongs to one the two edges involving P_j (Figure 5.10). □

Renaming the points if needed, we consider that the point found by the step is P_2 (whereas the others are P_0 and P_1). To find the second end point of the edge, we consider the scalar product $P_2A_2 \cdot P_0P_1$ and we conclude by the following statement, illustrated by Figure 5.10 (b):

$$\begin{cases} P_2A_2 \cdot P_0P_1 > 0 & \Longrightarrow \quad \text{the edge is } \{P_1, P_2\} \\ P_2A_2 \cdot P_0P_1 < 0 & \Longrightarrow \quad \text{the edge is } \{P_0, P_2\} \\ P_2A_2 \cdot P_0P_1 = 0 & \Longrightarrow \quad \text{the segment gets out of the triangle through the summit } P_2 \end{cases}$$

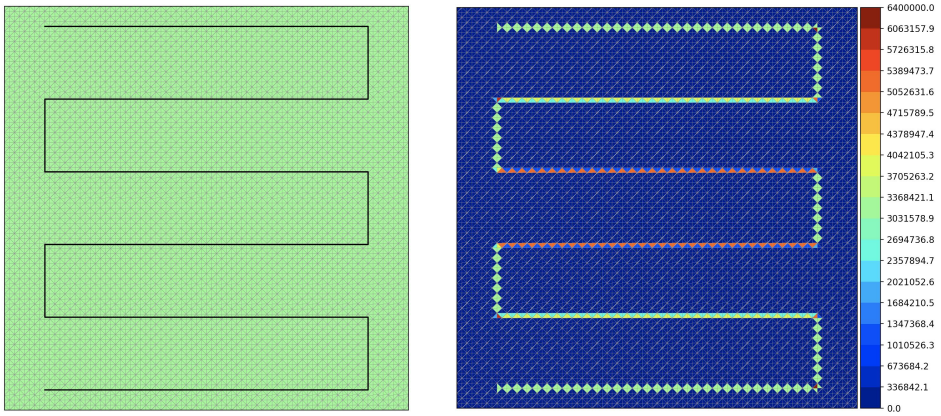
Note that this requires the computation of A_2. However, this has already been done to characterize the summit P_2.



(a) P_2 belongs to the edge cut by the segment (b) $\{P_0, P_2\}$ is the edge cut by the segment

Figure 5.10: Process to find the next intersection

Creation of the $P0$-function: once the list of triangles has been found, it is possible to compute, for each triangle of the physical mesh, the "amount of path" crossing them (for each triangle, the length of each segment crossing it is added and in case of a segment following an edge, each of the triangle affected get half of the length). Finally, this is normalized by the area of the triangle, leading to a heat source which thickness should thus not exceed one element. The function is finally multiplied to the source power given the path shown in Figure 5.11.



(a) (b)

Figure 5.11: Path Γ on the left and $P0$-heat source on the right, the color scale corresponding to the power in each cell

From $P0$ and $P1$-functions defined on the physical mesh to functions defined on the path

To optimize the path, an advection vector must be computed, involving the functions defined on the physical mesh and especially the temperature and the adjoint used for the derivatives computation (see Section 3.2.3 in Chapter 3 and Section 6.3.1 in Chapter 6). These functions must be known on the path Γ which implies approximations. Let F be a $P1$-function and G a $P0$-function defined on the physical mesh. At each point u of the path discretization must be associated a value. As for the $P0$-function G, this is quite easy: indeed, the physical mesh elements to which belongs the point u are computed and the mean value of G on these triangles is associated to the point u (the mean value is required since the point u might belong to an edge or coincide with a node of the physical mesh). As for the function F, it is first turned into a $P0$-function: for each element of the physical mesh, the average of the values of F at the summits is computed. Then, the same process that for a $P0$-function is applied.

5.3.5 Differentiation and integration along the line

To compute the derivatives, differentiation and integration along the path are required. For example, the regularization process described in Section 3.2.4 (Chapter 3) involves the resolution of a PDE along the path. Both integration and differentiation must be discretized .

As for finite element functions on a classic mesh, numerically differentiating and integrating variables depend on their formulation along the path. First assume that the continuous function $f \in H^1(\Gamma)$ is easy to characterize on the nodes. Then, a $P1-$formulation is matched to this function with the definition of $\overline{F} = (F_i^u)_{[\![1,N_u]\!]}$, the vector of the values at each node. This is for example the way of representing the temperature (and any function defined on the mesh) along the path. Then assume that the continuous function $g \in H^1(\Gamma)$ is easy to characterize on the segments. Then, a $P0-$formulation is matched to this function with the definition of $\overline{G} = \left(G_i^S\right)_{[\![1,N_u-1]\!]}$, the vector of the values at each segment. For example, the derivative of a $P1-$function along the curve (such as $\partial_\tau y$) satisfies this definition. In Chapter 8, each segment will also be given a power value ζ.

Integration

Assume that the function to integrate is fg. Then,

$$\int_\Gamma fgds = \sum_{i=1}^{N_u-1} l_i G_i^S \frac{F_{i+1}^u + F_i^u}{2}.$$

Set $\tilde{f}$ another function defined on the path and known at each node. Then,

$$\int_\Gamma f\tilde{f}gds = \sum_{i=1}^{N_u-1} l_i G_i^S \frac{F_{i+1}^u \tilde{F}_{i+1}^u + F_i^u \tilde{F}_i^u}{2}.$$

Differentiation

First consider the function F known at points. Then, the derivative is known at each segment ($P0$-function) and one has, $\forall i \in [\![1, N_u - 1]\!]$:

$$\partial_\tau F_i^S = \frac{F_{i+1}^u - F_i^u}{l_i}.$$

Integrating a derivative finally leads to:

$$\int_\Gamma \partial_\tau f ds = \sum_{i=1}^{N_u-1} l_i \partial_\tau F_i^S = \sum_{i=1}^{N_u-1} \left(F_{i+1}^u - F_i^u\right) = F_{N_u}^u - F_1^u.$$

Consider now function G known at each segment. The derivative corresponds to a differentiation with respect to the segment.

$$\partial_\tau G_1^u = -G_1^S, \qquad \forall i \in [\![2, N_u - 1]\!],\ \partial_\tau G_i^u = \frac{l_i G_i^S - l_{i-1} G_{i-1}^S}{l_i + l_{i-1}}, \qquad \partial_\tau G_{N_u}^u = G_{N_u-1}^S,$$

The values given to the end points are arbitrarily chosen so that:

$$\int_\Gamma \partial_\tau g ds = \sum_{i=1}^{N_u-1} l_i \frac{\partial_\tau G_i^u + \partial_\tau G_{i+1}^u}{2} = l_{N_u-1} G_{N_u-1}^S - l_1 G_1^S.$$

Remark 5.7. These approximations almost consist in considering the functions known on segments as $P0$-functions and the one defined at each point as $P1$-functions. Yet, two differences are to be noted. First, the differentiation of a function known at segments can be done and result in a function known at points. Then, the product of two functions defined at points is modeled by a function characterized by the value at each point. This value is the product of the values of each of the two functions at each point. Hence, this product does not fully respect the multiplication rules of $P1$-functions.

5.4 PATH DISCRETIZATION IN THE TRANSIENT CONTEXT - USING ANGLES

In the steady state context, the path is independent in time, simplifying the relations between the points it is composed of. In the transient context, however, the problematic is different and the discretization must be chosen accordingly. A first idea consists in following the optimal control literature [56, 111, 196, 197], choosing acceleration or velocity vectors as optimization variables. The velocity V of the source assumed constant in this work ($v = 1ms^{-1}$), the path is characterized by the starting point $\tilde{u}$, the final time t_F and the tangent of the path τ. The source center follows the ordinary equation (4.2.3), recalled here:

$$\begin{cases} \partial_t u = V\tau(t) \quad t \in [0, t_\mathrm{F}] \\ u(0) = \tilde{u}. \end{cases} \tag{5.4.1}$$

5.4.1 Path description

Following the steady state choices, a broken line description is used. Yet, if this line was defined by its nodal points, we now choose to control it through the tangent direction, modeled by the angle $\mathfrak{a}$ it makes with the horizontal at each point ($\mathfrak{a} = (\mathfrak{a}_1, ..., \mathfrak{a}_{N_u-1})$). Every segment element has the same fixed length l, which highly simplifies the heat equation's resolution (see Figure 5.12). Indeed, fixing the computation time step $\Delta t = l/V$ and keeping a fixed physical mesh, the rigidity matrix corresponding to the heat equation remains the same along the whole optimization process and only needs to be inverted once. To complete the path description, the starting point $\tilde{u}$ is added to the optimization variables as well as the

final time t_F, given by the number of segments $N_u - 1$ (and thus the number of broken line nodes N_u). The sequence of discrete centers of the source point is finally given by

$$\begin{cases} u_1 = \tilde{u} \\ u_{i+1} = u_i + V\left(\cos(\mathfrak{a}_i), \sin(\mathfrak{a}_i)\right) \quad i \in [\![1, N_u - 1]\!]. \end{cases} \tag{5.4.2}$$

The interest of this method has been pointed out: the resolution of the physical equations is eased and faster. Moreover, the segments' length being completely controlled, there is no need for any re-discretization at each iteration. On the other hand, the final time is only controlled in a discrete way. Indeed, as an optimization variable, it might be increased or decreased depending on the gradient of the optimization problem. In both cases, to keep the elements' length l fixed, a finite integer number of segments is added or removed to the path (process fully explained along the development of the path optimization in Chapter 7).

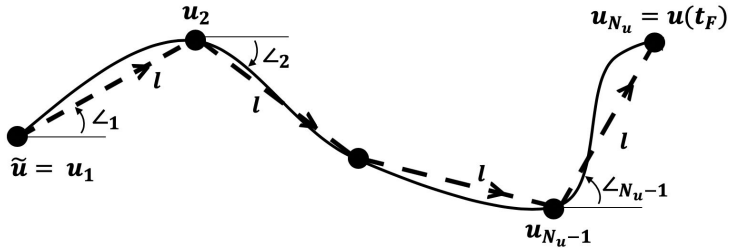


Figure 5.12: Angle based path discretization in the transient context

If no discretization is required, computing numerical normals and curvatures, linking the path and the mesh and discretizing and integrating along the path are still required.

5.4.2 Discretizing the continuous characteristics of the path

The computation of the normal to segments and points correspond to the process described in Section 5.3.3. As for the curvature, Gauss approximation is also chosen, with the angle ψ_i still satisfying $\forall i \in [\![1, N_u - 1]\!]$, $\psi_i = \mathfrak{a}_i - \mathfrak{a}_{i-1}$.

5.4.3 Linking the physical mesh and the line discretization

In the transient context, this issue gets very simple. Indeed, the laser beam is now modeled by a Gaussian at each time step. The source term directly is given by (4.2.8) as a $P1-$ function on the physical mesh.

5.4.4 Differentiation and integration along the line

In the transient context, integrating and differentiating along the path now correspond to integrating and differentiating with respect to time. The process given in the steady state case is still used, replacing the length l_i by $\Delta t_i = l_i/V$ ($= \Delta t$ since the time step is fixed in the angle based representation).

5.5 Path discretization in the transient context - using points

If efficient in terms of computational cost, the angle-based formulation presents some drawbacks. Indeed, the final time is dealt with in a very artificial way, preventing smooth lengthening and shrinking of the path. Assume for example that the beginning of path must be lengthened but the end should not be modified. Then, the algorithm must first add elements at the end to increase the path length, then modify the first angles to lengthen the path where it is needed, to finally change the angles at the end of the path to bring this end of the path as it were. These drawbacks are further illustrated when optimizing the path, in Chapter 7. A second approach, point-based approach, is thus developed.

5.5.1 Path description

This new approach still consists in describing the path by a broken line. Alike the steady state case, the control points are the optimization variables while the constant segment length property is given up.

The notations introduced in the steady state context are kept in this section. The heat equation is here again solved at each node point but the computation time step Δt is not constant (Figure 5.13). Indeed, the discretized heat equation is now given by, $\forall i \in [\![1, N_u]\!]$,

$$\begin{cases} \rho c_p \dfrac{y_{i+1} - y_i}{\Delta t_i} - \nabla\left(\lambda \nabla y_{i+1}\right) + \beta\left(y_{i+1} - y_{\text{ini}}\right) = P \exp\left(-\dfrac{(x - u_i)^2}{r^2}\right) & \text{in } D, \\ \lambda \partial_n y_{i+1} = 0 & \text{on } \partial D, \\ y_0 = y_{\text{ini}}, \end{cases} \tag{5.5.1}$$

with $\Delta t_i = \|u_{i+1} - u_i\|$, $i \in [\![1, N_u - 1]\!]$ and Δt_0 fixed.

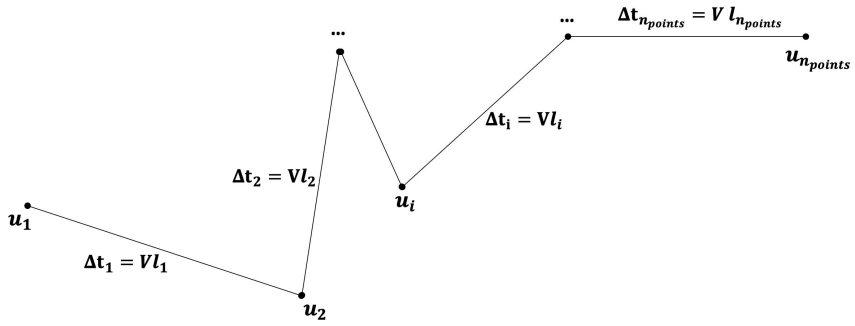


Figure 5.13: Points based path discretization in the transient context

In this new context, the discretization of the continuous properties of the line and the differentiation and integration along follow the method proposed in Section 5.3. As for linking the physical mesh with the path discretization, the method is the same as presented in Section 5.4.3.

5.6 CONCLUSION

While Chapter 4 described the model considered and the assumptions it is based upon, this chapter focuses on the numerical adaptation of this model for efficient and accurate optimization. Both of them present choices made along this PhD to allow for numerical optimization.

If it seemed very natural to consider an Eulerian approach and thus to uncouple the path and physics related representations, a first choice consists in using front tracking methods. Instead of working with continuous functions capturing the path, it discretization elements constitute the new variable optimization. A second choice is the discretization itself. Indeed, in any case, a broken line is used but other representations and especially splines could be considered. Adapting the presented algorithms to splines and results comparison is part of the perspectives. Inside the broken line category, the optimization variables must be defined. Two possibilities have been given in the transient case but many more could be used and might result in improved results. Finally, all these discretizations rest on several coefficients (number of physical mesh elements, size of discrete line elements, ...). Chapters 6 and 7 further consider the impact of some of them on the optimization. Yet, a deeper study would be of high interest to better understand their role and choose them to reach efficient compromises between accuracy and computational costs.

CHAPTER 6

PATH OPTIMIZATION IN THE STEADY STATE CONTEXT

Contents

6.1 Introduction . . . 111
6.2 Steady path optimization problem . . . 112
6.3 Shape derivative computation . . . 112
6.3.1 Computation in the continuous settings . . . 112
6.3.2 Regularization . . . 114
6.4 Numerical algorithms . . . 114
6.4.1 Numerical Augmented Lagrangian algorithm . . . 114
6.4.2 Numerical null space gradient flow algorithm . . . 116
6.5 Numerical results . . . 116
6.5.1 First results . . . 117
6.5.2 Impact of the numerical choices . . . 120
6.5.3 Impact of the physical model . . . 125
6.5.4 Building the whole working domain . . . 129
6.6 Initialization and geometry to build impact for the aluminium . . . 130
6.7 Initialization and geometry to build impact for the titanium . . . 137
6.7.1 Scanning a square . . . 137
6.7.2 Scanning complex geometries . . . 139
6.8 Conclusion . . . 144

6.1 Introduction

The previous chapters have given all the settings required for path optimization. We have defined two simulation models (transient and steady state) and set for each of them an optimization problem (Chapter 4). We have then adapted each framework to numerical computations (Chapter 5). The optimization tools presented in Chapter 2 can now be applied to path generation.

We focus on the steady state model. Getting rid of time considerations, this model is more simplified than the transient one. However, it still holds some physical meaning as the limit case of infinite velocity source. In the literature, many mechanical simulations actually model the scanning of each layer using a steady state assumption, involving time for switching from one layer to another only (for example in inherent strain methods [51, 124]). Most importantly, it provides cheap simulations and thus helps to point out and solve algorithmic issues that will be raised in the transient case. Because of the steadiness assumption, classical shape optimization theory can be used. Indeed, there is no time dependency between the different nodal points that compose the path and the boundary variation method given in Chapter 3 perfectly fits this specific case.

In Section 6.2 we briefly recall the steady state optimization problem introduced in Chapter 4. Then, in Section 6.3, we differentiate the functions involved and obtain their numerical gradient. Section 6.4 details the adaptation of the Augmented Lagrangian method and of the null space gradient algorithm to this applied context. Numerical results are finally provided in Section 6.5: the different algorithms, the discretization choices, the impact of the different physical coefficients on the optimized path are analyzed. Finally, in the aluminium context in Section 6.6 and in the titanium context in Section 6.7, several patterns are chosen as initialization to study both their efficiency and how they are modified, increasing the insight of the "good path" notion. The geometry of the part to build is also modified.

6.2 STEADY PATH OPTIMIZATION PROBLEM

This section recalls the steady state model defined in Section 4.4, which must be completed by the numerical tools given in Section 5.3. In the steady state context, the path Γ corresponds to a hot thread laid on the powder layer. The working domain D is split into the part to build $D_S \subset D$ and the domain that must remain powder $D \setminus D_S$ (Section 4.2.3, Chapter 4). The problem is

$$\min_{\Gamma \in \mathcal{G}} L_{\mathrm{F}} = \int_{\Gamma} ds, \quad \text{such that} \begin{cases} C_\phi(\Gamma) = \displaystyle\int_{D_S} \left[(y_\phi - y)^+\right]^2 dx = 0, \\ C_{\mathrm{M}}(\Gamma) = \displaystyle\int_{D} \left[(y - y_{\mathrm{M}})^+\right]^2 dx = 0, \end{cases} \tag{6.2.1}$$

with $y \in H^1(D)$ solution of

$$\begin{cases} -\nabla\left(\lambda \nabla y\right) + \beta\left(y - y_{\mathrm{ini}}\right) = P\chi_\Gamma & \text{in } D, \\ \lambda \partial_n y = 0 & \text{on } \partial D, \end{cases} \tag{6.2.2}$$

or, in its variational form, $y \in H^1(D)$ such that $\forall \phi \in H^1(D)$,

$$\int_D \lambda \nabla y \cdot \nabla \phi + \beta\left(y - y_{\mathrm{ini}}\right)\phi dx - \int_\Gamma P\phi ds = 0. \tag{6.2.3}$$

In this steady state model four constraints are considered: the phase constraint, the maximum temperature constraint, the partial differential heat equation that the temperature must satisfy and finally the box constraint $\Gamma \subset D$. Note that the latter could be transformed into inequality constraints. However, their formulation would not be straightforward. For the partial differential equation constraint, we will use the adjoint method presented in section 3.2.3.

To optimize the steady state path, the sensitivity of the cost and constraints are required: the shape optimization tools and especially the boundary variations differentiation propositions presented in Chapter 3 will be used.

Remark 6.1. On the contrary to the example of domain optimization performed in section 3.3.2, the curve to optimize is not represented by a level set anymore, but by a discrete broken line. Indeed, as stated in Chapter 5, this facilitates the control of the path topology. As in section 3.3.2, the shape derivative and gradient are computed, leading to an update velocity. However, instead of applying the advection equation (3.3.2) from section 3.3.2, the velocity is discretized and each nodal points of the broken line moved accordingly. In the approach chosen, the derivative is thus computed for the continuous path (before discretization) and then discretized. In this broken line context, we could also have applied simple parametric optimization instead of shape optimization. Indeed, the problem could have been differentiated with respect to the position of each nodal point. In this second approach, the path is discretized and only then, the derivative is computed [15].

6.3 SHAPE DERIVATIVE COMPUTATION

In this section, we compute the derivative of the functions involved in the optimization problem. Then, their transformation into numerical gradients is detailed. Let's first define the admissible path set $\mathcal{G}$.

Definition 6.1. The space of admissible paths $\mathcal{G}$ is composed by the curves $\Gamma \subset D$ satisfying the Definition 3.1. Their tangent is denoted by τ, their normal by n and their end points by A and B.

6.3.1 Computation in the continuous settings

Using the method of Céa, already presented in Chapter 3 [17, 46, 15], each function is differentiated and presented in the following Proposition 6.1.

Proposition 6.1. *Let $\Gamma \in \mathcal{G}$. Then, L_{F}, C_ϕ and C_{M} (respectively defined by (4.4.10), (4.4.8) and (4.4.9)) are differentiable at Γ, and for any perturbation $\theta \in \mathcal{C}^2\left(\overline{D}, \mathbb{R}^2\right)$, their derivatives are as follows:*

$$DL_{\mathrm{F}}(\Gamma)(\theta) = \int_\Gamma \kappa(s)\theta(s) \cdot n(s) ds + \theta(B) \cdot \tau(B) - \theta(A) \cdot \tau(A), \tag{6.3.1}$$

$$\begin{aligned} DC_\phi(\Gamma)(\theta) = \; & -\int_\Gamma P\left(\partial_n p_\phi(s) + \kappa(s) p_\phi(s)\right)\theta(s)\cdot n(s) ds \\ & -P p_\phi(B)\theta(B)\cdot\tau(B) + P p_\phi(A)\theta(A)\cdot\tau(A), \end{aligned} \tag{6.3.2}$$

$$\begin{aligned} DC_{\mathrm{M}}(\Gamma)(\theta) = \; & -\int_\Gamma P\left(\partial_n p_{\mathrm{M}}(s) + \kappa(s) p_{\mathrm{M}}(s)\right)\theta(s)\cdot n(s) ds \\ & -P p_{\mathrm{M}}(B)\theta(B)\cdot\tau(B) + P p_{\mathrm{M}}(A)\theta(A)\cdot\tau(A), \end{aligned} \tag{6.3.3}$$

with $p_\phi \in H^1(D)$ *solution of*

$$\begin{cases} -\nabla\left(\lambda\nabla p_\phi\right) + \beta p_\phi = 2\left(y_\phi - y\right)^+ & in \quad D \\ \lambda\partial_n p_\phi = 0 & on \quad \partial D \end{cases} \tag{6.3.4}$$

and $p_{\mathrm{M}} \in H^1(D)$ *solution of*

$$\begin{cases} -\nabla\left(\lambda\nabla p_{\mathrm{M}}\right) + \beta p_{\mathrm{M}} = -2\left(y - y_{\mathrm{M}}\right)^+ & in \quad D \\ \lambda\partial_n p_{\mathrm{M}} = 0 & on \quad \partial D \end{cases} \tag{6.3.5}$$

PROOF.
First of all, the derivative of L_{F} is obtained by simply applying Proposition 3.2.

Both the phase and maximum constraints are modeled by the function $C_{\mathrm{X}}(y) = \int_D f(y)dx$, with $f : x \in \mathbb{R} \mapsto f(x) \in \mathbb{R}$ differentiable. For the phase, $\forall x \in D$, $f(y(x)) = \left[\left(y_\phi - y(x)\right)^+\right]^2 \mathbb{1}_{D_S}$ and $f'(y(x)) = -2\left(y_\phi - y(x)\right)^+ \mathbb{1}_{D_S}$. For the maximum constraint, $f(y(x)) = \left[\left(y(x) - y_{\mathrm{M}}\right)^+\right]^2$ and $f'(y(x)) = 2\left(y(x) - y_{\mathrm{M}}\right)^+$.

To compute the shape derivative of this problem, from the method of Céa [17, 46, 15], the Lagrangian function $\mathcal{L} : \mathcal{G} \times H^1(D) \times H^1(D)$ is introduced:

$$\mathcal{L}(\Gamma, \Phi, q) = C_{\mathrm{X}}(\Phi) + \int_D \left(\lambda\Phi\cdot\nabla q + \beta(\Phi - y_{\mathrm{ini}})q\right) dx - \int_\Gamma P q ds.$$

Differentiating this Lagrangian with respect to Φ gives

$$\frac{\partial\mathcal{L}}{\partial\Phi}(\Gamma,\Phi,q)(\psi) = \int_D f'(\Phi)\psi dx + \int_D \left(-\nabla\cdot\left(\lambda\nabla q\right) + \beta q\right)\psi dx - \int_{\partial D}\lambda\partial_n q\psi ds, \qquad \forall\psi \in H^1(D).$$

The adjoint equations (6.3.4) and (6.3.5) are obtained by finding p such that, $\forall\psi \in H^1(D)$,

$$\frac{\partial\mathcal{L}}{\partial\Phi}(\Gamma, y, p)(\psi) = 0.$$

Moreover, $\forall\theta \in \mathcal{C}^2\left(\overline{D}, \mathbb{R}^2\right)$, $\forall q \in H^1(D)$:

$$DC_{\mathrm{X}}(\Gamma)(\theta) = \frac{\partial\mathcal{L}}{\partial\Gamma}(\Gamma, y, p)(\theta) + < \frac{\partial\mathcal{L}}{\partial\Phi}(\Gamma, y, p), \frac{\partial y}{\partial\Gamma}(\theta) > = \frac{\partial\mathcal{L}}{\partial\Gamma}(\Gamma, y, p)(\theta).$$

Thus, differentiating the constraint finally amounts to partially differentiating the Lagrangian with respect to the shape. The only term that must finally be considered is $-\int_\Gamma P p ds$. Since $Pp \in W^{2,1}(\Gamma)$, Proposition 3.2 can be applied (function J_2). □

6.3.2 Regularization

To apply first order optimization algorithms, the derivatives must be transformed into update directions. Following Section 3.2.4, a $H^1(\Gamma, \mathbb{R}^2)-$scalar product is chosen, which consists in computing the gradient by solving the system of minimization problems (3.2.19).
Since the path is numerically known by its nodes, the shape gradient must be known at each point. Thus, the regularization problems must be discretized and the variable W be considered as a vector $(W_i)_{i\in[\![1,N_u]\!]}$ known at the path discretization nodal points ($P1-$function, see Section 5.3.5). However, its derivative $\partial_\tau W$ is defined in each segment ($P0-$functions, see Section 5.3).

The left hand side of the regularization problems is given by

$$LHS = \int_\Gamma \left(\frac{\nu_\Gamma^2}{2} \left(\partial_\tau W\right)^2 + \frac{1}{2} W^2 \right) ds \approx \sum_{i=1}^{N_u-1} l_i \left[\frac{\nu_\Gamma^2}{2} \left(\frac{W_{i+1} - W_i}{l_i} \right)^2 + \frac{1}{2} \frac{W_{i+1}^2 + W_i^2}{2} \right]. \tag{6.3.6}$$

The differentiation of LHS with respect to the variable W is written $\partial_W LHS(W) = \tilde{LHS} \cdot W$ with $\overline{LHS} \in \mathcal{M}_{N_u,N_u}(\mathbb{R})$.

The right hand side RHS depends on the function (L_F, C_ϕ, C_M) considered. For the length, one has

$$\begin{cases} RHS_{L_\mathrm{F}}^\tau = -W(B) + W(A) \approx -W_{N_u} + W_1 \\ RHS_{L_\mathrm{F}}^n = \displaystyle\int_\Gamma \kappa\theta(s) \cdot n(s) ds \approx \sum_{i=1}^{N_u-1} l_i \frac{\kappa_i W_i + \kappa_{i+1} W_{i+1}}{2}. \end{cases} \tag{6.3.7}$$

For the constraints, computing the right hand side requires the computation of the adjoint and its gradient at each discretization node of the path. They are obtained using the process given in in Section 5.3.4. For both constraints (written as X with $X \in \{C_\phi, C_\mathrm{M}\}$), one has

$$\begin{cases} RHS_\mathrm{X}^\tau = -P p_\mathrm{X}(B) W(B) + P p_\mathrm{X}(A) W(A) \approx -P p_{\mathrm{X,N_u}} W_{N_u} + P p_{\mathrm{X},1} W_1 \\ RHS_\mathrm{X}^n = -\displaystyle\int_\Gamma P \left(\partial_n p_\mathrm{X} + \kappa p_\mathrm{X}\right) W ds \\ \qquad\qquad \approx \displaystyle\sum_{i=1}^{N_u-1} -l_i P \frac{\left(\partial_n p_{\mathrm{X,i}} + \kappa_i p_{\mathrm{X,i}}\right) W_i + \left(\partial_n p_{\mathrm{X,i+1}} + \kappa_{i+1} p_{\mathrm{X,i+1}}\right) W_{i+1}}{2} \end{cases} \tag{6.3.8}$$

The differentiation of each right hand side RHS with respect to W, $\partial_W RHS \in \mathbb{R}^{N_u}$.

Note that these discretizations have been obtained by a $P1-$quadrature rule: the multiplication of two $P1-$functions on the path is considered to be a $P1-$function defined by its nodal values resulting from "node by node" multiplication of these $P1-$functions at each node (see Remark 5.7).

The final discrete gradients are $L'_\mathrm{F} = (L'_\mathrm{F})_\tau \, \tau + (L'_\mathrm{F})_n \, n$, $C'_\phi = \left(C'_\phi\right)_\tau \tau + \left(C'_\phi\right)_n n$, $C'_\mathrm{M} = (C'_\mathrm{M})_\tau \, \tau + (C'_\mathrm{M})_n \, n$, with

$$(L'_\mathrm{F})_\tau = \overline{LHS}^{-1} \cdot \partial_W RHS_{L_\mathrm{F}}^\tau \in \mathbb{R}^{N_u}, \quad (L'_\mathrm{F})_n = \overline{LHS}^{-1} \cdot \partial_W RHS_{L_\mathrm{F}}^n \in \mathbb{R}^{N_u},$$

$$\left(C'_\phi\right)_\tau = \overline{LHS}^{-1} \cdot \partial_W RHS_{C_\phi}^\tau \in \mathbb{R}^{N_u}, \quad \left(C'_\phi\right)_n = \overline{LHS}^{-1} \cdot \partial_W RHS_{C_\phi}^n \in \mathbb{R}^{N_u},$$

$$(C'_\mathrm{M})_\tau = \overline{LHS}^{-1} \cdot \partial_W RHS_{C_\mathrm{M}}^\tau \in \mathbb{R}^{N_u}, \quad (C'_\mathrm{M})_n = \overline{LHS}^{-1} \cdot \partial_W RHS_{C_\mathrm{M}}^n \in \mathbb{R}^{N_u}.$$

6.4 NUMERICAL ALGORITHMS

Since we have obtained the gradients of the cost and constraints, we can apply the first-order algorithms for constrained optimization presented in Section 2.4, adapted to the steady state problem.

6.4.1 Numerical Augmented Lagrangian algorithm

The Augmented Lagrangian method, presented in Section 2.4, is first used.

Augmented Lagrangian method and corresponding derivatives

In the steady state context, we notice that the phase and maximum temperature constraints are very similar and can easily be compared. In order to simplify the optimization process, they are aggregated in one unique constraint

$$C = C_\phi + C_{\mathrm{M}} \quad \left(= C_\phi + C_{\mathrm{M,D_S}} + C_{\mathrm{M,D}\setminus\mathrm{D_S}}\right). \tag{6.4.1}$$

The penalized problem consists in

$$\min_{\Gamma\in\mathcal{G}} f_{\mathrm{ALM}}(\Gamma) = L_{\mathrm{F}} + \frac{c}{2}C^2 \quad \text{such that} \quad C = 0, \quad \text{and} \quad (6.2.2)\,\text{holds}, \tag{6.4.2}$$

with $c > 0$ the penalization coefficient (Section 2.4.4). The projected Augmented Lagrangian Algorithm 2.5 can be used. The condition $\Gamma \subset D$ is addressed with a projection and the (normalized) Lagrangian function is

$$\mathcal{L}_{\mathrm{ALM}}(\Gamma, l_C) = \frac{L_{\mathrm{F}}}{L_{\mathrm{F}}^0} + l_C\frac{C}{C^0} + \frac{c}{2}\left(\frac{C}{C^0}\right)^2. \tag{6.4.3}$$

Numerical descent direction

In Algorithm 2.5, the Lagrange multiplier and the path are updated following

$$\begin{cases} l_C^{k+1} = l_C^k + cC^{k+1}, \\ \Gamma^{k+1} = \Gamma^k - s_\Gamma^k \mathcal{L}'_{\mathrm{ALM}}\left(\Gamma^k, l_C^k\right), \end{cases} \tag{6.4.4}$$

with the numerical gradient $\mathcal{L}'_{\mathrm{ALM}}$ given by

$$\mathcal{L}'_{\mathrm{ALM}}(\Gamma, l_C) = \frac{1}{L_{\mathrm{F}}^0}L'_{\mathrm{F}} + \frac{1}{C^0}\left(l_C + c\frac{C}{C^0}\right)\left(C'_\phi + C'_{\mathrm{M}}\right), \tag{6.4.5}$$

with L'_{F}, C'_ϕ, C'_{M} defined in Section 6.3.2.

Step and projection

For each iteration $k > 0$, the advection velocity $d_\Gamma^k = \left(d_{\Gamma,i}^k\right)_{i\in[\![1,N_u]\!]} = -\mathcal{L}'_{\mathrm{ALM}}(\Gamma^k, l_C^k)$ is determined at each point and the path is updated by:

$$\forall i \in [\![1, N_u]\!], \qquad u_i^{k+1} = \left(x_i^{k+1}, y_i^{k+1}\right) = u_i^k + s_\Gamma^k d_{\Gamma,i}^k.$$

The step s_Γ^k is given by:

$$s_\Gamma^k = \frac{C_s^k \Delta x}{\max_i(\|d_{\Gamma,i}^k\|)},$$

with C_s^k a coefficient initialized to 1 and updated at each iteration so that, if the objective function decreases enough, namely $\mathcal{L}_{\mathrm{ALM}}(\Gamma^{k+1}) < tol^k * \mathcal{L}_{\mathrm{ALM}}(\Gamma^k)$, $C_s^{k+1} = \min\left(1.2C_s^k, 1\right)$ and else $C_s^{k+1} = 0.6C_s^k$. This corresponds to the line search strategy given in Chapter 2. The tolerance is set to $tol^0 = 2$ at the beginning and is multiplied by 0.9 every 50 iterations. The coefficients 1.2 and 0.6 and the tolerance values are arbitrarily fixed following experimental tests.

After this update, if some points are outside from the domain D they are orthogonally projected back to D. The algorithm is thus a projected Augmented Lagrangian.

Remark 6.2. As stated in section 2.4.3, the convergence of this algorithm has not been proved. However, the descent properties are being controlled through the tolerance process.

Algorithm

The numerical process for solving problem (6.2.1) is presented in Algorithm 6.1. In this algorithm, the iterations are stopped if the step coefficient C_s is smaller that 10^{-8}. An additional maximum number of

iteration N_{MAX} is also considered in the stopping criterion ($N_{\text{MAX}} = 500$).

$k = 0$ and initialization of the path Γ^0 and multiplier l_C^0
resolution of the heat equation and computation of the objective function and constraints
computation of the derivatives
while $C_s^k \geq 10^{-8}$ *and* $k \leq N_{\text{MAX}}$ **do**
    update of the tolerance
    path variation such that $\Gamma^{k+1} = \Gamma^k - s_\Gamma^k \mathcal{L}'_{\text{ALM}}(\Gamma^k, l_C^k)$
    re-discretization of the path Γ^{k+1}
    resolution of the heat equation, computation of the objective function and constraint
    if $\mathcal{L}_{\text{ALM}}(\Gamma^{k+1}) < \mathcal{L}_{\text{ALM}}(\Gamma^k) * tol^k$ **then**
        iteration accepted
        Lagrange multiplier l_C updated: $l_C^{k+1} = l_C^k + cC^{k+1}$
        step increased
        update of the variables
        computation of the derivatives
    end
    else
        iteration rejected
        step decreased
    end
end

Algorithm 6.1: Iterative algorithm to optimize the steady state problem.

6.4.2 Numerical null space gradient flow algorithm

The null space gradient flow algorithm is also tested. In order to compute the directions $\xi_{L_{\text{F}}}$ and ξ_C, the derivative with respect to each function and a scalar product are required. The derivatives are given by Proposition 6.1 and their discretization and scalar product by the regularization process. Then, the code provided by [82] is used with the time step Δt (related to the path discretization) fixed to d_{l} (minimum distance between two nodal points in the path discretization) with a slight modification: after each advection of the path nodes, they are projected within the working domain.

6.5 Numerical results

As mentioned in Chapter 5, we consider a working domain $D = [-0.7, 0.7] \times [-0.7, 0.7]$ (given in mm). The accuracy of the discretized path is given by the distance Δu between two nodes ($d_{\text{l}} = 0.35\Delta x \leq \Delta u \leq 0.7\Delta x = d_{\text{u}}$).

For each material, the physical properties (conductivity and change of phase temperature) and the process parameters chosen (power and initial temperature) are given in the Nomenclature. The absorption coefficient and effective power correspond to the results from the calibration tests (4.4.17). The maximum temperature in the domain D_S is $y_{\text{M,D}_\text{S}} = 1670K$ for the aluminium and $y_{\text{M,D}_\text{S}} = 3400K$ for the titanium. Out of the domain, to make sure that the temperature does not overcome the change of phase temperature y_ϕ, we choose $y_{\text{M,D}\setminus\text{D}_\text{S}} = y_{\text{M,D}_\text{S}} - 100$. However, applying this rule for the aluminium makes the maximum temperature out of the domain smaller than the initial temperature. We finally set $y_{\text{M,D}\setminus\text{D}_\text{S}} = y_\phi = 870K$ for the aluminium and $y_{\text{M,D}\setminus\text{D}_\text{S}} = y_\phi - 100 = 1800K$ for the titanium. This choice is arbitrary and the temperature could also be chosen smaller.
The finite element computations are performed with Freefem 3.56 [96] whereas the descent algorithm is coded in Python 3.6.0. This problem is run on a MacBook laptop equipped with 2,3 GHz Intel Core i5 and a RAM of 16GB. No specific efforts for optimizing the Python code have been made but for a Cholesky decomposition of the stiffness matrix involved in the heat and adjoint equations. The computational time per iteration is provided for two tests in Figures 6.3 and 6.5.

After a brief illustration of the optimization process, the numerical choices such as the regularization parameter, the path discretization or the optimization algorithms are studied. Then, the physical parameters are analyzed, to characterize the impact of each coefficient on the optimal path. Finally, some tests are conducted to highlight how the projection process modifies the results.

In all these first tests, the building domain is a square object, homothety of the working domain with coefficient 0.9 (see Figure 6.1).

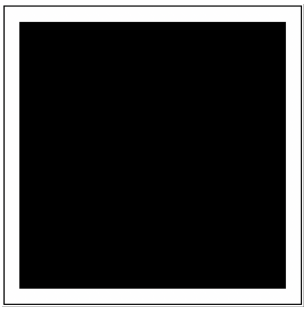


Figure 6.1: Square object to build (in black) in the working domain D

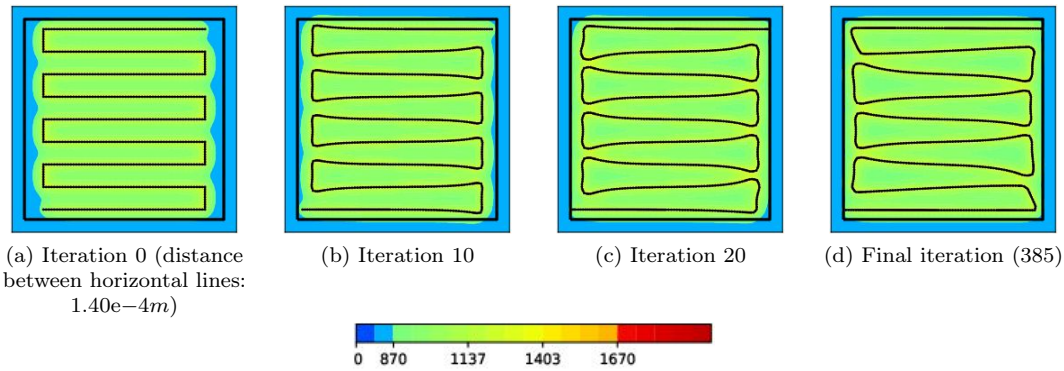
In the following, the numerical results are presented in Figures, showing the temperature map and the path, and in Tables. The phase and maximum temperature constraints are adimensionalized in these Tables as follows:

$$\overline{C}_{\phi} = \frac{C_{\phi}}{|D_S| y_{\phi}^2}, \quad \overline{C}_{\mathrm{M,D_S}} = \frac{C_{\mathrm{M,D_S}}}{|D_S| y_{\mathrm{M,D_S}}^2}, \quad \overline{C}_{\mathrm{M,D\setminus D_S}} = \frac{C_{\mathrm{M,D\setminus D_S}}}{|D \setminus D_S| y_{\mathrm{M,D\setminus D_S}}^2}. \tag{6.5.1}$$

6.5.1 First results

To illustrate the optimization process, two preliminary cases are run, both using an Augmented Lagrangian method with a fixed penalization multiplier $c = 10$ and the Lagrange multiplier initialized to 1 (this choice is arbitrary and different algorithms are tested in the following). The regularization coefficient is fixed to $\nu_\Gamma = 15 d_\mathrm{l}$ and the part to build is shown in Figure 6.1. The objective of these results is to illustrate, in the aluminium and titanium cases, the effect of the optimization process. Both zigzag initializations are chosen to be well adapted to the shape to build D_S. The distance between two horizontal lines is $1.40\mathrm{e}{-4} m$ for the aluminium and $1.05\mathrm{e}{-4} m$ for the titanium.

Consider first aluminium powder. The path shape at different iterations is shown in Figure 6.2. The evolution of the path length and the constraints is given in Figure 6.3. The maximum constraint is split into two different plots: the first one represents the over temperature within the part D_S and the second one out of the part. Finally, the computational time per iteration is provided. In all the Figures, the color bar indicates the temperature values. To satisfy the constraints, the temperature must be in between $[870K, 1670K]$, (green to yellow colors). The overheated areas are represented in red whereas underheated ones are represented in blue. Out of the object, the temperature must remain under $870K$ and thus blue. These conditions appear to be quite well fulfilled in Figure 6.2. Indeed, if the path is not covering the square at the initial iteration, it then expands while managing to keep the heat within the domain. Regarding the objective function and constraint evolution, they confirm this trend. The length increases whereas the phase constraint decreases: if the objective function is worsened, the phase constraint is better satisfied. The maximum temperature within the domain D_S is not reached and does not influence the results. As for the maximum temperature out of the domain, it slightly increases. Note that the initialization is especially well adapted in this example. The final results hint that zigzag paths could be improved by slightly tilting the horizontal segments. This result is thus very promising since it corroborates results obtained in the literature [77].



(a) Iteration 0 (distance between horizontal lines: 1.40e−4m)

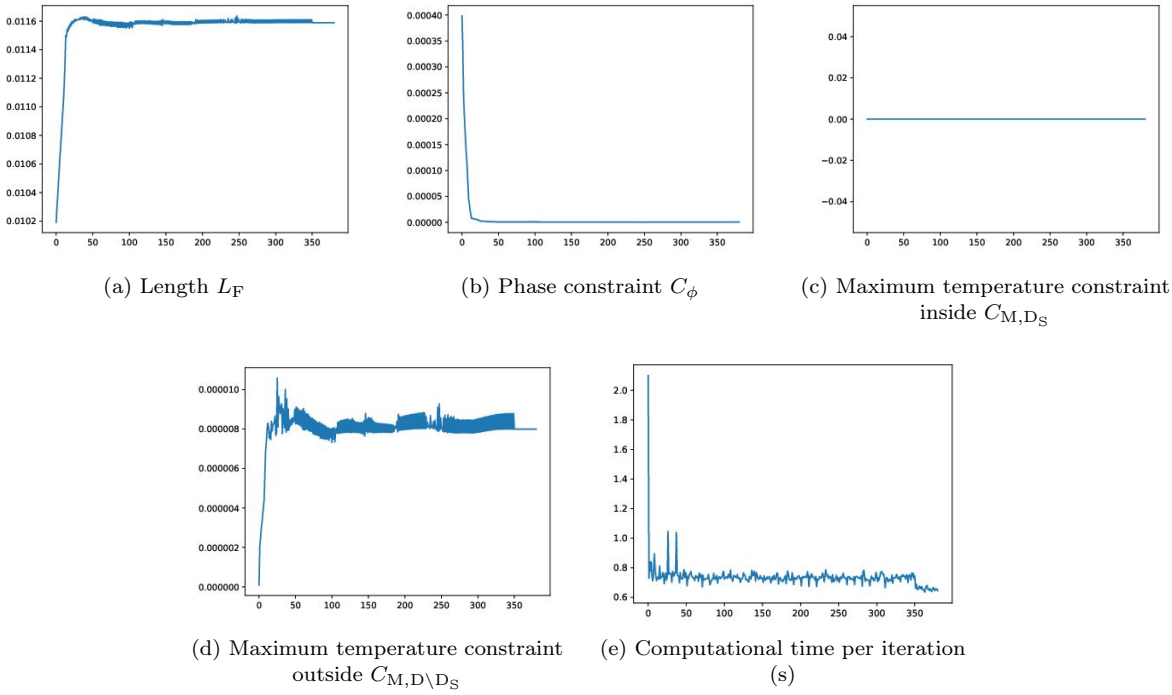
(b) Iteration 10

(c) Iteration 20

(d) Final iteration (385)

Temperature color bar (K)

Figure 6.2: Evolution of the path with respect to the iterations and corresponding temperature (aluminium).



(a) Length L_{F}

(b) Phase constraint C_{ϕ}

(c) Maximum temperature constraint inside $C_{\mathrm{M,D_S}}$

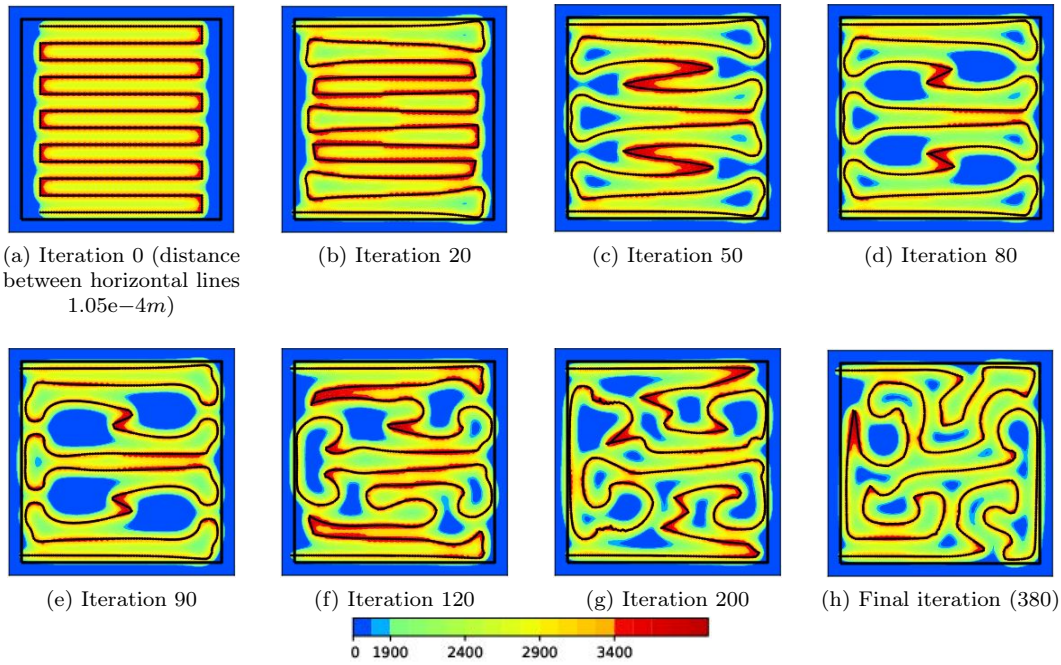
(d) Maximum temperature constraint outside $C_{\mathrm{M,D\backslash D_S}}$

(e) Computational time per iteration (s)

Figure 6.3: Evolution of the different values with respect to the iterations (aluminium).

Consider the titanium powder, the conductivity of which is far lower than the aluminium ($15Wm^{-1}K^{-1}$ for the titanium and $130Wm^{-1}K^{-1}$ for the aluminium). The path shape during iterations is shown in Figure 6.4. The evolution of the path length and the constraints are given in Figure 6.5. To satisfy the constraints, the temperature must be in between $[1900K, 3400K]$ (green to yellow colors). Out of the object, the temperature must remain under $1800K$ (deep blue). These conditions are not satisfied at the initial iteration. Indeed, the path seems to be too long and not well spread since it does not cover the left and right sides. The first iterations focus on the second point: an increase of the length to cover the whole domain D_S results in a decrease in the phase constraint (zone 1 on Figure 6.5 and from (a) to (b) on Figure 6.4). In a second step, the length is reduced inducing sharp angles and an increase of the maximal temperature constraint (zone 2 on Figure 6.5 and from (b) to (c) on Figure 6.4). The angles are then smoothed and the length still reduced until a second increase to reconsider the phase constraint(zone 3 on Figure 6.5 and from (c) to (d) on Figure 6.4). The length is then increased again (zones 4 and 5 on Figure 6.5 and from (d) to (f) on Figure 6.4) leading to a decrease of the phase constraint (zone 5 on Figure 6.5 and from (e) to (f) on Figure 6.4). The final path does not fulfill the constraints. Indeed, the

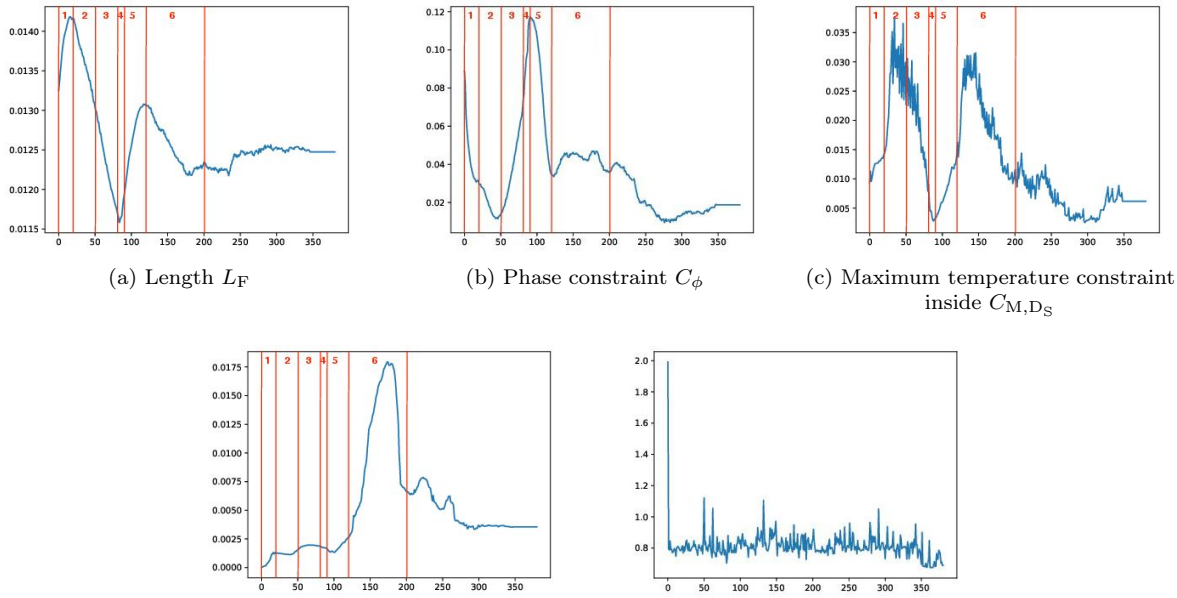
temperature remains too high at some points within the domain (sharp angles remaining) and outside. Moreover, there are still some holes where the powder has not solidified. However, the path has adapted and we recall that, since the problem lacks convexity, we cannot ensure the convergence of the algorithm but only control the constraints and cost decreases.



(a) Iteration 0 (distance between horizontal lines 1.05e−4m)
(b) Iteration 20
(c) Iteration 50
(d) Iteration 80
(e) Iteration 90
(f) Iteration 120
(g) Iteration 200
(h) Final iteration (380)

Temperature color bar (K)

Figure 6.4: Evolution of the path with respect to the iterations and corresponding temperature (titanium).



(a) Length L_{F}
(b) Phase constraint C_{ϕ}
(c) Maximum temperature constraint inside $C_{\mathrm{M,D_S}}$
(d) Maximum temperature constraint outside $C_{\mathrm{M,D\setminus D_S}}$
(e) Computational time per iteration (s)

Figure 6.5: Evolution of the different values with respect to the iterations (titanium). The red lines correspond to iterations 1, 20, 50, 80, 90, 120, 200 (see Figure 6.4)

These first tests appear promising to optimize the path in the steady state context and tend to confirm the intuition that a bounded curvature is an important feature of a good path, to prevent localized high temperatures.

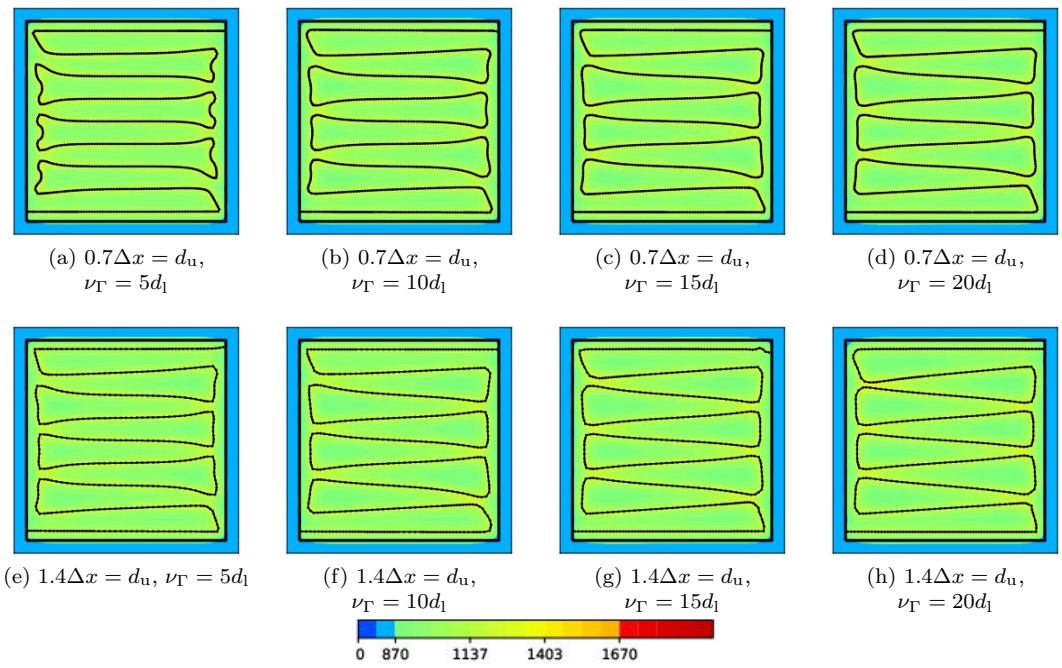
6.5.2 Impact of the numerical choices

Along the settings of this path optimization, many choices on the model and its numerical discretization have been made. Some of them are now tested to evaluate their influence on the results.

Impact of the path discretization and of the regularization coefficient

Among the several parameters influencing the process, we first focus on those determining the freedom left to the algorithm to design small patterns: the path discretization which is directly related to the detail characteristic size that the path can achieve, and the regularization coefficient ν_Γ that fixes the dependence between the evolution of a point with respect to the evolution of its neighbors. Two characteristic sizes have been tested: the regular one, with $d_\mathrm{l} = 0.35\Delta x \leq \Delta u \leq 0.7\Delta x = d_\mathrm{u}$ and a coarser one, $d_\mathrm{l} = 0.7\Delta x \leq \Delta u \leq 1.4\Delta x = d_\mathrm{u}$. Concerning the regularization, the choices are $\nu_\Gamma/d_\mathrm{l} \in \{5, 10, 15, 20\}$. The eight tests have been realized with an Augmented Lagrangian method with $c = 10$ and $l_C^0 = 1$.

The results in the aluminium case are given in Figure 6.6, the initialization corresponds to the previous one (zigzag with 9 horizontal lines, see Figure 6.2(a)). Table 6.1 compares the final values in each case. For the titanium case, the initialization corresponds to a zigzag with 12 horizontal lines (see Figure 6.4(a)) and the final paths are displayed in Figure 6.7 and Table 6.2.



(a) $0.7\Delta x = d_\mathrm{u}$, $\nu_\Gamma = 5d_\mathrm{l}$ (b) $0.7\Delta x = d_\mathrm{u}$, $\nu_\Gamma = 10d_\mathrm{l}$ (c) $0.7\Delta x = d_\mathrm{u}$, $\nu_\Gamma = 15d_\mathrm{l}$ (d) $0.7\Delta x = d_\mathrm{u}$, $\nu_\Gamma = 20d_\mathrm{l}$

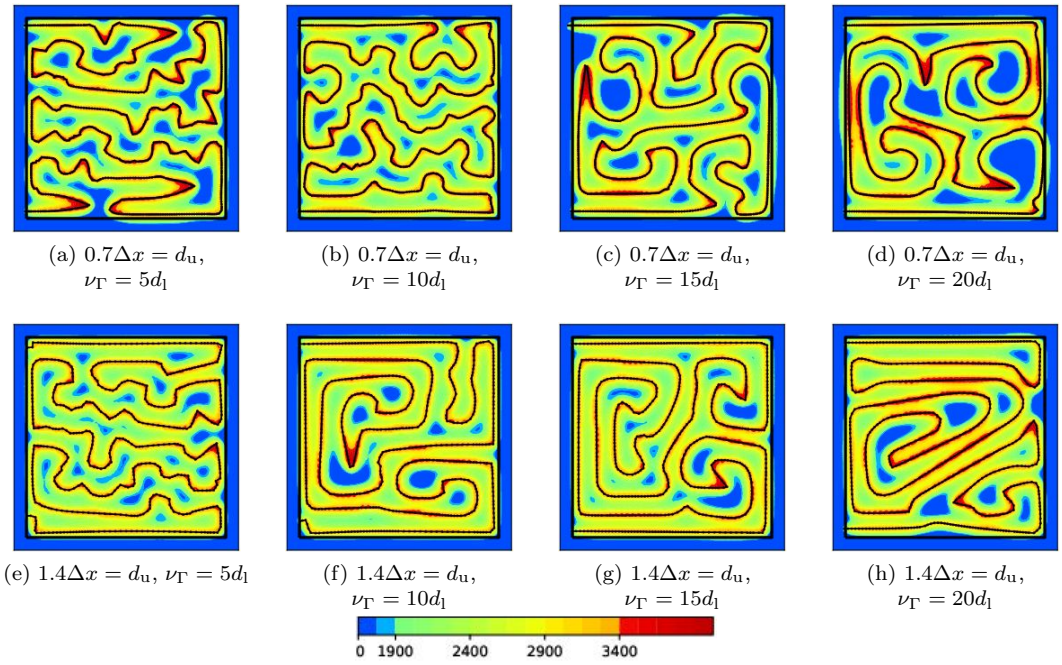
(e) $1.4\Delta x = d_\mathrm{u}$, $\nu_\Gamma = 5d_\mathrm{l}$ (f) $1.4\Delta x = d_\mathrm{u}$, $\nu_\Gamma = 10d_\mathrm{l}$ (g) $1.4\Delta x = d_\mathrm{u}$, $\nu_\Gamma = 15d_\mathrm{l}$ (h) $1.4\Delta x = d_\mathrm{u}$, $\nu_\Gamma = 20d_\mathrm{l}$

Temperature color bar (K)

Figure 6.6: Final temperature and path depending on the discretization accuracy and on the chosen regularization (aluminium).

Case	Length (m)	$\overline{C}_{\phi}$	$\overline{C}_{\mathrm{M,D_S}}$	$\overline{C}_{\mathrm{M,D\setminus D_S}}$
Initialization (Figure 6.3(a))	1.020e−2	3.31e−4	0.00	3.09e−7
$0.7\Delta x = d_{\mathrm{u}}$, $\nu_{\Gamma} = 5d_{\mathrm{l}}$	1.170e−2	3.15e−7	0.00	1.97e−5
$0.7\Delta x = d_{\mathrm{u}}$, $\nu_{\Gamma} = 10d_{\mathrm{l}}$	1.160e−2	4.05e−7	0.00	2.69e−5
$0.7\Delta x = d_{\mathrm{u}}$, $\nu_{\Gamma} = 15d_{\mathrm{l}}$	1.160e−2	4.19e−7	0.00	2.83e−5
$0.7\Delta x = d_{\mathrm{u}}$, $\nu_{\Gamma} = 20d_{\mathrm{l}}$	1.160e−2	5.52e−7	0.00	2.87e−5
$1.4\Delta x = d_{\mathrm{u}}$, $\nu_{\Gamma} = 5d_{\mathrm{l}}$	1.160e−2	3.20e−7	0.00	3.13e−5
$1.4\Delta x = d_{\mathrm{u}}$, $\nu_{\Gamma} = 10d_{\mathrm{l}}$	1.160e−2	4.52e−7	0.00	3.20e−5
$1.4\Delta x = d_{\mathrm{u}}$, $\nu_{\Gamma} = 15d_{\mathrm{l}}$	1.160e−2	4.93e−7	0.00	2.97e−5
$1.4\Delta x = d_{\mathrm{u}}$, $\nu_{\Gamma} = 20d_{\mathrm{l}}$	1.160e−2	7.88e−7	0.00	3.44e−5

Table 6.1: Comparison of the cost and the constraints of the final results (aluminium).



(a) $0.7\Delta x = d_{\mathrm{u}}$, $\nu_{\Gamma} = 5d_{\mathrm{l}}$ (b) $0.7\Delta x = d_{\mathrm{u}}$, $\nu_{\Gamma} = 10d_{\mathrm{l}}$ (c) $0.7\Delta x = d_{\mathrm{u}}$, $\nu_{\Gamma} = 15d_{\mathrm{l}}$ (d) $0.7\Delta x = d_{\mathrm{u}}$, $\nu_{\Gamma} = 20d_{\mathrm{l}}$

(e) $1.4\Delta x = d_{\mathrm{u}}$, $\nu_{\Gamma} = 5d_{\mathrm{l}}$ (f) $1.4\Delta x = d_{\mathrm{u}}$, $\nu_{\Gamma} = 10d_{\mathrm{l}}$ (g) $1.4\Delta x = d_{\mathrm{u}}$, $\nu_{\Gamma} = 15d_{\mathrm{l}}$ (h) $1.4\Delta x = d_{\mathrm{u}}$, $\nu_{\Gamma} = 20d_{\mathrm{l}}$

Temperature color bar (K)

Figure 6.7: Final temperature and path depending on the discretization accuracy and on the chosen regularization (titanium).

Case	Length (m)	$\overline{C}_\phi$	$\overline{C}_{\mathrm{M,D_S}}$	$\overline{C}_{\mathrm{M,D\setminus D_S}}$
Initialization (Figure 6.5(a))	1.330e−2	1.55e−2	5.11e−4	0.00
$0.7\Delta x = d_\mathrm{u}$, $\nu_\Gamma = 5d_\mathrm{l}$	1.240e−2	1.62e−3	2.11e−4	1.78e−3
$0.7\Delta x = d_\mathrm{u}$, $\nu_\Gamma = 10d_\mathrm{l}$	1.250e−2	5.04e−4	3.36e−5	1.90e−4
$0.7\Delta x = d_\mathrm{u}$, $\nu_\Gamma = 15d_\mathrm{l}$	1.250e−2	3.26e−3	3.37e−4	2.95e−3
$0.7\Delta x = d_\mathrm{u}$, $\nu_\Gamma = 20d_\mathrm{l}$	1.270e−2	6.37e−3	4.82e−4	5.07e−3
$1.4\Delta x = d_\mathrm{u}$, $\nu_\Gamma = 5d_\mathrm{l}$	1.260e−2	3.32e−4	2.09e−5	3.08e−4
$1.4\Delta x = d_\mathrm{u}$, $\nu_\Gamma = 10d_\mathrm{l}$	1.330e−2	7.47e−4	8.39e−5	6.17e−4
$1.4\Delta x = d_\mathrm{u}$, $\nu_\Gamma = 15d_\mathrm{l}$	1.320e−2	4.80e−4	5.36e−5	6.78e−4
$1.4\Delta x = d_\mathrm{u}$, $\nu_\Gamma = 20d_\mathrm{l}$	1.330e−2	1.80e−3	1.72e−4	7.08e−4

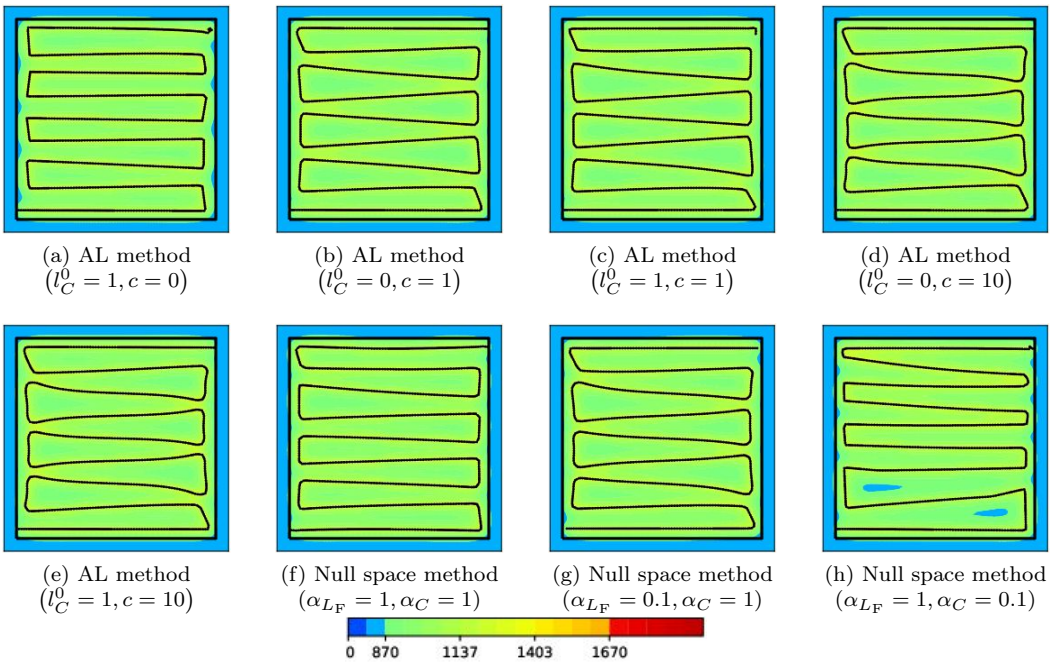
Table 6.2: Comparison of the cost and the constraints of the final results (titanium)

The conductivity being high in the aluminium case, the results are not extremely different from one test to another. Yet, they endorse the regularization effect of the H^1 scalar product. Indeed, small coefficients ν_Γ authorize the points to move in different directions than their neighbors. With the increase of ν_Γ, these directions get smoother to end up with Figure 6.6(d) which almost consists of straight lines. Increasing the discretization coefficient has also a regularizing effect: from one line to another, the slight variations are erased.

The impact of the regularization coefficient ν_Γ and the discretization is on the contrary very clear for the titanium. The conductivity being smaller, the number of local minima increases and any difference in the optimization parameters leads to drastically different scanning path shapes. These results confirm the previous observations. From left to right, the path gets smoother, as well as from one line to another. In addition to the previous analysis, one can notice on these new results that some configurations better fulfill the constraints than others: a small regularization allows for a significant freedom to the path however not smooth, and a high regularization decreases the freedom in the design to the advantage of smoothness.
In the following, a compromise must be chosen between freedom and smoothness. On the one hand, freedom enlarges the number of admissible path and thus facilitates the fulfillment of the constraints. On the other hand, in industrial applications, the source is better controlled in case of smooth patterns. Indeed, to achieve sharp corners, the source must slow down requiring a decrease in its power which is not available in all technologies. Time may not be considered in this steady state case but this issue can still be taken into account by demanding a relative path regularity. The parameters in the remaining of this work are finally $d_\mathrm{l} = 0.35\Delta x \leq \Delta u \leq 0.7\Delta x = d_\mathrm{u}$ and $\nu_\Gamma = 15d_\mathrm{l}$, values already in use in the first tests (Figures 6.3 and 6.5).

Impact of the optimization algorithm

The second choice to consider is the optimization algorithm. In the Lagrangian method, the initial multiplier l_C and the penalization coefficient c must be set. Five tests have been run: $\left(l_C^0 = 1, c = 0\right)$ (fixed Lagrange multiplier), $\left(l_C^0 = 0, c = 1\right)$, $\left(l_C^0 = 1, c = 1\right)$, $\left(l_C^0 = 0, c = 10\right)$, $\left(l_C^0 = 1, c = 10\right)$. Finally the second algorithm with null flows gradient has been run with $(\alpha_{L_\mathrm{F}}, \alpha_C) = (1, 1)$, $(\alpha_{L_\mathrm{F}}, \alpha_C) = (1, 0.1)$, $(\alpha_{L_\mathrm{F}}, \alpha_C) = (0.1, 1)$. The results in the aluminium case initialized with a zigzag with 9 lines (Figure 6.2(a)) are given in Figure 6.8 and Table 6.3 whereas the results in the titanium case initialized with a zigzag with 12 lines (Figure 6.4(a)) are given in Figure 6.9 and Table 6.4.



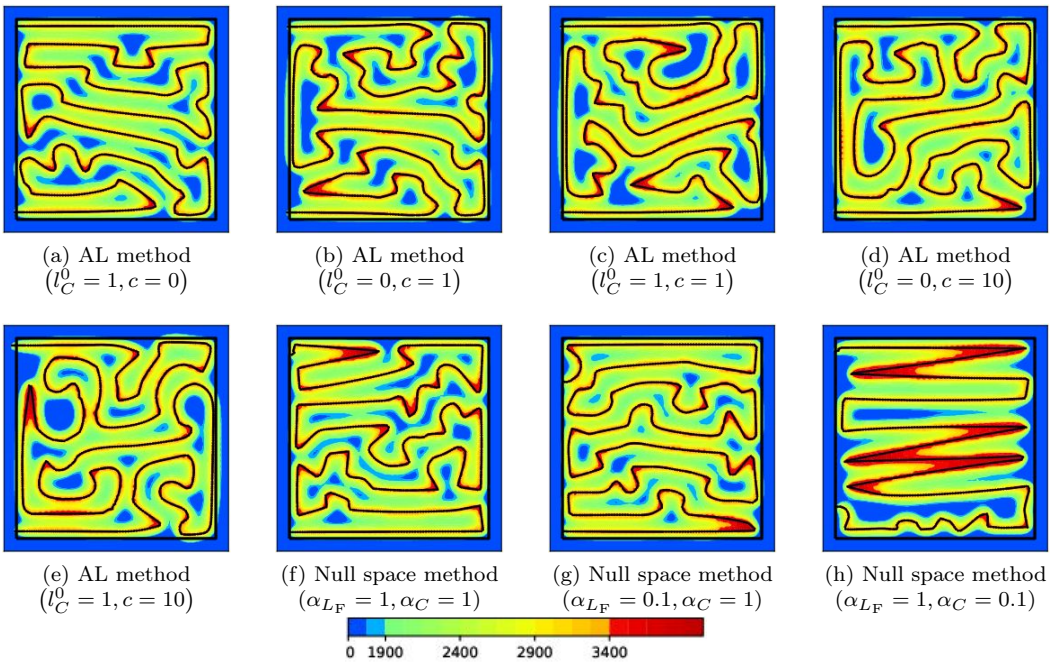
(a) AL method ($l_C^0 = 1, c = 0$)
(b) AL method ($l_C^0 = 0, c = 1$)
(c) AL method ($l_C^0 = 1, c = 1$)
(d) AL method ($l_C^0 = 0, c = 10$)
(e) AL method ($l_C^0 = 1, c = 10$)
(f) Null space method ($\alpha_{L_F} = 1, \alpha_C = 1$)
(g) Null space method ($\alpha_{L_F} = 0.1, \alpha_C = 1$)
(h) Null space method ($\alpha_{L_F} = 1, \alpha_C = 0.1$)

Temperature color bar (K)

Figure 6.8: Final temperature and path depending on the algorithm (aluminium).

Case	Length (m)	$\overline{C}_\phi$	$\overline{C}_{\mathrm{M,D_S}}$	$\overline{C}_{\mathrm{M,D\backslash D_S}}$
Initialization (Figure 6.3(a))	1.019e−2	3.31e−4	0.00	3.09e−7
Aug. Lagrangian $\left(l_C^0 = 1, c = 0\right)$	1.119e−2	7.18e−6	0.00	2.38e−5
Aug. Lagrangian $\left(l_C^0 = 0, c = 1\right)$	1.150e−2	5.61e−7	0.00	2.89e−5
Aug. Lagrangian $\left(l_C^0 = 1, c = 1\right)$	1.152e−2	5.11e−7	0.00	2.76e−5
Aug. Lagrangian $\left(l_C^0 = 0, c = 10\right)$	1.159e−2	5.07e−7	0.00	2.72e−5
Aug. Lagrangian $\left(l_C^0 = 1, c = 10\right)$	1.159e−2	4.20e−7	0.00	2.84e−5
Null space method ($\alpha_{L_F} = 1, \alpha_C = 1$)	1.130e−2	2.73e−6	0.00	6.42e−6
Null space method ($\alpha_{L_F} = 0.1, \alpha_C = 1$)	1.150e−2	7.31e−7	0.00	6.81e−6
Null space method ($\alpha_{L_F} = 1, \alpha_C = 0.1$)	1.130e−2	2.65e−6	0.00	1.67e−5

Table 6.3: Comparison of the cost and the constraints of the final results (aluminium).



(a) AL method $(l_C^0 = 1, c = 0)$ (b) AL method $(l_C^0 = 0, c = 1)$ (c) AL method $(l_C^0 = 1, c = 1)$ (d) AL method $(l_C^0 = 0, c = 10)$

(e) AL method $(l_C^0 = 1, c = 10)$ (f) Null space method $(\alpha_{L_F} = 1, \alpha_C = 1)$ (g) Null space method $(\alpha_{L_F} = 0.1, \alpha_C = 1)$ (h) Null space method $(\alpha_{L_F} = 1, \alpha_C = 0.1)$

Temperature colorbar (K)

Figure 6.9: Final temperature and path depending on the algorithm (titanium).

Case	Length (m)	$\overline{C}_\phi$	$\overline{C}_{\mathrm{M,D_S}}$	$\overline{C}_{\mathrm{M,D\backslash D_S}}$
Initialization (Figure 6.5(a))	1.325e−2	1.55e−2	5.11e−4	0.00
Aug. Lagrangian $\left(l_C^0 = 1, c = 0\right)$	1.177e−2	1.51e−3	2.33e−5	8.79e−5
Aug. Lagrangian $\left(l_C^0 = 0, c = 1\right)$	1.245e−2	1.58e−3	1.44e−4	1.98e−3
Aug. Lagrangian $\left(l_C^0 = 1, c = 1\right)$	1.274e−2	2.45e−3	2.69e−4	2.64e−3
Aug. Lagrangian $\left(l_C^0 = 0, c = 10\right)$	1.269e−2	1.23e−3	1.49e−4	7.35e−4
Aug. Lagrangian $\left(l_C^0 = 1, c = 10\right)$	1.247e−2	3.26e−3	3.37e−4	2.95e−3
Null space method $(\alpha_{L_F} = 1, \alpha_C = 1)$	1.220e−2	6.47e−4	3.51e−4	1.60e−4
Null space method $(\alpha_{L_F} = 0.1, \alpha_C = 1)$	1.280e−2	4.33e−4	1.62e−4	1.38e−4
Null space method $(\alpha_{L_F} = 1, \alpha_C = 0.1)$	1.230e−2	5.85e−3	5.04e−3	3.50e−6

Table 6.4: Comparison of the cost and the constraints of the final results (titanium).

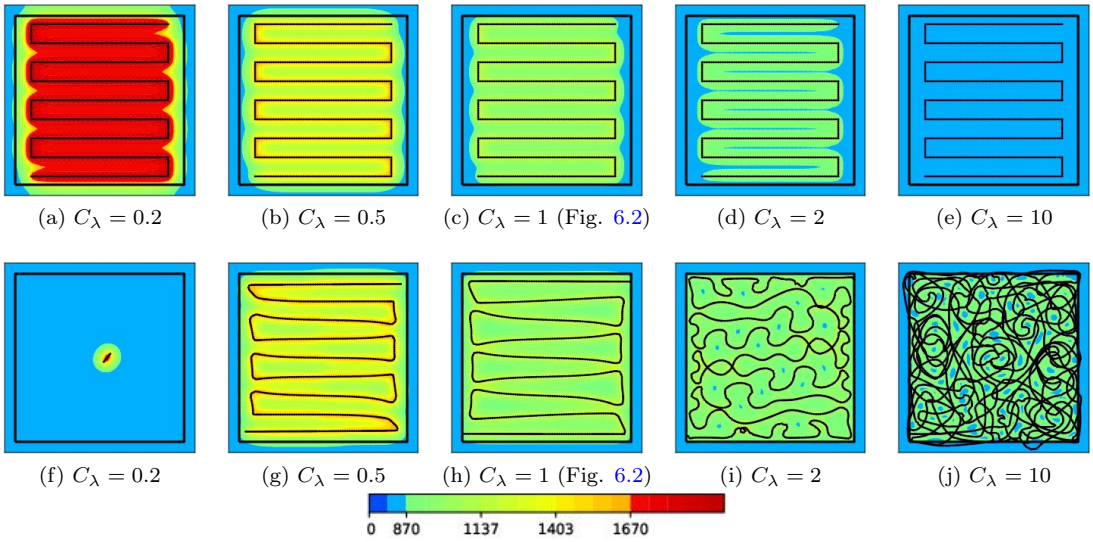
First of all, these tests demonstrate the existence of many different local minima. The dependence of the result with respect to the optimization parameters is significant. From one choice to another, the path can be completely different. This can be clearly observed in the titanium case which confirms that the optimization is significantly more complicated for lower conduction cases.
In the Augmented Lagrangian approach, if the final patterns are different, the quantitative results are similar (see Tables 6.3 and 6.4). Note that the choice $l_C = 1$, $c = 0$ is called "Augmented Lagrangian" but is actually an unconstrained problem. Indeed, the objective function has just been transformed to be the sum of the length and the constraint, without an effective penalization process. With the null space gradient algorithm, the impact of the coefficients α_{L_F} and α_C is well illustrated. The test $(\alpha_{L_F} = 1, \alpha_C = 0.1)$ (Figures 6.8 (f) and 6.9 (f)), focusing more on decreasing the length than satisfying the constraints, result in both material cases to final phase and maximal temperature in D_S higher than in the other tests. On the contrary, at least for the titanium, the choice $(\alpha_{L_F} = 0.1, \alpha_C = 1)$ satisfy them better. In the following, the Augmented Lagrangian method with $\left(l_C^0 = 1, c = 10\right)$ is used.

6.5.3 Impact of the physical model

The physical model results from several assumptions. Even if it has been calibrated, it is still relevant to understand the dependence of the results on the conductivity, the absorption, the source power as well as the maximal temperature. For each of them, the aluminium case and the titanium case are tested respectively initialized with a zigzag with 9 lines and a zigzag with 12 lines (Figures 6.2 (a) and 6.4 (a)).

Impact of the conductivity

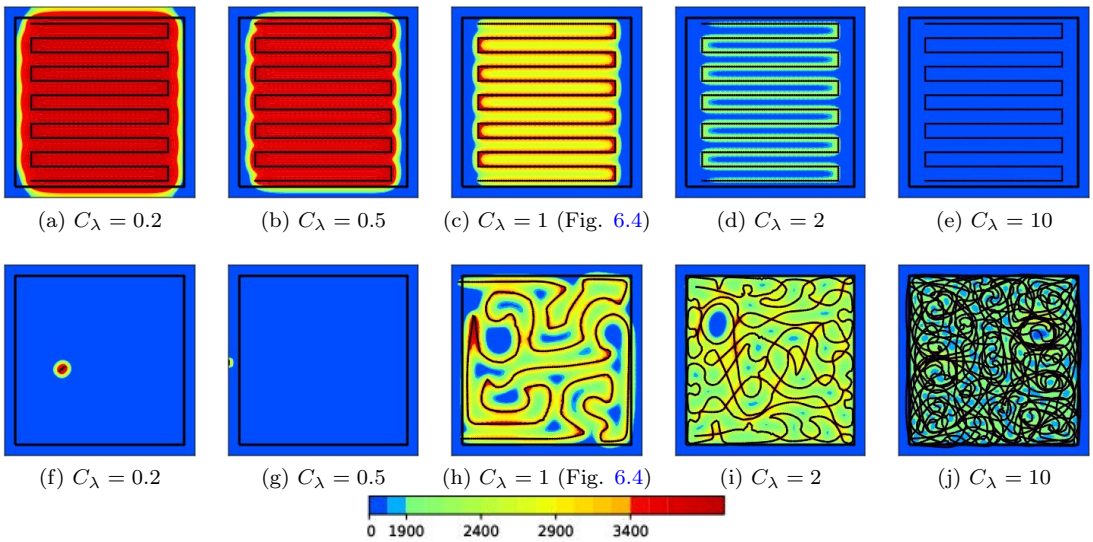
The first test consists in modifying the conductivity λ by multiplying the reference $\lambda = 130WK^{-1}m^{-1}$ by a factor C_λ. Note that modifying this coefficient also impacts the absorption coefficient $\beta = \lambda\Delta Z$. Figures 6.10 and 6.11 present five different results respectively in the aluminium and in the titanium cases, with $C_\lambda \in \{0.2, 0.5, 1, 2, 10\}$. The initialization is displayed on the first line and the result on the second.



(a) $C_\lambda = 0.2$ (b) $C_\lambda = 0.5$ (c) $C_\lambda = 1$ (Fig. 6.2) (d) $C_\lambda = 2$ (e) $C_\lambda = 10$

(f) $C_\lambda = 0.2$ (g) $C_\lambda = 0.5$ (h) $C_\lambda = 1$ (Fig. 6.2) (i) $C_\lambda = 2$ (j) $C_\lambda = 10$

Figure 6.10: Initial (first line) and final (second line) temperature and path depending on the conductivity factor C_λ (aluminium)



(a) $C_\lambda = 0.2$ (b) $C_\lambda = 0.5$ (c) $C_\lambda = 1$ (Fig. 6.4) (d) $C_\lambda = 2$ (e) $C_\lambda = 10$

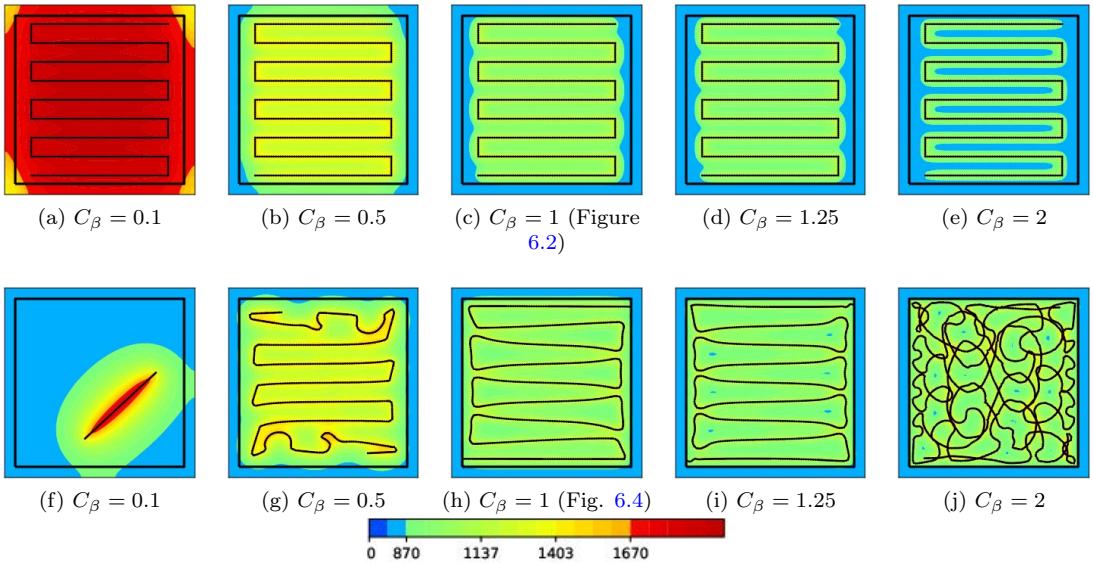
(f) $C_\lambda = 0.2$ (g) $C_\lambda = 0.5$ (h) $C_\lambda = 1$ (Fig. 6.4) (i) $C_\lambda = 2$ (j) $C_\lambda = 10$

Figure 6.11: Initial (first line) and final (second line) temperature and path depending on the conductivity factor C_λ (titanium)

The objective is to better understand the impact of the conductivity λ. We can conclude similarly for both materials: when the conductivity is decreased, the material is insulating. Thus, the temperature gets high and localized. To satisfy the maximum temperature constraint, the path length must be reduced. On the contrary, when the conductivity is increased, a source point impacts a large domain but the increase of temperature is limited. The energy does not only go in the rest of the domain but the absorption coefficient is also increased ($\beta = \lambda/\Delta Z$). This induces the stretching of the path.

Impact of the absorption

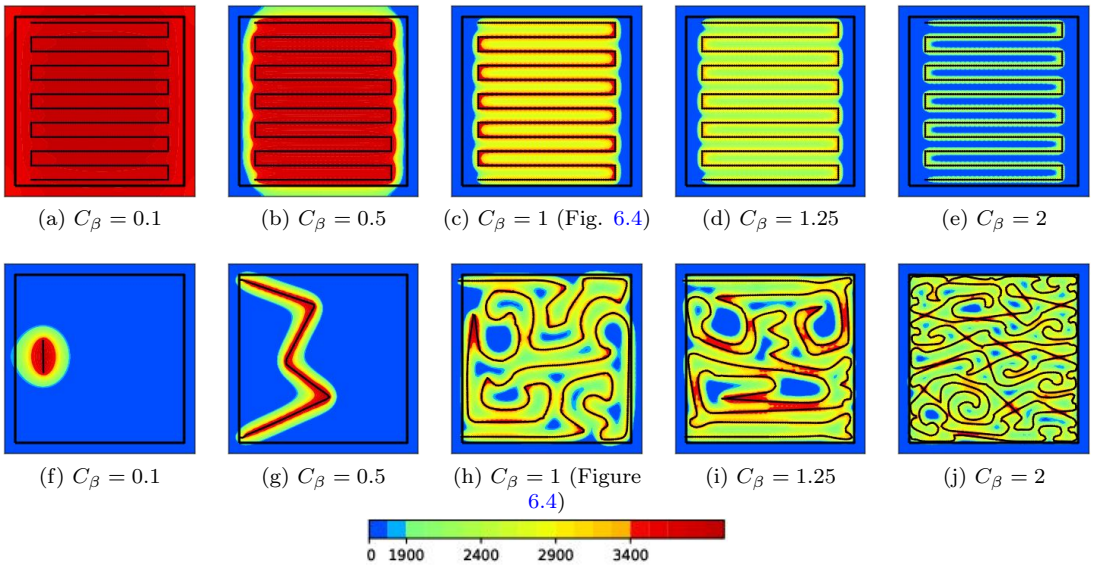
The second test consists in modifying the absorption β by multiplying the reference one $\beta = \frac{\lambda}{2.21229310^{-9}}$ by a factor C_β. Figure 6.12 presents the five different results in the aluminium case, with $C_\beta \in \{0.1, 0.5, 1, 1.25, 2\}$. On the first line, the initialization is displayed and on the second line the result. Figure 6.13 presents the titanium case.



(a) $C_\beta = 0.1$ (b) $C_\beta = 0.5$ (c) $C_\beta = 1$ (Figure 6.2) (d) $C_\beta = 1.25$ (e) $C_\beta = 2$

(f) $C_\beta = 0.1$ (g) $C_\beta = 0.5$ (h) $C_\beta = 1$ (Fig. 6.4) (i) $C_\beta = 1.25$ (j) $C_\beta = 2$

Figure 6.12: Initial (first line) and final (second line) temperature and path depending on the conductivity factor C_β (aluminium)



(a) $C_\beta = 0.1$ (b) $C_\beta = 0.5$ (c) $C_\beta = 1$ (Fig. 6.4) (d) $C_\beta = 1.25$ (e) $C_\beta = 2$

(f) $C_\beta = 0.1$ (g) $C_\beta = 0.5$ (h) $C_\beta = 1$ (Figure 6.4) (i) $C_\beta = 1.25$ (j) $C_\beta = 2$

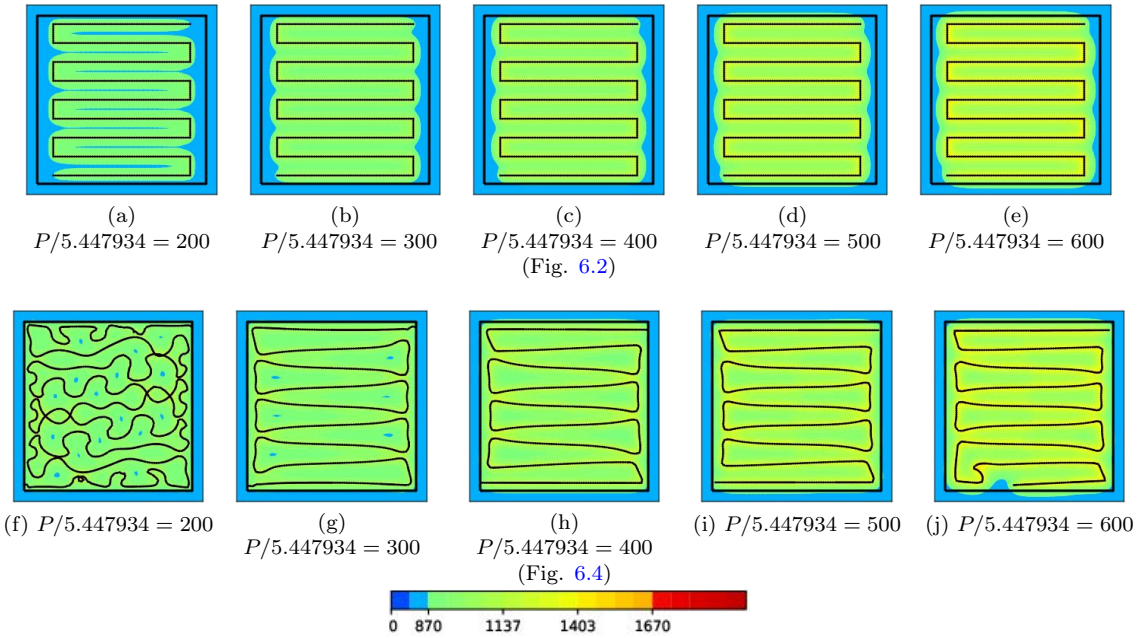
Figure 6.13: Initial (first line) and final (second line) temperature and path depending on the conductivity factor C_β (titanium)

From the energy deposed by the source, only part of it actually favors the change of phase on the top layer. Indeed, the rest is conducted in the already built layers. These tests aim at considering the impact of this absorption coefficient β. When the absorption is decreased, a larger part of the source energy remains in the top layer, increasing the temperatures. Thus, the path tends to be smaller on the two first columns of Figures 6.12 and 6.13. Actually, this temperature becomes high enough so that the maximum temperature constraint becomes impossible to fulfill, leading to a drastic reduction of the path length without taking into account the phase constraint. On the other hand, if the absorption is increased, less energy is available for the top layer, leading to an increase of the path length.

This absorption coefficient has been artificially introduced to create a two dimensional model. It appears to strongly impact the results, completely accounting for the importance of a calibration process. Yet, considering the three dimensional model improve the computations accuracy and allow for the evaluation of the scanning path's impact on the already built layers (crucial for porosity and residual stresses quantification, see Chapter 1).

Impact of the power

The third test consists in modifying the power P. Figure 6.14 presents five different results in the aluminium case, with $P/5.447934 \in \{200, 300, 400, 500, 600\}$. On the first line, the initialization is displayed and on the second line the result. Figure 6.15 presents the titanium case with $P/5.447934 \in \{100, 200, 300, 400, 500\}$.



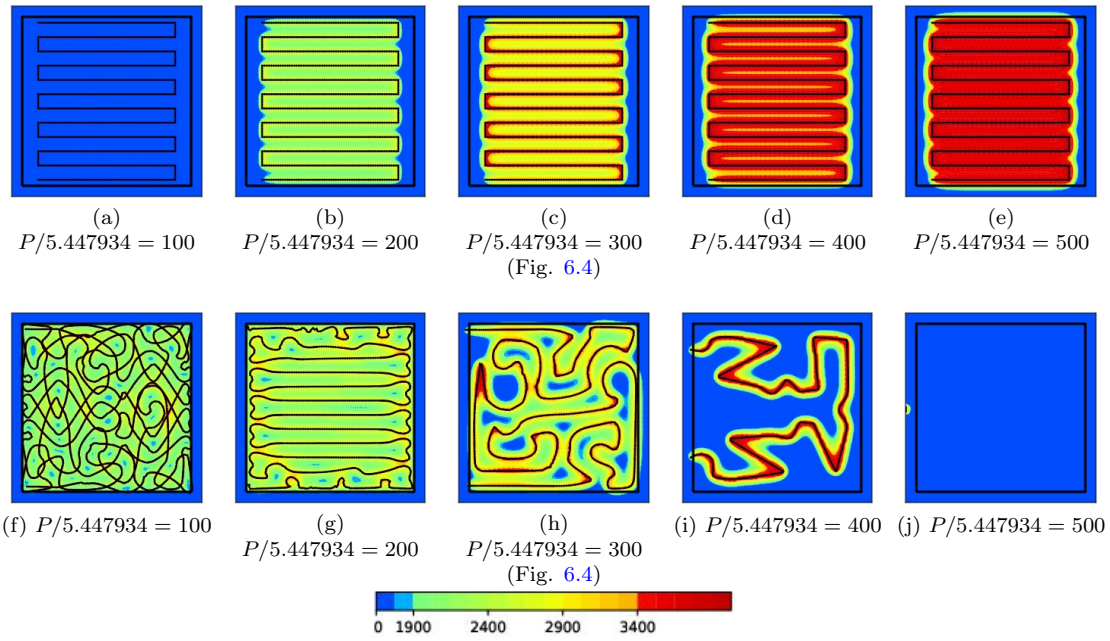
(a) $P/5.447934 = 200$ (b) $P/5.447934 = 300$ (c) $P/5.447934 = 400$ (Fig. 6.2) (d) $P/5.447934 = 500$ (e) $P/5.447934 = 600$

(f) $P/5.447934 = 200$ (g) $P/5.447934 = 300$ (h) $P/5.447934 = 400$ (Fig. 6.4) (i) $P/5.447934 = 500$ (j) $P/5.447934 = 600$

Figure 6.14: Initial (first line) and final (second line) temperature and path depending on the power (aluminium)

In most industrial applications, the laser power is adapted to the material used (or equivalently, the velocity could be modified). The values chosen are $P/5.447934 = 400W$ for the aluminium and $P/5.447934 = 300W$ for the titanium (see the Nomenclature). The study of the source power leads to similar conclusions than for the two previous tests: depending on the amount of energy deposed on the top layer, the path is shrunken or stretched. From this analysis of the parameters, it seems than the amount of deposed energy is a critical parameter in the scanning path design.

On the contrary to the conduction or absorption, which depend on the material, the power is a scanning parameter: different powers can be chosen before the scanning, the source can be switched on and off during the scanning and, in the future, this power might even be continuously modulated during the scanning. The tests have been realized for a reasonable range of powers but still lead to important results



(a) $P/5.447934 = 100$ (b) $P/5.447934 = 200$ (c) $P/5.447934 = 300$ (Fig. 6.4) (d) $P/5.447934 = 400$ (e) $P/5.447934 = 500$

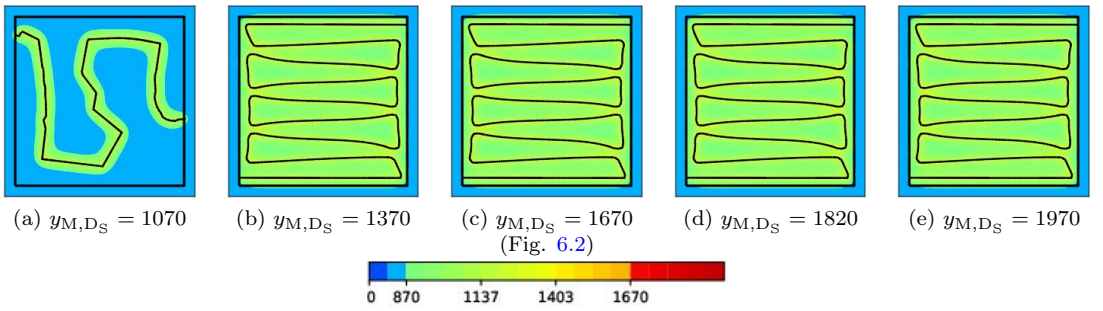
(f) $P/5.447934 = 100$ (g) $P/5.447934 = 200$ (h) $P/5.447934 = 300$ (Fig. 6.4) (i) $P/5.447934 = 400$ (j) $P/5.447934 = 500$

Figure 6.15: Initial (first line) and final (second line) temperature and path depending on the power (aluminium)

variations. The study of coupled control of the scanning path and the power is developed in Chapter 8.

Impact of the maximum temperature

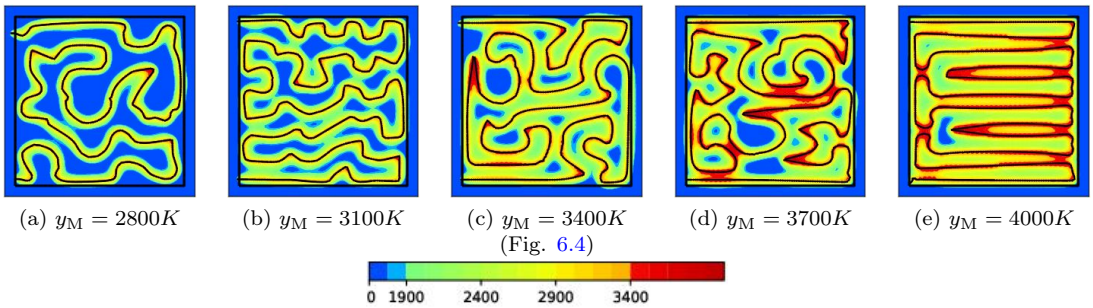
The fourth test consists in modifying the maximum temperature inside the object $y_{\mathrm{M,D_S}}$. Figure 6.16 presents five different results in the aluminium case, with $y_{\mathrm{M,D_S}} \in \{1070, 1370, 1670, 1820, 1970\}$, initialized with a zigzag with 9 lines (Figure 6.2(a)). Figure 6.17 presents the titanium case with $y_{\mathrm{M}} \in \{2800, 3100, 3400, 3700, 4000\}$, initialized with a zigzag with 12 lines (Figure 6.4(a)). The color scales are not modified: the objective is to intuitively observe the design differences in a qualitative way.



(a) $y_{\mathrm{M,D_S}} = 1070$ (b) $y_{\mathrm{M,D_S}} = 1370$ (c) $y_{\mathrm{M,D_S}} = 1670$ (Fig. 6.2) (d) $y_{\mathrm{M,D_S}} = 1820$ (e) $y_{\mathrm{M,D_S}} = 1970$

Figure 6.16: Initial (first line) and final (second line) temperature and path depending on the maximum temperature authorized within the solid y_{M} (aluminium)

The maximum temperature constraint aims at controlling thermal expansion and residual stresses creation during the building process. It is clear that the model is far too simple to constraint such complicated phenomena. However, although the results are preliminary, it allows for first computations and an algorithm that could be adapted to more realistic problems. These tests aim at assessing the impact of the maximal temperature $y_{\mathrm{M,D_S}}$ on the results. In the aluminium case, the temperature remains low because of the high conduction (and thus absorption). High maximum temperatures lead to similar results. On the contrary, a lower maximal temperature leads to adaptations of the path. In the titanium case, the impact of this choice is clear. Indeed, the final path clearly adapts more for smaller temperatures,



(a) $y_{\mathrm{M}} = 2800K$ (b) $y_{\mathrm{M}} = 3100K$ (c) $y_{\mathrm{M}} = 3400K$ (Fig. 6.4) (d) $y_{\mathrm{M}} = 3700K$ (e) $y_{\mathrm{M}} = 4000K$

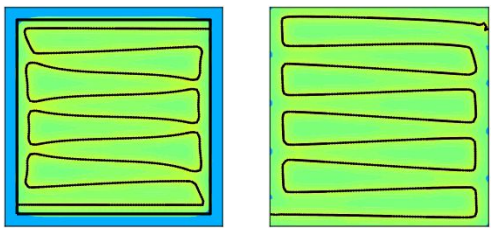
Figure 6.17: Initial (first line) and final (second line) temperature and path depending on the maximum temperature authorized within the solid y_{M} (titanium)

drastically reducing its length.

For the titanium, the value chosen in the following ($y_{\mathrm{M,D_S}} = 3400K$, see the Nomenclature) is close to the boiling temperature for the titanium. As for the aluminium ($y_{\mathrm{M,D_S}} = 1670K$, see the Nomenclature) it has been arbitrarily chosen.

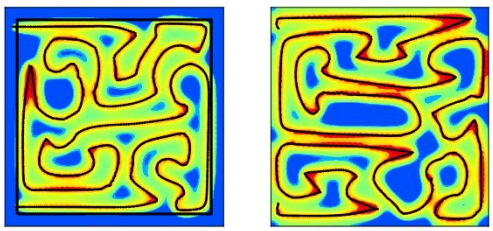
6.5.4 Building the whole working domain

A last feature to consider is the projection step in the algorithms presented (see Section 6.4.1 and 6.4.2). The convergence of the projected AL method and projected null space method have not been proved and may impact the results. To illustrate this, the AL algorithm with $\left(l_C^0 = 1, c = 10\right)$ has been run in the same conditions that in Section 6.5.1 and we have applied an homothety (ratio 1.1) to adapt the zigzag initializations to the initializations used for the square part. Figure 6.18 displays the comparison between both results in the aluminium case whereas Figure 6.19 gives the titanium results.



(a) Building a square (Figure 6.2) (b) Building the whole working domain

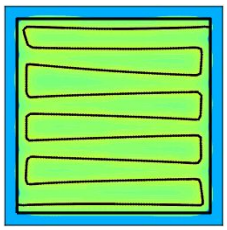
Figure 6.18: Comparison between building a square and the whole working domain, computed using an augmented Lagrangian algorithm (aluminium).



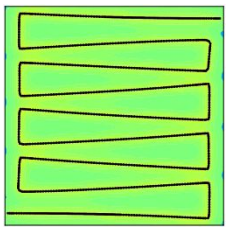
(a) Building a square (Figure 6.4) (b) Building the whole working domain

Figure 6.19: Comparison between building a square and the whole working domain, computed using an augmented Lagrangian algorithm (titanium).

The same tests have been run with the null space gradient algorithm with $(\alpha_{L_{\mathrm{F}}}, \alpha_C) = (1, 1)$, to be compared to the results presented in Figures 6.8 (f) and 6.9 (f). Figure 6.20 shows the comparison between both results in the aluminium case whereas the titanium case is shown in Figure 6.21.

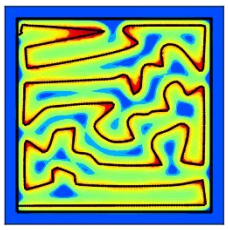


(a) Building a square (Figure 6.2)

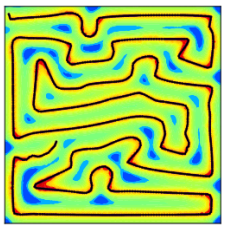


(b) Building the whole working domain

Figure 6.20: Comparison between building a square and the whole working domain, computed using a null space gradient algorithm (aluminium).



(a) Building a square (Figure 6.4)



(b) Building the whole working domain

Figure 6.21: Comparison between building a square and the whole working domain, computed using a null space gradient algorithm (titanium).

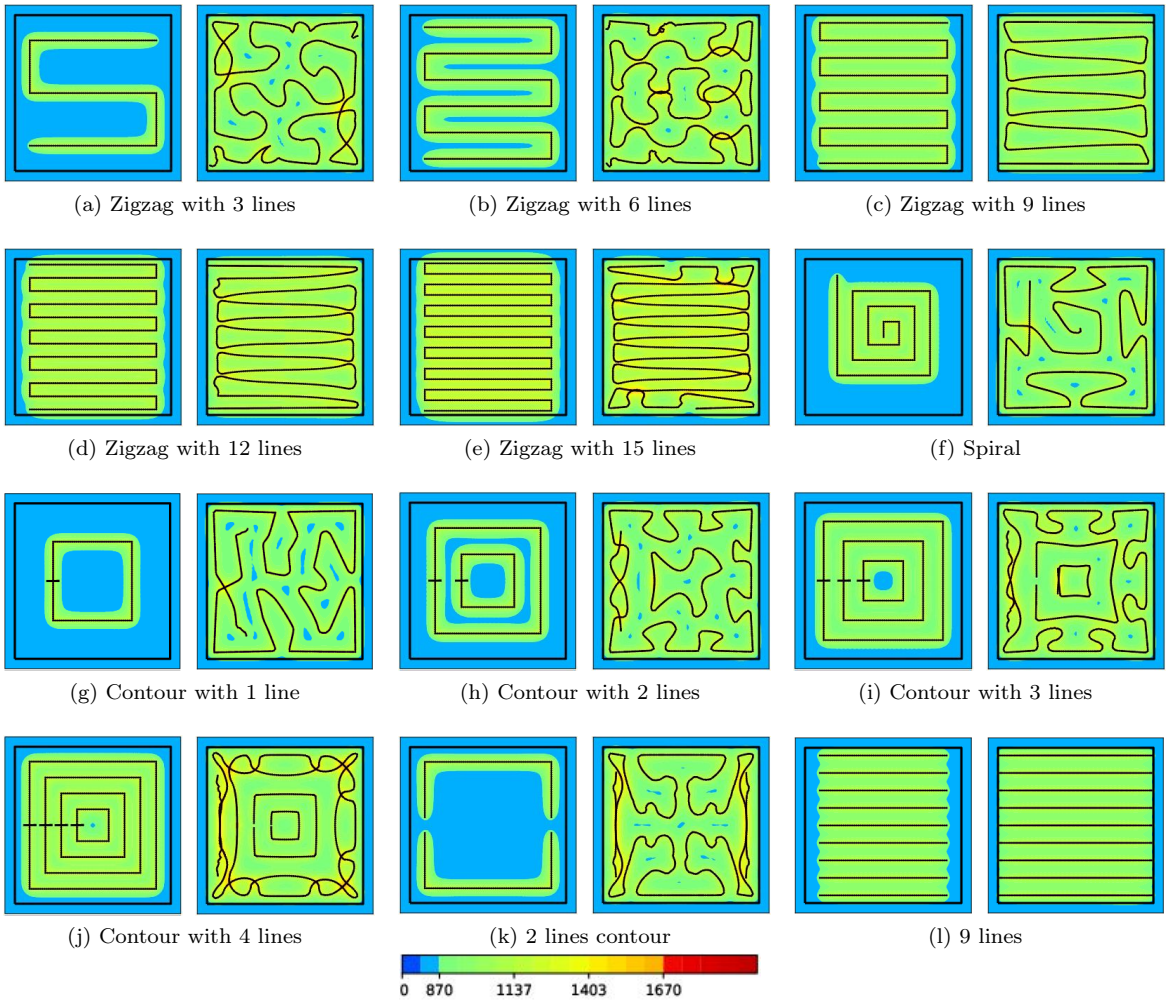
For the aluminium material, in both the Augmented Lagrangian and null space gradient methods, the results for the square or the whole box are similar and the impact of the projection is not clear. For the titanium, the differences are stronger but it is not clear if this is because of the projection or because of the strong non convex phenomena.

6.6 Initialization and geometry to build impact for the aluminium

The analysis of the optimization and physical parameters has provided a better understanding on the various choices we made in this work. We now further study the optimal path, first running the process from several different initializations and then modifying the part to build from the square shown in Figure 6.1 to the three geometries shown in Figures 3.6, 3.7 and 3.8.

Scanning a square

The first test consists in running the code starting from several different initializations, to scan the square presented in Figure 6.1. The optimization problem is not convex and, as noticed while testing different algorithms, many local minima exist. To better understand these minima, ten different initializations are tested, including some composed of several disconnected curves. The initializations and results are presented in Figure 6.22 and the final values summed up in Table 6.5.



(a) Zigzag with 3 lines (b) Zigzag with 6 lines (c) Zigzag with 9 lines

(d) Zigzag with 12 lines (e) Zigzag with 15 lines (f) Spiral

(g) Contour with 1 line (h) Contour with 2 lines (i) Contour with 3 lines

(j) Contour with 4 lines (k) 2 lines contour (l) 9 lines

Figure 6.22: Initialization and final results depending on the initialization (aluminium).

Case	$L_{\mathrm{F}}^{\mathrm{ini}}$ (m)	$L_{\mathrm{F}}^{\mathrm{fin}}$ (m)	$\overline{C}_{\phi}^{\mathrm{ini}}$	$\overline{C}_{\phi}^{\mathrm{fin}}$	$\overline{C}_{\mathrm{M,D_S}}^{\mathrm{ini}}$	$\overline{C}_{\mathrm{M,D_S}}^{\mathrm{fin}}$	$\overline{C}_{\mathrm{M,D\setminus D_S}}^{\mathrm{ini}}$	$\overline{C}_{\mathrm{M,D\setminus D_S}}^{\mathrm{fin}}$
zigzag 3 lines	3.864e−3	1.119e−2	4.15e−3	6.40e−7	0.00	0.00	0.00	1.11e−6
zigzag 6 lines	7.098e−3	1.265e−2	5.85e−4	4.92e−7	0.00	0.00	0.00	2.17e−6
zigzag 9 lines	1.019e−2	1.159e−2	3.31e−4	4.20e−7	0.00	0.00	3.09e−7	2.84e−5
zigzag 12 lines	1.325e−2	1.479e−2	2.34e−4	5.46e−7	0.00	0.00	2.70e−4	7.63e−5
zigzag 15 lines	1.630e−2	1.727e−2	1.78e−4	9.91e−7	0.00	0.00	1.12e−3	3.53e−5
spiral	5.635e−3	1.060e−2	4.60e−3	1.37e−6	0.00	0.00	0.00	1.61e−5
contour 1 line	2.520e−3	9.670e−3	6.83e−3	2.93e−6	0.00	0.00	0.00	1.28e−5
contour 2 lines	5.040e−3	1.080e−2	2.85e−3	1.27e−6	0.00	0.00	0.00	4.16e−6
contour 3 lines	7.560e−3	1.190e−2	1.11e−3	6.40e−7	0.00	0.00	0.00	3.59e−6
contour 4 lines	1.008e−2	1.334e−2	4.25e−4	2.20e−7	0.00	0.00	0.00	2.80e−7
2 lines contour	3.780e−3	1.214e−2	4.81e−3	6.44e−7	0.00	0.00	0.00	5.82e−7
9 lines	9.072e−3	1.093e−2	7.99e−4	4.17e−7	0.00	0.00	2.17e−7	2.02e−6

Table 6.5: Comparison of the cost and the constraints of the final results (aluminium).

The graph of the final length with respect to the aggregated constraint $\overline{C} = \overline{C}_\phi + \overline{C}_{\mathrm{M,D_S}} + \overline{C}_{\mathrm{M,D\setminus D_S}}$ in each case is given in Figure 6.23, facilitating the comparison between the different initializations.

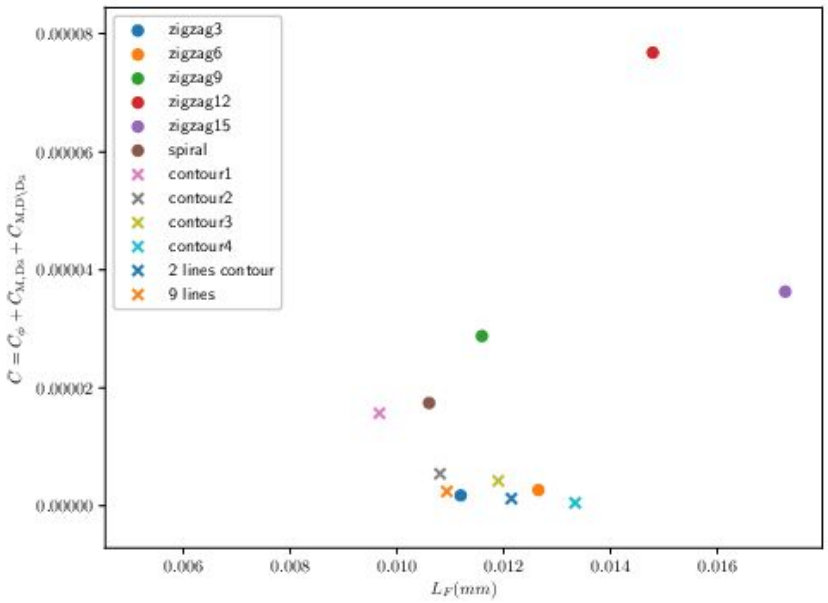


Figure 6.23: Constraint $\overline{C} = \overline{C}_\phi + \overline{C}_{\mathrm{M,D_S}} + \overline{C}_{\mathrm{M,D\setminus D_S}}$ with respect to the length L_F (aluminium).

First note that, whatever initialization is chosen, the optimization algorithm seems to work since the path is adapted to the objective and constraints. If these latter do not vanish at the end of the process, they have still decreased since the starting guess. Two results are further discussed. The initialization with 9 disconnected lines leads to 9 straight lines. Then the initialization with a zigzag with 9 lines confirms recent studies [77] and can already be used in industrial applications.

A second consideration is relative to the final lengths. Indeed, starting from different initializations, it appears that most of the results belong to the same length range $[0.010, 0.014]$. Two exceptions must be pointed out: the zigzag with 15 lines and the zigzag with 12 lines. Indeed, the initialization paths were already quite long and the optimization process had to shrink them. However, shrinking is not an easy process. Indeed, it often induces sharp corners that drastically increase the maximal temperature constraint and the process must go through local maxima to reach better results. This might be blocked by a low tolerance in the iteration acceptation.

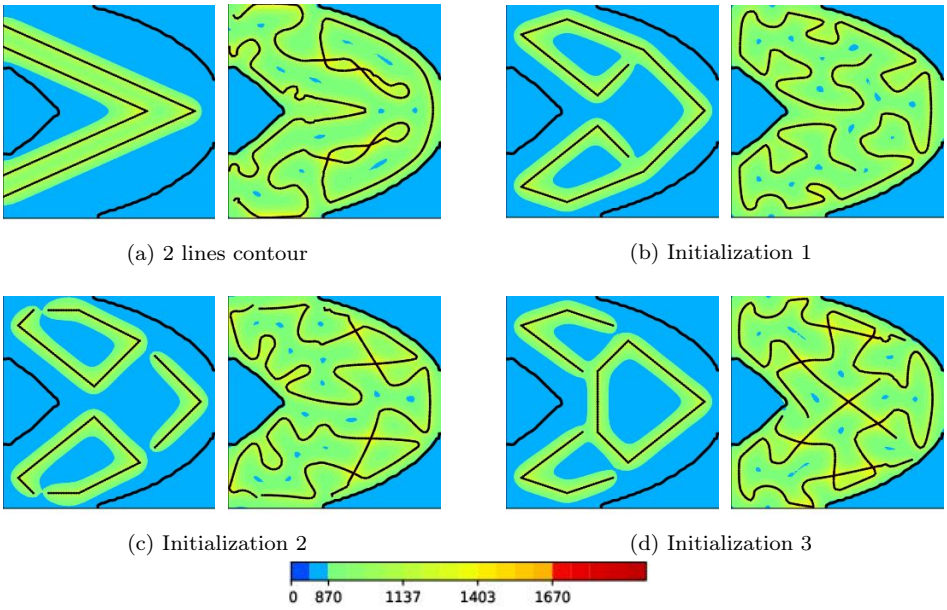
This common length range steers the notion of optimal amount of energy for the layer. For this design, the energy $PL_\mathrm{F} \approx 0.012 * P$ is enough to melt the whole required powder. This observation matches the existing assessments on optimal energy deposition rates [43, 48, 83] and is an interesting perspective to ease the path design.

Scanning complex geometries

The final objective of this work being to couple shape and path optimization, the test now consists in designing a path adapted to the different shapes in Chapter 3, obtained for volume and compliance optimization. One of the objective of this work is indeed to determine new design criteria adapted to scanning constraints. Three reference shapes have been generated by the shape optimization problem presented in Section 3.3.2 and are presented in Figures 3.6, 3.7 and 3.8. For each of them, four initializations have been tested and the results are respectively presented in Figures 6.24, 6.26, 6.28, Tables 6.6, 6.7, 6.8. We recall that the values presented in the Tables follow (6.5.1). In this context, we have:

- for the zero hole object: $|D_S| = 1.53\mathrm{e}{-6}m^2$ and $|D \setminus D_S| = 4.34\mathrm{e}{-7}m^2$
- for the one hole object: $|D_S| = 1.13\mathrm{e}{-6}m^2$ and $|D \setminus D_S| = 8.31\mathrm{e}{-7}m^2$,
- for the three holes object: $|D_S| = 1.02\mathrm{e}{-6}m^2$ and $|D \setminus D_S| = 9.40\mathrm{e}{-7}m^2$.

Figures 6.25, 6.27, 6.29 represent the graphs of the length versus the aggregated constraint.



(a) 2 lines contour (b) Initialization 1

(c) Initialization 2 (d) Initialization 3

Figure 6.24: Initialization and final results depending on the initialization for the zero hole object shown in Figure 3.6 (aluminium).

Case	$L_{\mathrm{F}}^{\mathrm{ini}}$ (m)	$L_{\mathrm{F}}^{\mathrm{fin}}$ (m)	$\overline{C}_{\phi}^{\mathrm{ini}}$	$\overline{C}_{\phi}^{\mathrm{fin}}$	$\overline{C}_{\mathrm{M,D_S}}^{\mathrm{ini}}$	$\overline{C}_{\mathrm{M,D_S}}^{\mathrm{fin}}$	$\overline{C}_{\mathrm{M,D\setminus D_S}}^{\mathrm{ini}}$	$\overline{C}_{\mathrm{M,D\setminus D_S}}^{\mathrm{fin}}$
2 lines contour	4.826e−3	1.042e−2	4.76e−3	1.85e−7	0.00	0.00	0.00	5.79e−7
initialization 1	4.687e−3	9.961e−3	2.95e−3	3.69e−7	0.00	0.00	0.00	1.81e−7
initialization 2	4.555e−3	1.041e−2	3.01e−3	3.48e−7	0.00	0.00	0.00	1.48e−7
initialization 3	4.511e−3	1.111e−2	3.29e−3	4.63e−7	0.00	0.00	0.00	1.41e−6

Table 6.6: Comparison of the cost and the constraints of the final results for the zero hole object (Figure 3.6) (aluminium).

In spite of the volume difference, this first object is built as simply as the square object and the same conclusions hold: all the computations result in similar path length with a comparable satisfaction of the aggregated constraint C (slightly above for Initialization 3, see Figure 6.24(d))). Note that although all initializations are symmetric, the results are not. Indeed, even if the conductivity is large, the optimization process remains unstable and any small perturbation can lead to a different solution.

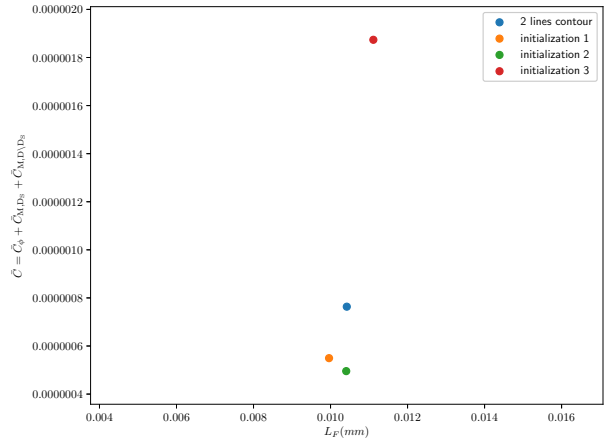
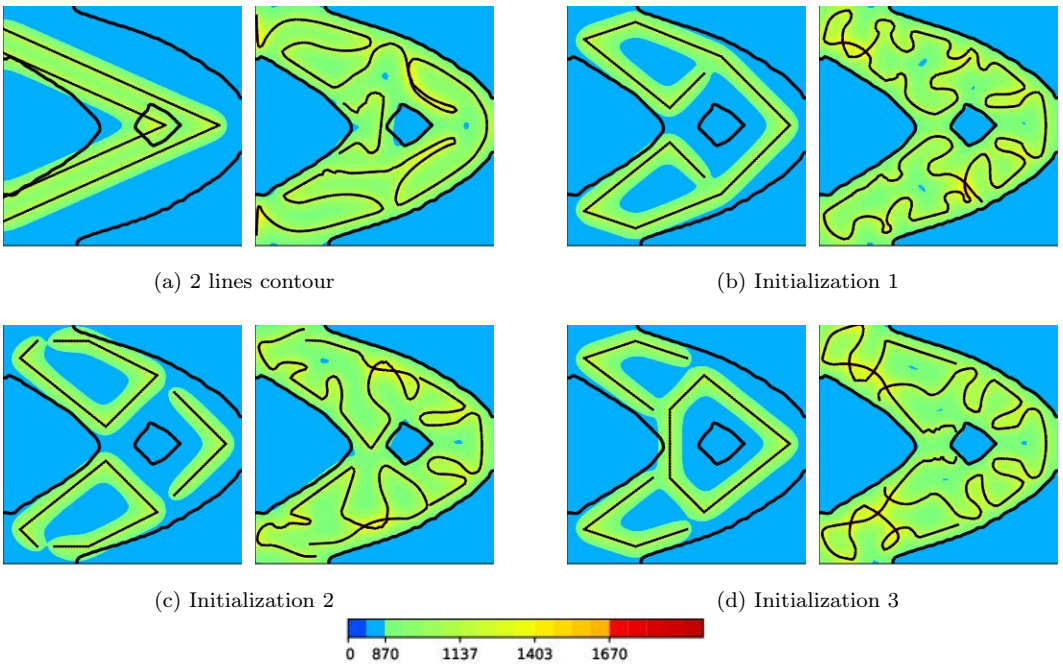


Figure 6.25: Constraint $\overline{C} = \overline{C}_\phi + \overline{C}_{\mathrm{M,D_S}} + \overline{C}_{\mathrm{M,D\backslash D_S}}$ with respect to the length L_F for the final results for the zero hole object (Figure 3.6) (aluminium).



(a) 2 lines contour

(b) Initialization 1

(c) Initialization 2

(d) Initialization 3

Figure 6.26: Initialization and final results depending on the initialization for the one hole object shown in Figure 3.7 (aluminium).

Case	$L_\mathrm{F}^\mathrm{ini}$ (m)	$L_\mathrm{F}^\mathrm{fin}$ (m)	$\overline{C}_\phi^\mathrm{ini}$	$\overline{C}_\phi^\mathrm{fin}$	$\overline{C}_{\mathrm{M,D_S}}^\mathrm{ini}$	$\overline{C}_{\mathrm{M,D_S}}^\mathrm{fin}$	$\overline{C}_{\mathrm{M,D\backslash D_S}}^\mathrm{ini}$	$\overline{C}_{\mathrm{M,D\backslash D_S}}^\mathrm{fin}$
2 lines contour	4.826e−3	8.861e−3	4.12e−3	1.08e−6	0.00	0.00	3.83e−3	2.85e−6
initialization 1	4.687e−3	9.335e−3	1.74e−3	3.64e−7	0.00	0.00	6.33e−7	3.95e−7
initialization 2	4.555e−3	8.977e−3	1.90e−3	5.59e−7	0.00	0.00	1.32e−5	2.84e−6
initialization 3	4.511e−3	9.434e−3	1.92e−3	1.56e−7	0.00	0.00	4.75e−6	1.01e−7

Table 6.7: Comparison of the cost and the constraints of the final results for the one hole object (Figure 3.7) (aluminium).

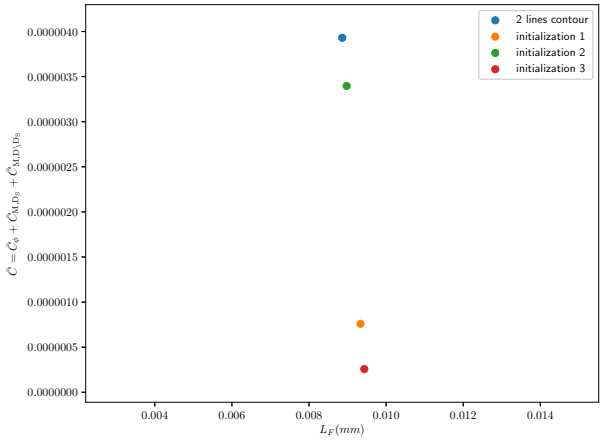
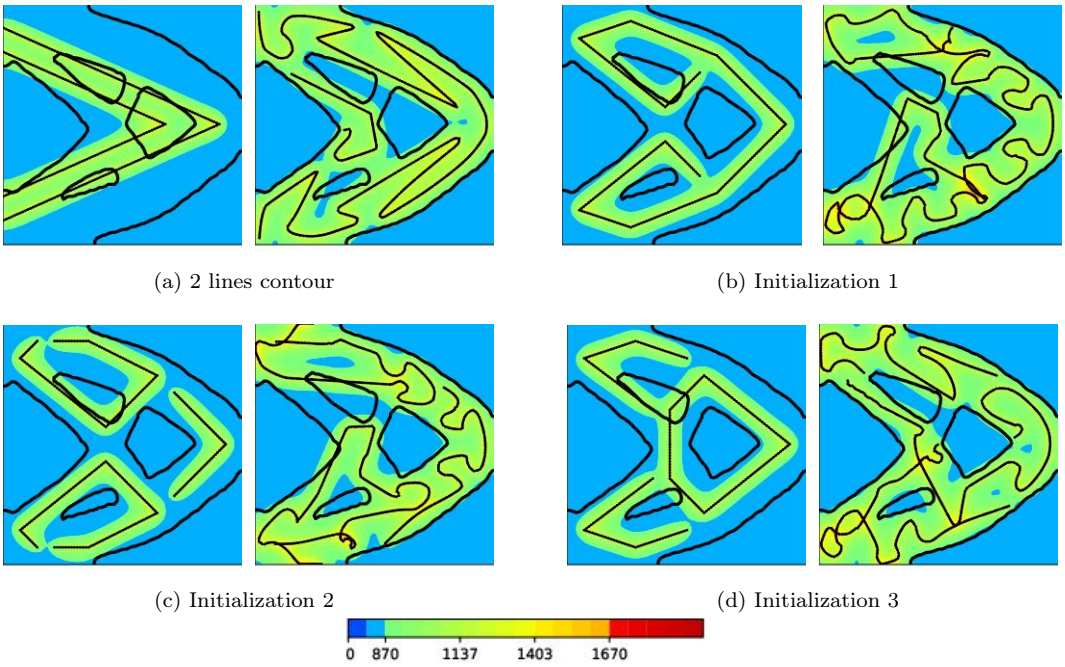


Figure 6.27: Constraint $\overline{C} = \overline{C}_\phi + \overline{C}_{\mathrm{M,D_S}} + \overline{C}_{\mathrm{M,D\backslash D_S}}$ with respect to the length L_F for the final results for the one hole object (Figure 3.7) (aluminium).

The second object is slightly more complicated to build: the hole introduced at the center must be avoided. Once again, the optimization algorithm seems to work: even when the initial path crosses a hole, it manages to generate a correct final path (Figure 6.26(a)).



(a) 2 lines contour

(b) Initialization 1

(c) Initialization 2

(d) Initialization 3

Figure 6.28: Initialization and final results depending on the initialization for the three holes object shown in Figure 3.8 (aluminium).

Case	$L_{\mathrm{F}}^{\mathrm{ini}}$ (m)	$L_{\mathrm{F}}^{\mathrm{fin}}$ (m)	$\overline{C}_{\phi}^{\mathrm{ini}}$	$\overline{C}_{\phi}^{\mathrm{fin}}$	$\overline{C}_{\mathrm{M,D_S}}^{\mathrm{ini}}$	$\overline{C}_{\mathrm{M,D_S}}^{\mathrm{fin}}$	$\overline{C}_{\mathrm{M,D\setminus D_S}}^{\mathrm{ini}}$	$\overline{C}_{\mathrm{M,D\setminus D_S}}^{\mathrm{fin}}$
2 lines contour	4.826e−3	8.735e−3	4.80e−3	2.68e−5	0.00	0.00	1.02e−2	8.22e−5
initialization 1	4.687e−3	8.880e−3	1.45e−3	1.80e−4	0.00	0.00	7.03e−4	1.24e−3
initialization 2	4.555e−3	8.646e−3	1.57e−3	2.44e−4	0.00	0.00	7.06e−4	1.17e−3
initialization 3	4.511e−3	8.794e−3	1.86e−3	4.71e−5	0.00	0.00	1.31e−3	1.75e−4

Table 6.8: Comparison of the cost and the constraints of the final results for the three holes object (Figure 3.8) (aluminium).

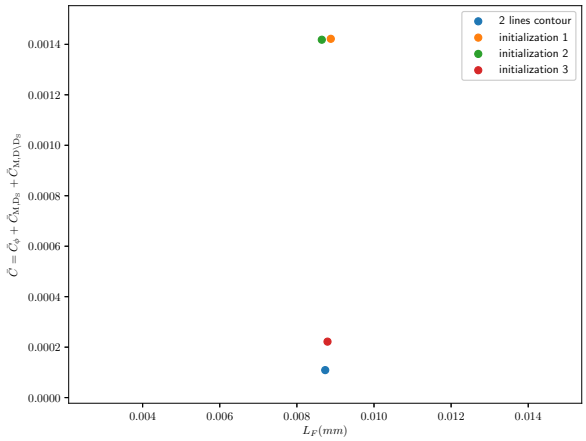
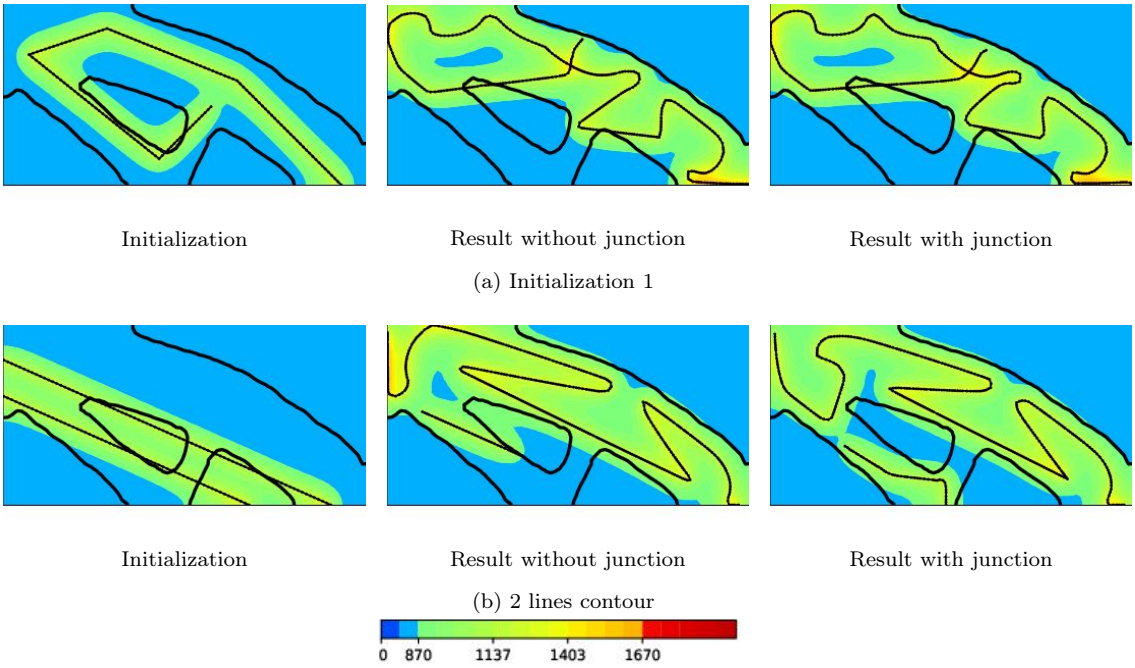


Figure 6.29: Constraint $\overline{C} = \overline{C}_{\phi} + \overline{C}_{\mathrm{M,D_S}} + \overline{C}_{\mathrm{M,D\setminus D_S}}$ with respect to the length L_{F} for the final results for the three holes object (Figure 3.8) (aluminium).

The final object is the hardest case because of the three holes. The tests starting from the 2 lines contour initialization, Initialization 1 and Initialization 2 do not manage to satisfy the constraints and remain out of the admissible domain. The last one, however, gives correct results. The initialization thus matters and, in order to get to reasonable solutions, should be carefully chosen.

The non-symmetry of the results can be explained by two different reasons. First, each of the considered objects are not symmetric. In Chapter 3, they have been generated by shape optimization of volume and compliance. However, no symmetry condition had been enforced. Then, even if the shape was symmetric, because of small numerical errors along the iterations, the resulting path might not be symmetric. To remedy this problem and enforce the symmetry, the shape optimization process detailed in Section 3.3.2 has been run on half the object only, with two possibilities: first, one end of the path connected component must belong to the symmetry axis (Ox) to connect the components with the second half, and then, no connecting condition. Two initializations are starting from: the 2 lines contour and Initialization 1. The initializations and results are given by Figure 6.30 and the final values by Table 6.9 (for the whole domain D, results comparable to Table 6.8).



(a) Initialization 1

(b) 2 lines contour

Figure 6.30: Final results in the aluminium case depending on the initialization for the three holes object with symmetry

Case	$L_{\mathrm{F}}^{\mathrm{ini}}$ (m)	$L_{\mathrm{F}}^{\mathrm{fin}}$ (m)	$\overline{C}_{\phi}^{\mathrm{ini}}$	$\overline{C}_{\phi}^{\mathrm{fin}}$	$\overline{C}_{\mathrm{M,D_S}}^{\mathrm{ini}}$	$\overline{C}_{\mathrm{M,D_S}}^{\mathrm{fin}}$	$\overline{C}_{\mathrm{M,D\setminus D_S}}^{\mathrm{ini}}$	$\overline{C}_{\mathrm{M,D\setminus D_S}}^{\mathrm{fin}}$
Initialization 1								
without junction	4.687e−3	7.345e−3	1.50e−3	1.36e−3	0.00	0.00	5.87e−2	5.15e−2
with junction	4.687e−3	7.452e−3	1.50e−3	1.40e−3	0.00	0.00	5.87e−2	5.23e−2
2 lines contour								
without junction	4.826e−3	8.569e−3	4.72e−3	8.50e−4	0.00	0.00	4.98e−1	1.46e−2
with junction	4.826e−3	8.399e−3	4.72e−3	3.86e−5	0.00	0.00	4.98e−1	5.86e−3

Table 6.9: Comparison of the cost and the constraints of the final results for the three holes object with symmetry (aluminium).

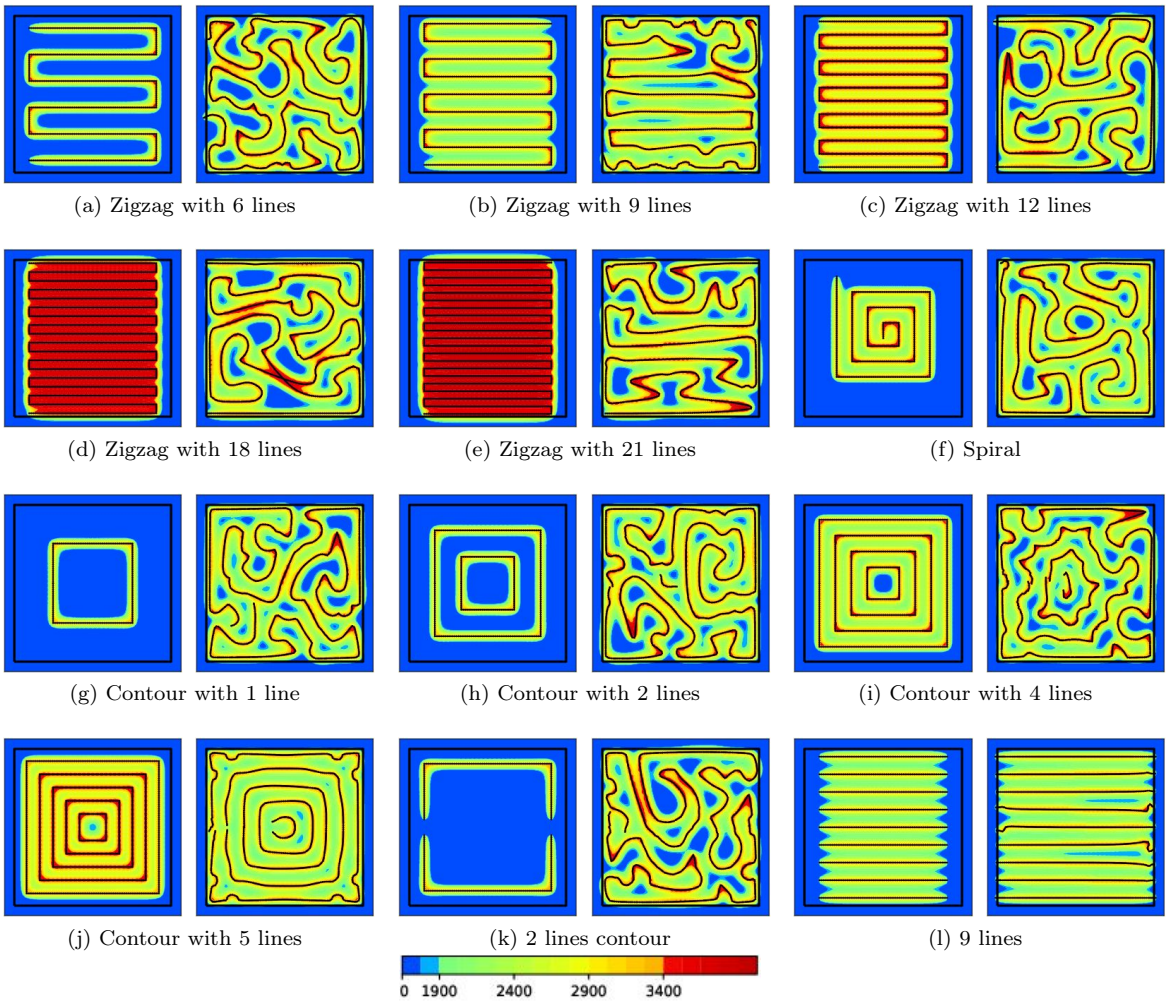
The results confirm that this geometry is not easily scanned for aluminium powder. The result from Initialization 1, in both tests, does not build the lower bar of the object: it seems that this bar is to thin to be correctly realized. The result from the 2 lines contour is slightly better. Indeed, allowing two connected components let one of them to be well adjusted to the thin bar.

6.7 Initialization and geometry to build impact for the titanium

The conclusions brought by the analysis in the aluminium case are now evaluated with the titanium, which conductivity is much lower.

6.7.1 Scanning a square

The first tests consider the building of the square object (Figure 6.1). The initializations and results are summed up in Figure 6.31 and the final values summed up in Table 6.10.



(a) Zigzag with 6 lines (b) Zigzag with 9 lines (c) Zigzag with 12 lines

(d) Zigzag with 18 lines (e) Zigzag with 21 lines (f) Spiral

(g) Contour with 1 line (h) Contour with 2 lines (i) Contour with 4 lines

(j) Contour with 5 lines (k) 2 lines contour (l) 9 lines

Figure 6.31: Initialization and final results depending on the initialization (titanium).

Case	$L_{\mathrm{F}}^{\mathrm{ini}}$ (m)	$L_{\mathrm{F}}^{\mathrm{fin}}$ (m)	$\overline{C}_{\phi}^{\mathrm{ini}}$	$\overline{C}_{\phi}^{\mathrm{fin}}$	$\overline{C}_{\mathrm{M,D_S}}^{\mathrm{ini}}$	$\overline{C}_{\mathrm{M,D_S}}^{\mathrm{fin}}$	$\overline{C}_{\mathrm{M,D\backslash D_S}}^{\mathrm{ini}}$	$\overline{C}_{\mathrm{M,D\backslash D_S}}^{\mathrm{ini}}$
zigzag 6 lines	7.098e−3	1.231e−2	4.81e−2	2.80e−3	3.48e−6	1.11e−4	0.00	3.25e−3
zigzag 9 lines	1.019e−2	1.232e−2	1.96e−2	8.99e−4	1.96e−5	1.19e−4	0.00	4.35e−4
zigzag 12 lines	1.325e−2	1.247e−2	1.55e−2	3.26e−3	5.11e−4	3.37e−4	0.00	2.95e−3
zigzag 18 lines	1.933e−2	1.286e−2	1.14e−2	3.21e−3	3.83e−2	9.34e−4	8.34e−3	3.77e−3
zigzag 21 lines	2.237e−2	1.284e−2	9.82e−3	2.39e−3	1.10e−1	2.63e−4	2.45e−2	2.64e−3
spiral	5.635e−3	1.286e−2	1.50e−1	5.40e−4	4.71e−5	4.21e−5	0.00	4.97e−4
contour 1 line	2.520e−3	1.249e−2	2.21e−1	1.20e−3	0.00	1.05e−4	0.00	1.24e−3
contour 2 lines	5.040e−3	1.270e−2	1.17e−1	1.13e−3	0.00	8.00e−5	0.00	1.07e−3
contour 4 lines	1.008e−2	1.234e−2	2.74e−2	1.01e−3	2.43e−5	2.34e−4	0.00	1.39e−3
contour 5 lines	1.260e−2	1.269e−2	1.14e−2	1.06e−5	4.54e−4	1.12e−7	0.00	4.27e−5
2 lines contour	3.780e−3	1.243e−2	1.71e−1	2.87e−3	0.00	1.56e−4	0.00	1.52e−3
9 lines	9.072e−3	1.148e−2	3.60e−2	5.67e−4	0.00	7.90e−7	0.00	3.40e−4

Table 6.10: Comparison of the cost and the constraints of the final results (titanium).

The graph of the final length with respect to the aggregated constraint $\overline{C} = \overline{C}_{\phi} + \overline{C}_{\mathrm{M,D_S}} + \overline{C}_{\mathrm{M,D\setminus D_S}}$ in each case is given in Figure 6.32, facilitating the comparison between the different initializations.

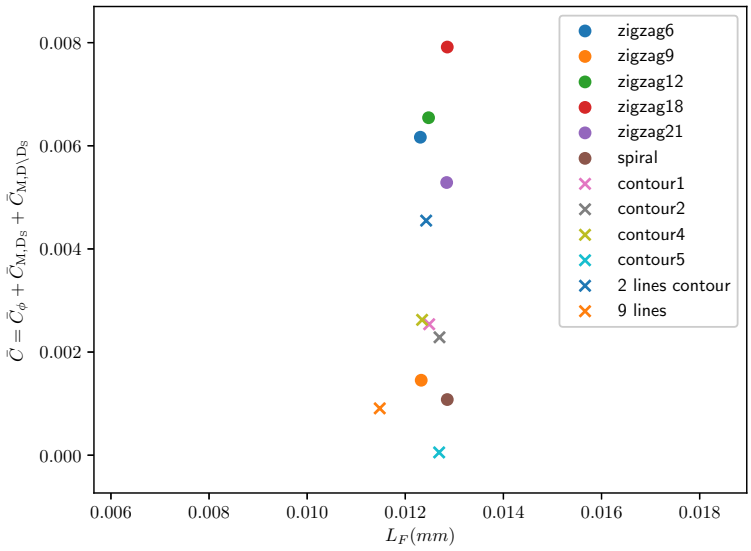


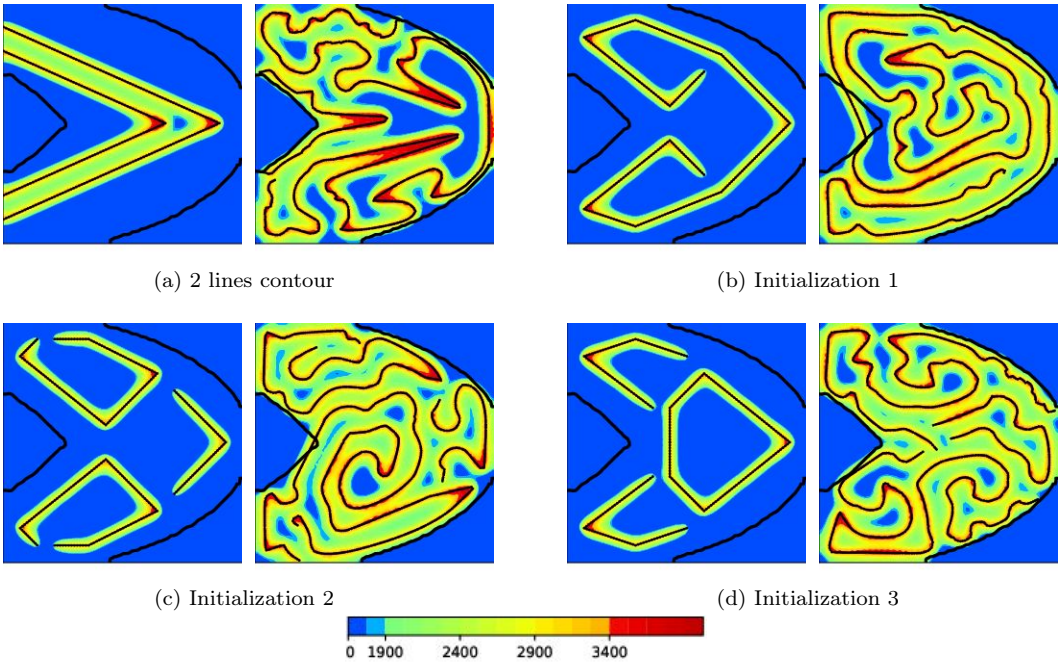
Figure 6.32: Constraint $\overline{C} = \overline{C}_{\phi} + \overline{C}_{\mathrm{M,D_S}} + \overline{C}_{\mathrm{M,D\setminus D_S}}$ with respect to the length L_{F} (titanium).

The results in the titanium context coincide with the conclusion obtained in the aluminium case. First of all, even with a lower conductivity, the algorithm still manages to compute optimized path while satisfying the constraints. Moreover, all these final paths have very similar lengths: the range width is even smaller than for the aluminium. One case is to be noticed: the zigzag with 18 lines. Indeed, the algorithm could not manage to remove some length and crossing have been introduced breaking the maximum temperature requirements. One could think that any too long initialization would lead to the same kind of results. However, the zigzag with 21 lines initialization leads to a correct path, with final values even better than for the initialization with a zigzag with 12 lines (Table 6.10). It thus seems easier for the algorithm to increase the path length (Figure 6.31(f, g, h, i, k)) than reducing it even if good results can be obtained in both cases.

Two other results must be pointed out. The first one is the contour with 4 lines, leading to very low final constraints. One can notice that the initialization was already a good guess and the algorithm has not completely transformed the path. It has simply slightly adapted it to fully satisfy the objectives. A second result is the initialization with 9 lines. Indeed, it led to the smallest final path and, once again, very few modifications have been done on the initial guess. However, the result found is not as clear as it was in the aluminium case (Figure 6.22(l)), once again pointing out the influence of the conductivity on the final path. These observations confirm the strong influence of the initialization.

6.7.2 Scanning complex geometries

The building objects tests are now run for the titanium. For each object, four initializations have been tested and the results are respectively presented in Figures 6.33, 6.35, 6.37, Tables 6.11, 6.12, 6.13 and the graphs representing the length versus the aggregated constraint are shown in Figures 6.34, 6.36, 6.38.



(a) 2 lines contour (b) Initialization 1

(c) Initialization 2 (d) Initialization 3

Figure 6.33: Initialization and final results depending on the initialization for the zero hole object shown in Figure 3.6 (titanium).

Case	$L_{\mathrm{F}}^{\mathrm{ini}}$ (m)	$L_{\mathrm{F}}^{\mathrm{fin}}$ (m)	$\overline{C}_{\phi}^{\mathrm{ini}}$	$\overline{C}_{\phi}^{\mathrm{fin}}$	$\overline{C}_{\mathrm{M,D_S}}^{\mathrm{ini}}$	$\overline{C}_{\mathrm{M,D_S}}^{\mathrm{fin}}$	$\overline{C}_{\mathrm{M,D\setminus D_S}}^{\mathrm{ini}}$	$\overline{C}_{\mathrm{M,D\setminus D_S}}^{\mathrm{fin}}$
2 lines contour	4.826e−3	1.159e−2	1.60e−1	6.87e−3	5.23e−5	1.34e−3	0.00	1.95e−3
initialization 1	4.687e−3	1.217e−2	1.28e−1	2.21e−3	7.79e−6	8.89e−5	0.00	1.55e−2
initialization 2	4.555e−3	1.187e−2	1.31e−1	1.24e−3	1.96e−6	1.07e−4	0.00	4.32e−3
initialization 3	4.511e−3	1.193e−2	1.36e−1	1.43e−3	9.17e−6	7.28e−5	0.00	1.02e−3

Table 6.11: Comparison of the cost and the constraints of the final results for the zero hole object (Figure 3.6) (titanium).

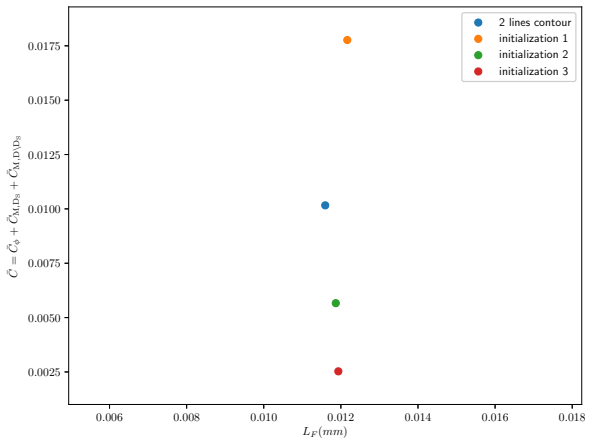
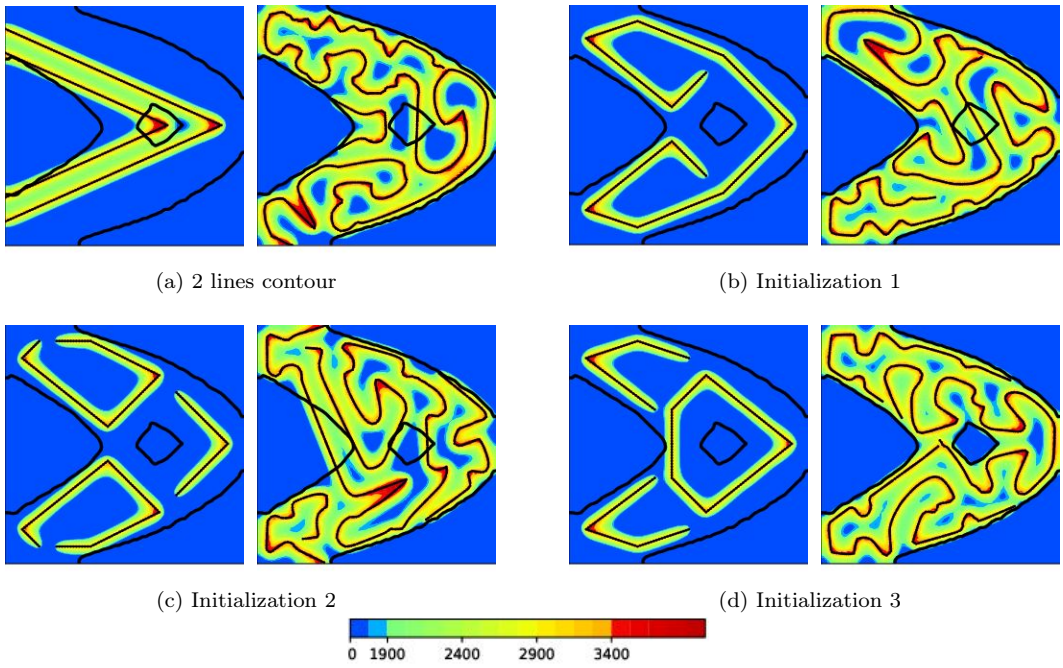


Figure 6.34: Constraint $\overline{C} = \overline{C}_{\phi} + \overline{C}_{\mathrm{M,D_S}} + \overline{C}_{\mathrm{M,D\setminus D_S}}$ with respect to the length L_{F} for the final results for the zero hole object (Figure 3.6) (titanium).

The path adapts well to the object, similarly to the aluminium case. However, the accuracy of this adaptation remains dependent on the initialization. Indeed, if the final path lengths are very similar, the fulfillment of the constraints is not: the results are far more precise on the second line of Figure 6.33 (Initialization 2 and Initialization 3) than on the first.



(a) 2 lines contour (b) Initialization 1

(c) Initialization 2 (d) Initialization 3

Figure 6.35: Initialization and final results depending on the initialization for the one hole object shown in Figure 3.7 (titanium).

Case	$L_{\mathrm{F}}^{\mathrm{ini}}$ (m)	$L_{\mathrm{F}}^{\mathrm{fin}}$ (m)	$\overline{C}_{\phi}^{\mathrm{ini}}$	$\overline{C}_{\phi}^{\mathrm{fin}}$	$\overline{C}_{\mathrm{M,D_S}}^{\mathrm{ini}}$	$\overline{C}_{\mathrm{M,D_S}}^{\mathrm{fin}}$	$\overline{C}_{\mathrm{M,D\setminus D_S}}^{\mathrm{ini}}$	$\overline{C}_{\mathrm{M,D\setminus D_S}}^{\mathrm{fin}}$
2 lines contour	4.826e−3	9.684e−3	1.42e−1	2.88e−3	1.25e−5	2.96e−4	2.06e−2	5.35e−3
initialization 1	4.687e−3	9.848e−3	9.04e−2	1.51e−3	1.00e−5	1.64e−4	0.00	7.86e−3
initialization 2	4.555e−3	9.961e−3	9.72e−2	1.72e−3	2.51e−6	2.66e−4	3.96e−7	1.63e−2
initialization 3	4.511e−3	9.653e−3	9.61e−2	4.50e−4	1.18e−5	3.17e−5	2.06e−7	1.63e−4

Table 6.12: Comparison of the cost and the constraints of the final results for the one hole object (Figure 3.7) (titanium).

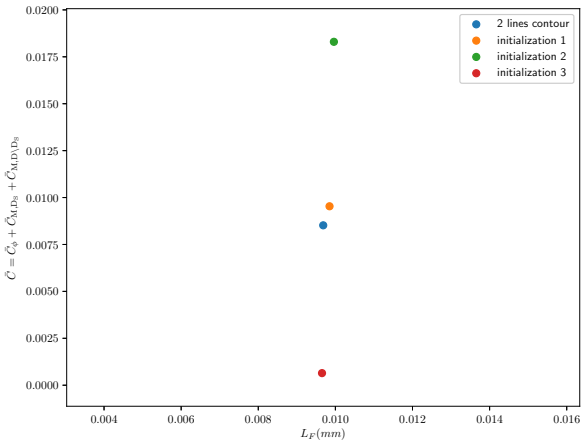
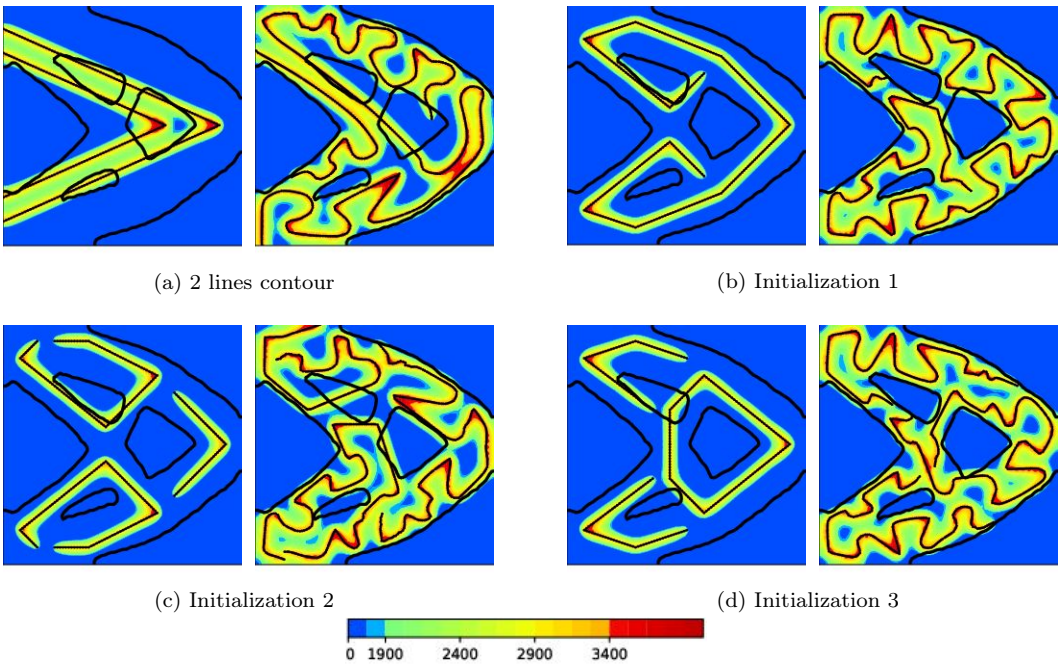


Figure 6.36: Constraint $\overline{C} = \overline{C}_\phi + \overline{C}_{\mathrm{M,D_S}} + \overline{C}_{\mathrm{M,D\backslash D_S}}$ with respect to the length L_F for the final results for the one hole object (Figure 3.7) (titanium).

The analysis for this second object is the same than in the first object with the different points forming a straight line in Figure 6.36: if the final length is the same, the constraints are not always well fitted and the algorithm seems to struggle to correctly avoid the holes.



(a) 2 lines contour

(b) Initialization 1

(c) Initialization 2

(d) Initialization 3

Figure 6.37: Initialization and final results depending on the initialization for the three holes object shown in Figure 3.8 (titanium).

Case	$L_{\mathrm{F}}^{\mathrm{ini}}$ (m)	$L_{\mathrm{F}}^{\mathrm{fin}}$ (m)	$\overline{C}_{\phi}^{\mathrm{ini}}$	$\overline{C}_{\phi}^{\mathrm{fin}}$	$\overline{C}_{\mathrm{M,D_S}}^{\mathrm{ini}}$	$\overline{C}_{\mathrm{M,D_S}}^{\mathrm{fin}}$	$\overline{C}_{\mathrm{M,D\setminus D_S}}^{\mathrm{ini}}$	$\overline{C}_{\mathrm{M,D\setminus D_S}}^{\mathrm{fin}}$
2 lines contour	4.826e−3	9.720e−3	1.65e−1	4.91e−3	1.31e−5	2.02e−4	5.16e−2	1.88e−2
initialization 1	4.687e−3	9.580e−3	8.38e−2	1.62e−3	1.05e−5	1.15e−4	2.79e−3	1.88e−3
initialization 2	4.555e−3	9.508e−3	8.87e−2	3.24e−3	2.63e−6	1.99e−4	2.82e−3	9.62e−3
initialization 3	4.511e−3	9.434e−3	9.61e−2	2.17e−3	1.23e−5	1.43e−4	5.64e−3	1.06e−3

Table 6.13: Comparison of the cost and the constraints of the final results for the three holes object (Figure 3.8) (titanium).

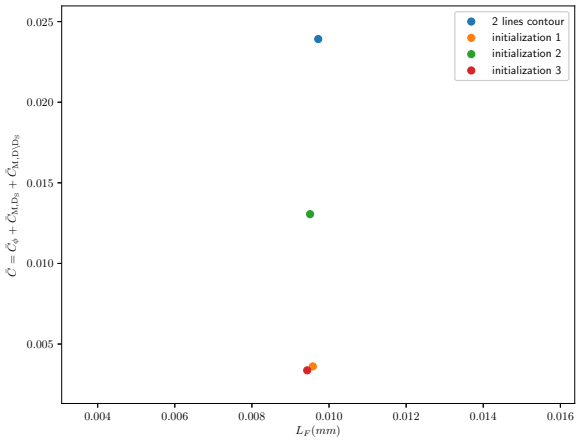


Figure 6.38: Constraint $\overline{C} = \overline{C}_{\phi} + \overline{C}_{\mathrm{M,D_S}} + \overline{C}_{\mathrm{M,D\setminus D_S}}$ with respect to the length L_{F} for the final results for the three holes object (Figure 3.8) (titanium).

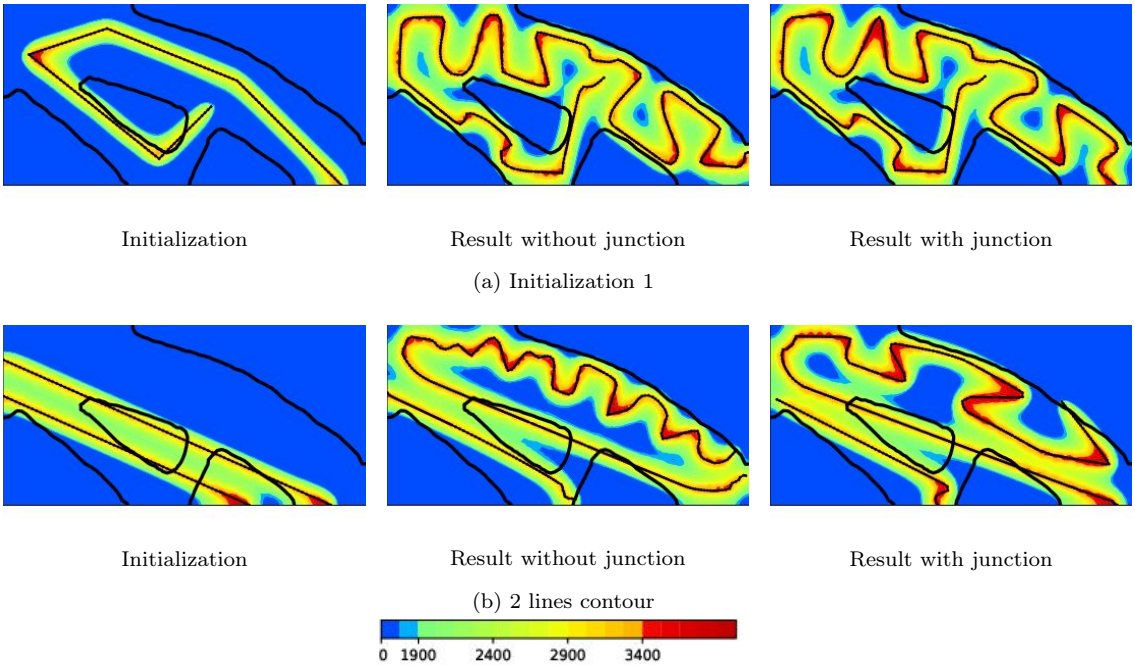
The final object once again leads to the same conclusions. This third object however also points out that a lower conductivity could be an advantage. Indeed, it seems a lot easier for the path to respect the geometry for the titanium than for the aluminium since the low conductivity facilitates the building of the thin bars.

To remedy once again to the non symmetry of the results, the same tests than for the aluminium (see Figure 6.30) are run. The results are given by Figure 6.39 and Table 6.14 (for the whole domain D, results comparable to Table 6.13)

Case	$L_{\mathrm{F}}^{\mathrm{ini}}$ (m)	$L_{\mathrm{F}}^{\mathrm{fin}}$ (m)	$\overline{C}_{\phi}^{\mathrm{ini}}$	$\overline{C}_{\phi}^{\mathrm{fin}}$	$\overline{C}_{\mathrm{M,D_S}}^{\mathrm{ini}}$	$\overline{C}_{\mathrm{M,D_S}}^{\mathrm{fin}}$	$\overline{C}_{\mathrm{M,D\setminus D_S}}^{\mathrm{ini}}$	$\overline{C}_{\mathrm{M,D\setminus D_S}}^{\mathrm{fin}}$
Initialization 1								
no junction	4.687e−3	9.527e−3	8.55e−2	1.54e−3	1.05e−5	6.33e−5	2.33e−1	9.45e−2
with junction	4.687e−3	9.376e−3	8.55e−2	2.22e−3	1.05e−5	1.65e−4	2.33e−1	1.27e−1
2 lines contour								
no junction	4.826e−3	9.019e−3	1.64e−1	8.55e−3	1.31e−5	8.01e−5	2.48	1.06
with junction	4.826e−3	9.622e−3	1.64e−1	1.09e−2	1.31e−5	3.53e−4	2.48	1.48

Table 6.14: Comparison of the cost and the constraints of the final results for the three holes object with symmetry (titanium).

It appears in the titanium case that the test with one connected component leads to better results, which actually corroborates the results found without symmetry.



(a) Initialization 1

(b) 2 lines contour

Figure 6.39: Final results in the titanium case depending on the initialization for the three holes object with symmetry

6.8 Conclusion

The numerical applications resulting from the optimization algorithm developed in the steady state context give interesting results. In the aluminium case first, the path adapts well to the object to build, with a decrease of the constraints until fulfillment and a length adaptation similar from one test to another. Yet, if the final values are similar, the initialization and algorithm choices seem to deeply impact the final shape of the path. The model and constraints provide a non convex optimization problem and thus the minimum is not unique. Finally, the building of complex geometries could still be improved. We could imagine for example choosing different optimization algorithms to better balance the constraints. We could also modify the constraints formulation: for example, further work could be done on changing the maximum temperature out of the domain constraint into a projection on the geometry to build.

If the optimization in the titanium context seems more complicated, the results confirm the observations made for the aluminium tests: the path adapts to the shape and, even if it may not fully satisfy the constraints at convergence, it decreases them. The final path shapes, widely varying from one test to another, lead to similar final values. This second physical test case highlights the difficulty of an even temperature spreading when the material conductivity is low. Indeed, the points at which the source is applied get very hot compared to their neighbors. This emphasizes the interest of the maximum temperature constraint, that enforces the path to remain smooth. This constraint should thus reduce the number of local minima. However, on the other hand, decreasing the temperature requires increasing the path length to fulfill the phase constraint thus increasing the final number of local minima. These conclusions should be further tested, especially in the transient context, which remain part of perspectives.

CHAPTER 7

PATH OPTIMIZATION IN THE TRANSIENT CONTEXT

Contents

7.1 Introduction . . . 145
7.2 Path optimization based on an angle parametrization . . . 145
7.2.1 Problem settings . . . 146
7.2.2 Theoretical descent in a continuous setting . . . 147
7.2.3 Numerical descent direction . . . 148
7.2.4 Algorithmic details . . . 151
7.2.5 Numerical results . . . 152
7.2.6 Conclusion . . . 157
7.3 Path optimization based on a points parametrization . . . 158
7.3.1 Descent direction computation . . . 159
7.3.2 Numerical Augmented Lagrangian algorithm . . . 162
7.3.3 First results . . . 163
7.3.4 Impact of the numerical choices . . . 165
7.3.5 Aluminium material test case . . . 170
7.3.6 Titanium material test case . . . 173
7.3.7 Conclusion . . . 177
7.4 Conclusion . . . 178

7.1 Introduction

In chapter 6, the optimization has been set in the steady state context. In addition to encouraging results, it has highlighted the importance of optimization parameters such as the path discretization and the algorithm chosen, and of the physical coefficients. Aware of these difficulties, this chapter proposes a numerical algorithm for path optimization in the transient context.

If shape optimization is really convenient in the steady state case, it cannot be used anymore. Indeed, the introduction of time dependent geometric variables prevent the use of classic shape differentiation formula. A more convenient approach would be to use optimal control theory. Thus, a direct optimization with respect to the discretization variables must be applied, leading to two different experiments. First, Section 7.2 focuses on the angle based discretization described in Section 5.4. In these settings, a differentiation of the objective function and constraints is given, as well as an optimization algorithm. Numerical results are finally presented. Section 7.3 corresponds to the point based discretization proposed in Section 5.5, supplemented by the differentiation of the different functions and the design of an optimization algorithm. Numerical results are finally provided and an analysis of the impact of the optimization choices on the final results is conducted.

7.2 Path optimization based on an angle parametrization

The encouraging results in the simplified steady state model allow for the adaptation of the gradient descent algorithm to the transient case. In this new context, the time dependence induces a hierarchy between the different points of the path, which must be translated into the sensitivity computation. Thus, the shape optimization used in the steady state case cannot be kept anymore and parametric optimization is set up. Chapter 5 described two different discretization methods (see Section 5.4 and 5.5). The angle based approach is considered in this section (Section 5.4).

7.2.1 Problem settings

The objective of this work is to look for the shortest path. The optimization problem thus belongs to optimal control involving a partial differential equation. If there exist well developed theories about both the optimal control of ordinary differential equations [90, 182] and optimal control of partial differential equations [111, 125], only a few focus on mixing both [56, 99, 196, 197]. Following these works, the path is defined through its tangent direction. This latter is usually controlled by the path curvature but, since no information is given about the first point nor the first direction, we choose here to control directly the tangent direction itself (or the velocity vector which norm is constant) thus reducing the number of initial condition unknowns.
In this new context, and following [196, 197], the path Γ is fully described by the angle $\mathfrak{a} \in \mathcal{A} = L^2((0,t_{\mathrm{F}}),\mathbb{R})$, the starting point $\tilde{u} \in D$ and the final time $t_{\mathrm{F}} \in (0, t_{\mathrm{Final}})$. These three variables constitute the new optimization parameters leading to an effective optimization problem:

$$\min_{t_{\mathrm{F}},\mathfrak{a},\tilde{u}} = t_{\mathrm{F}} \quad \text{such that} \quad \begin{cases} C_\phi = \displaystyle\int_{D_S} \left[(y_\phi - N_{\mathfrak{p}})^+\right]^2 dx = 0, \qquad N_{\mathfrak{p}} = \left(\frac{1}{t_{\mathrm{F}}}\displaystyle\int_0^{t_{\mathrm{F}}} |y|^{\mathfrak{p}} dt\right)^{1/\mathfrak{p}} \\ C_{\mathrm{M,D_S}} = \dfrac{1}{t_{\mathrm{F}}}\displaystyle\int_{D_S}\int_0^{t_{\mathrm{F}}} \left[(y - y_{\mathrm{M,D_S}})^+\right]^2 dtdx = 0, \\ C_{\mathrm{M,D\backslash D_S}} = \dfrac{1}{t_{\mathrm{F}}}\displaystyle\int_{D\backslash D_S}\int_0^{t_{\mathrm{F}}} \left[(y - y_{\mathrm{M,D\backslash D_S}})^+\right]^2 dtdx = 0, \end{cases} \tag{7.2.1}$$

with $y \in L^2([0,t_{\mathrm{F}}], H^1(D,\mathbb{R}^2)) \cap C^0\left([0,t_{\mathrm{F}}], L^2(D)\right)$ solution of

$$\begin{cases} \rho c_p \partial_t y(t,x) - \nabla\cdot(\lambda\nabla y(t,x)) + \beta\left(y(t,x) - y_{\mathrm{ini}}\right) = q(t,x) & \text{in } (0,t_{\mathrm{F}})\times D, \\ \lambda\partial_n y(t,x) = 0 & \text{on } (0,t_{\mathrm{F}})\times\partial D, \\ y(0,x) = y_{\mathrm{ini}}(x) & \text{in } D, \end{cases} \tag{7.2.2}$$

where $y_{\mathrm{ini}} \in L^2(D)$ [10] and the source term given by

$$q(t,x) = P\exp\left(-\frac{(x-u(t))^2}{r^2}\right), \tag{7.2.3}$$

where $u \in H^1([0,t_{\mathrm{Final}}],\mathbb{R})$ solution of the ordinary equation

$$\begin{cases} \dot{u}(t) = V\tau\left(\mathfrak{a}(t)\right), \quad 0 \le t \le t_{\mathrm{F}}, \\ u(0) = \tilde{u}. \end{cases} \tag{7.2.4}$$

The tangent is related to the angle through, $\forall 0 \le t \le t_{\mathrm{F}}$,

$$\tau(t) = \tau\left(\mathfrak{a}(t)\right) = \left(\cos\left(\mathfrak{a}(t)\right), \sin\left(\mathfrak{a}(t)\right)\right). \tag{7.2.5}$$

Note that in the transient case, not only the phase and maximum constraints are not aggregated but the maximum temperature constraint is split into two different ones, respectively focusing on inside and outside the domain to build. This complicates the choice of the optimization coefficients and especially the multiplier and penalization in the Augmented Lagrangian method. On the other hand, this can improve the accuracy of the updates.

To deal with the constraints, an Augmented Lagrangian method is used and a penalized function is introduced:

$$\mathcal{L}_{\mathrm{ALM}} : \begin{cases} \mathbb{R}^+\times\mathcal{A}\times D & \to & \mathbb{R} \\ (t_{\mathrm{F}},\mathfrak{a},\tilde{u}) & \mapsto & t_{\mathrm{F}} + l_\phi C_\phi + \dfrac{\mu_\phi}{2}C_\phi^2 + l_{\mathrm{M,s}}C_{\mathrm{M,D_S}} + \dfrac{\mu_{\mathrm{M,s}}}{2}C_{\mathrm{M,D_S}}^2 + l_{\mathrm{M,e}}C_{\mathrm{M,D\backslash D_S}} + \dfrac{\mu_{\mathrm{M,e}}}{2}C_{\mathrm{M,D\backslash D_S}}^2. \end{cases} \tag{7.2.6}$$

Remark 7.1. As stated in Chapter 4, the phase constraint is an approximation. Indeed, the maximum over time is approximated by a $L^{\mathfrak{p}}-$norm. Since for any $\mathfrak{p} \le 1$ and for any function $f \in L^{\mathfrak{p}}([0,t_{\mathrm{F}}],\mathbb{R})$, $\left(\frac{1}{t_{\mathrm{F}}}\int_0^{t_{\mathrm{F}}} |f|^\infty dt\right)^{1/\mathfrak{p}} \le \|f\|_{L^\infty([0,t_{\mathrm{F}}],\mathbb{R})}$ the approximation of the phase constraint will be greater than the real phase constraint in the numerical applications.

Remark 7.2. Unlike the steady state case, the different functions are not normalized in the Augmented Lagrangian function $\mathcal{L}_{\text{ALM}}$.

7.2.2 Theoretical descent in a continuous setting

In the continuous settings, the update of the dual step is given by:

$$\begin{cases} l_{\phi}^{k+1} = l_{\phi}^{k} + \mu_{\phi} C_{\phi}^{k}, \\ l_{\text{M,s}}^{k+1} = l_{\text{M,s}}^{k} + \mu_{\text{M,s}} C_{\text{M,D}_\text{S}}^{k}, \\ l_{\text{M,e}}^{k+1} = l_{\text{M,e}}^{k} + \mu_{\text{M,e}} C_{\text{M,D}\setminus\text{D}_\text{S}}^{k}. \end{cases} \tag{7.2.7}$$

As for the differentiation with respect to the optimization variables, it is given by Proposition7.1

Proposition 7.1. *The derivatives of the function $\mathcal{L}_{\text{ALM}}$ with respect to $t_{\text{F}} \in \mathbb{R}^+$, $\mathfrak{a} \in \mathcal{A}$ and $\tilde{u} \in D$ are, $\forall \delta\mathfrak{a} \in \mathcal{A}$ angle differentiation direction, $\forall \delta\tilde{u} \in \mathbb{R}^2$ first point differentiation direction, and $\forall \delta t \in \mathbb{R}$,*

$$\begin{aligned} &D_{\mathfrak{a}}\mathcal{L}_{\text{ALM}}(t_{\text{F}}, \mathfrak{a}, \tilde{u})(\delta\mathfrak{a}) = -\int_0^{t_{\text{F}}} V\tau'(\mathfrak{a}(t)) \cdot w(t)\delta\mathfrak{a}(t)dt, \\ &D_{\tilde{u}}\mathcal{L}_{\text{ALM}}(t_{\text{F}}, \mathfrak{a}, \tilde{u})(\delta\tilde{u}) = -w(0) \cdot \delta\tilde{u}, \\ &D_{t_{\text{F}}}\mathcal{L}_{\text{ALM}}(t_{\text{F}}, \mathfrak{a}, \tilde{u}) = 1 \\ &\qquad + (l_{\phi} + \mu_{\phi} C_{\phi}) \frac{2}{t_{\text{F}}\mathfrak{p}} \int_{D_S} (y_{\phi} - N_{\mathfrak{p}}(t_{\text{F}}, y))^{+} N_{\mathfrak{p}}(t_{\text{F}}, y)^{1-\mathfrak{p}} \left(N_{\mathfrak{p}}(t_F, y)^{\mathfrak{p}} - y(t_F)^{\mathfrak{p}}\right) dx \\ &\qquad + \frac{l_{\text{M,s}} + \mu_{\text{M,s}} C_{\text{M,D}_\text{S}}}{t_{\text{F}}} \left(\int_{D_S} \left[(y(t_{\text{F}}, x) - y_{\text{M}})^{+} \right]^2 dx - C_{\text{M,D}_\text{S}} \right) \\ &\qquad + \frac{l_{\text{M,e}} + \mu_{\text{M,e}} C_{\text{M,D}\setminus\text{D}_\text{S}}}{t_{\text{F}}} \left(\int_{D\setminus D_S} \left[(y(t_{\text{F}}, x) - y_{\text{M}})^{+} \right]^2 dx - C_{\text{M,D}\setminus\text{D}_\text{S}} \right). \end{aligned} \tag{7.2.8}$$

In the above formulas, $w \in H^1([0, t_{\text{Final}}], \mathbb{R})$ is the adjoint for the ODE (7.2.4), solution of

$$\begin{cases} \dot{w}(t) = \dfrac{2}{r^2} \displaystyle\int_D q(u(t), x) p(t, x) \left(u(t) - x\right) dx \quad t \in (0, t_{\text{F}}), \\ w(t_{\text{F}}) = 0, \end{cases} \tag{7.2.9}$$

and $p \in L^2([0, t_{\text{F}}], H^1(D, \mathbb{R}^2))$ is the adjoint of the heat equation, solution of

$$\begin{cases} -\rho c_p \partial_t p - \nabla \cdot (\lambda \nabla p) + \beta p = \dfrac{2\left(l_{\phi} + \mu_{\phi} C_{\phi}\right)}{t_{\text{F}}} (y_{\phi} - N_{\mathfrak{p}}(t_{\text{F}}, y))^{+} N_{\mathfrak{p}}(t_{\text{F}}, y)^{1-\mathfrak{p}} y^{\mathfrak{p}-1} \mathbb{1}_{D_S} \\ \qquad\qquad - \dfrac{2\left(l_{\text{M,s}} + \mu_{\text{M,s}} C_{\text{M,D}_\text{S}}\right)}{t_{\text{F}}} \left(y - y_{\text{M,D}_\text{S}}\right)^{+} \mathbb{1}_{D_S} \qquad \text{in } (0, t_{\text{F}}) \times D, \\ \qquad\qquad - \dfrac{2\left(l_{\text{M,e}} + \mu_{\text{M,e}} C_{\text{M,D}\setminus\text{D}_\text{S}}\right)}{t_{\text{F}}} \left(y - y_{\text{M,D}\setminus\text{D}_\text{S}}\right)^{+} \mathbb{1}_{D\setminus D_S} \\ \lambda \partial_n p = 0 \qquad \text{on } (0, t_{\text{F}}) \times \partial D, \\ p(t_{\text{F}}, x) = 0 \qquad \text{in } D. \end{cases} \tag{7.2.10}$$

PROOF.
As in the steady state context, we rely on the method of Céa to compute the derivatives by introducing an adjoint [46]. A Lagrangian function $\mathcal{L} : [0, t_{\text{Final}}] \times \mathcal{A} \times D \times H^1([0, t_{\text{Final}}], \mathbb{R}) \times H^1([0, t_{\text{Final}}], \mathbb{R}) \times L^2([0, t_{\text{F}}], H^1(D, \mathbb{R}^2)) \times L^2([0, t_{\text{F}}], H^1(D, \mathbb{R}^2)) \to \mathbb{R}$, involving the variational formulations of both the heat and trajectory equations, is introduced:

$$\mathcal{L}\left(t_{\text{F}}, \mathfrak{a}, \tilde{u}, v, w, \Phi, p\right) \quad = \mathcal{L}_{\text{ALM}}(t_{\text{F}}, \mathfrak{a}, \tilde{u}) + \tilde{\mathcal{L}}(t_{\text{F}}, \mathfrak{a}, \tilde{u}, v, w, \Phi, p)$$

with

$$\begin{aligned}\tilde{\mathcal{L}}\left(t_{\mathrm{F}},\mathfrak{a},\tilde{u},v,w,\Phi,p\right)=&\int_0^{t_{\mathrm{F}}}\left(\dot{v}-V\tau\left(\mathfrak{a}\left(t\right)\right)\right)\cdot w(t)dt+\left(v(0)-\tilde{u}\right)\cdot w(0)\\&+\int_0^{t_{\mathrm{F}}}\int_D\left(\left(\rho c_p\partial_t\Phi+\beta\Phi-\beta y_{\mathrm{ini}}-q\left(v(t)\right)\right)p+\lambda\nabla\Phi\cdot\nabla p\right)dxdt\\&+\int_D\rho c_p\left(\Phi(0,x)-y_{\mathrm{ini}}(x)\right)p(0,x)dx.\end{aligned}$$

Then, $\forall\left(t_{\mathrm{F}},\mathfrak{a},\tilde{u}\right)\in[0,t_{\mathrm{Final}}]\times\mathcal{A}\times D$, $\forall(w,p)\in H^1([0,t_{\mathrm{Final}}],\mathbb{R})\times L^2([0,t_{\mathrm{F}}],H^1(D,\mathbb{R}^2))$, with $u\in H^1([0,t_{\mathrm{Final}}],\mathbb{R})$ and $y\in L^2([0,t_{\mathrm{F}}],H^1(D,\mathbb{R}^2))$ respectively solutions of (7.2.4) and (7.2.2):

$$\tilde{\mathcal{L}}\left(t_{\mathrm{F}},\mathfrak{a},\tilde{u},u,w,y,p\right)=0\qquad\Longrightarrow\qquad\mathcal{L}\left(t_{\mathrm{F}},\mathfrak{a},\tilde{u},u,w,y,p\right)=\mathcal{L}_{\mathrm{ALM}}(t_{\mathrm{F}},\mathfrak{a},\tilde{u}).$$

Differentiating with respect to the control parameters $X=(t_{\mathrm{F}},\mathfrak{a},\tilde{u})\in[0,T_F]\times\mathcal{A}\times D$, for any differentiation direction δX,

$$\begin{aligned}\frac{d\mathcal{L}}{dX}(X,u,w,T,p)(\delta X)=&\,\partial_X\mathcal{L}(X,u,w,T,p)(\delta X)\\&+<\partial_v\mathcal{L}(X,u,w,T,p),\partial_X u\left(\delta X\right)>+<\partial_\Phi\mathcal{L}(X,u,w,T,p),\partial_X T\left(\delta X\right)>.\end{aligned}\tag{7.2.11}$$

Setting to 0 the derivatives of the Lagragian function $\mathcal{L}$ with respect to v and Φ, evaluated at $\Phi=y$ (solution of (7.2.2)) and $v=u$ (solution of (7.2.4)), comes down to solving the adjoint equations (7.2.9) and (7.2.10). Particularizing w to be solution of (7.2.9) and p solution of (7.2.10), the differentiation finally results in

$$d_X f_{\mathrm{ALM}}(X)=\partial_X\mathcal{L}\left(X,u,w,y,p\right).\tag{7.2.12}$$

Classically, (7.2.11) yields the derivatives with respect to $\mathfrak{a}$ and $\tilde{u}$. As for the differentiation with respect to the final time t_{F}, one gets:

$$\begin{aligned}&\partial_{t_{\mathrm{F}}}\mathcal{L}\left(t_{\mathrm{F}},\mathfrak{a},\tilde{u},u,w,y,p\right)=1\\&+\left(l_\phi+\mu_\phi C_\phi\right)\int_{D_S}-2\left(y_\phi-N_{\mathfrak{p}}(t_{\mathrm{F}},y)\right)^+N_{\mathfrak{p}}(t_{\mathrm{F}},y)^{1-\mathfrak{p}}\left(\frac{-1}{\mathfrak{p}t_{\mathrm{F}}^2}\int_0^{t_{\mathrm{F}}}|y|^{\mathfrak{p}}dt+\frac{1}{\mathfrak{p}t_{\mathrm{F}}}y(t_{\mathrm{F}})^{\mathfrak{p}}\right)dx\\&+\frac{l_{\mathrm{M,s}}+\mu_{\mathrm{M,s}}C_{\mathrm{M,D_S}}}{t_{\mathrm{F}}}\left(\int_{D_S}\left[\left(y(t_{\mathrm{F}},x)-y_{\mathrm{M,D_S}}\right)^+\right]^2dx-C_{\mathrm{M,D_S}}\right)\\&+\frac{l_{\mathrm{M,e}}+\mu_{\mathrm{M,e}}C_{\mathrm{M,D\backslash D_S}}}{t_{\mathrm{F}}}\left(\int_{D\backslash D_S}\left[\left(y(t_{\mathrm{F}},x)-y_{\mathrm{M,D\backslash D_S}}\right)^+\right]^2dx-C_{\mathrm{M,D\backslash D_S}}\right)\\&+\left(\dot{u}(t_{\mathrm{F}})-V\tau\left(\mathfrak{a}\left(t_{\mathrm{F}}\right)\right)\right)\cdot w(t_{\mathrm{F}})\\&+\int_D\left(\left(\rho c_p\partial_t y(t_F,x)+\beta y(t_{\mathrm{F}},x)-q\left(u(t_{\mathrm{F}}),x\right)\right)p(t_{\mathrm{F}},x)+\lambda\nabla y(t_{\mathrm{F}},x)\cdot\nabla p(t_{\mathrm{F}},x)\right)dx.\end{aligned}\tag{7.2.13}$$

For smooth solutions in time, the variational formulation of (7.2.2) and (7.2.4) are satisfied at the final time t_{F}. Thus, the two last terms of (7.2.13) cancel and we get the result (7.2.8). □

7.2.3 Numerical descent direction

To discretize these equations and compute an effective numerical descent direction, the path is split into N_u equidistant points defining N_u-1 elements of size l. This corresponds to the discretization of the time by a sequence $\{0=t_0,...,t_{N_u}=t_{\mathrm{F}}\}$ and $\forall i\in[\![1,N_u]\!]$, $t_i=t_{i-1}+\Delta t$ $\left(\Delta t=\frac{l}{V}\right)$. To this time vector $(t_i)_{i\in[\![0,N_u]\!]}$ are associated an angle vector $(\mathfrak{a}_i)_{i\in[\![1,N_u-1]\!]}$, a point vector $(u_i)_{i\in[\![1,N_u]\!]}$ and a temperature vector $(y_i)_{i\in[\![0,N_u]\!]}$. The heat equation (7.2.2) is discretized with respect to time by an implicit Euler scheme whereas an Euler forward scheme is used for the trajectory equation (7.2.4):

$$\begin{cases}\dfrac{u_i-u_{i-1}}{\Delta t}=\tau\left(\mathfrak{a}_{i-1}\right) & \forall i\in\{2,..,N_u\}\\ u_1=\tilde{u} & .\end{cases}\tag{7.2.14}$$

and for $i \in \{1, .., N_u\}$,

$$\begin{cases} \rho c_p \dfrac{y_i(x) - y_{i-1}(x)}{\Delta t} - \nabla \cdot (\lambda \nabla y_i(x)) + \beta(y_i(x) - y_{\text{ini}}) = q\,(u_i, x,) & \forall x \in D \\ \lambda \partial_n y_i(x) = 0 & \forall x \in \partial D \\ y_0(x) = y_{\text{ini}}(x) & \forall x \in D, \end{cases} \tag{7.2.15}$$

The constraints are discretized through an implicit scheme corresponding to the heat equation scheme:

$$C_\phi = \int_{D_S} \left[(y_\phi - N_{\mathfrak{p}})^+ \right]^2 dx = \int_{D_S} \left[\left(y_\phi - \left(\frac{1}{t_{\text{F}}} \sum_{i=0}^{N_u - 1} \Delta t\, y_{i+1}^{\mathfrak{p}} \right)^{1/\mathfrak{p}} \right)^+ \right]^2 dx. \tag{7.2.16}$$

and

$$\begin{cases} C_{\text{M,D}_\text{S}} = \displaystyle\int_{D_S} \sum_{i=0}^{N_u - 1} \frac{\Delta t}{t_{\text{F}}} \left[(y_{i+1} - y_{\text{M,D}_\text{S}})^+ \right]^2 dx, \\ C_{\text{M,D}\setminus\text{D}_\text{S}} = \displaystyle\int_{D\setminus D_S} \sum_{i=0}^{N_u - 1} \frac{\Delta t}{t_{\text{F}}} \left[(y_{i+1} - y_{\text{M,D}\setminus\text{D}_\text{S}})^+ \right]^2 dx. \end{cases} \tag{7.2.17}$$

Remark 7.3. Note that the constraints could also be discretized by a "centered scheme way" as

$$C_\phi = \int_{D_S} \left[(y_\phi - N_{\mathfrak{p}})^+ \right]^2 dx = \int_{D_S} \left[\left(y_\phi - \left(\frac{1}{t_{\text{F}}} \sum_{i=1}^{N_u - 1} \Delta t \frac{y_i^{\mathfrak{p}} + y_{i+1}^{\mathfrak{p}}}{2} \right)^{1/\mathfrak{p}} \right)^+ \right]^2 dx. \tag{7.2.18}$$

and

$$C_{\text{M}} = \int_{D} \sum_{i=1}^{N_u - 1} \frac{\Delta t}{t_{\text{F}}} \left[\left(\frac{y_i + y_{i+1}}{2} - y_{\text{M}} \right)^+ \right]^2 dx. \tag{7.2.19}$$

In numerical applications, to compute the state variables u and y, the ODE (7.2.14) is first solved for index i followed by the resolution of the PDE (7.2.15) also for the index i.

The differentiation process described in the continuous case (Section 7.2.2) can be also be applied in the discrete case (see [16] for a similar approach in shape optimization). The corresponding adjoint equations are backward. The discrete equations for the heat adjoint $p = (p_0, ..., p_{N_u})$ are, $\forall i \in 0, ... N_u - 1$:

$$\begin{cases} \rho c_p \dfrac{p_i - p_{i+1}}{\Delta t} - \nabla \cdot (\lambda \nabla p_i) + \beta p_i = \dfrac{2\,(l_\phi + cC_\phi)}{t_{\text{F}}} (y_\phi - N_{\mathfrak{p}}(t_{\text{F}}, y))^+ N_{\mathfrak{p}}(t_{\text{F}}, y)^{1-\mathfrak{p}} y_i^{\mathfrak{p}-1} \mathbb{1}_{D_S} \\ \qquad - \dfrac{2\,(l_{\text{M,s}} + \mu_{\text{M,s}} C_{\text{M,D}_\text{S}})}{t_{\text{F}}} (y_i - y_{\text{M,D}_\text{S}})^+ \mathbb{1}_{D_S} \qquad \text{in } D \\ \qquad - \dfrac{2\,(l_{\text{M,e}} + \mu_{\text{M,e}} C_{\text{M,D}\setminus\text{D}_\text{S}})}{t_{\text{F}}} (y_i - y_{\text{M,D}\setminus\text{D}_\text{S}})^+ \mathbb{1}_{D\setminus D_S} \\ \lambda \partial_n p_i = 0 \qquad \text{on } \partial D. \end{cases} \tag{7.2.20}$$

$$\begin{cases} \dfrac{\rho c_p}{\Delta t} p_{N_u} - \nabla \cdot (\lambda \nabla p_{N_u}) + \beta p_{N_u} = \dfrac{2\,(l_\phi + cC_\phi)}{t_{\text{F}}} (y_\phi - N_{\mathfrak{p}}(t_{\text{F}}, y))^+ N_{\mathfrak{p}}(t_{\text{F}}, y)^{1-\mathfrak{p}} y_{N_u}^{\mathfrak{p}-1} \mathbb{1}_{D_S} \\ \qquad - \dfrac{2\,(l_{\text{M,s}} + \mu_{\text{M,s}} C_{\text{M,D}_\text{S}})}{t_{\text{F}}} (y_{N_u} - y_{\text{M,D}_\text{S}})^+ \mathbb{1}_{D_S} \qquad \text{in } D \\ \qquad - \dfrac{2\,(l_{\text{M,e}} + \mu_{\text{M,e}} C_{\text{M,D}\setminus\text{D}_\text{S}})}{t_{\text{F}}} (y_{N_u} - y_{\text{M,D}\setminus\text{D}_\text{S}})^+ \mathbb{1}_{D\setminus D_S} \\ \lambda \partial_n p_{N_u} = 0 \qquad \text{on } \partial D. \end{cases}$$

The discrete equations for the ODE adjoint $w = (w_1, ..., w_{N_u - 1})$ are $\forall i \in \{2, .., N_u - 1\}$

$$\begin{cases} \dfrac{w_i - w_{i-1}}{\Delta t} = \dfrac{2}{r^2} \displaystyle\int_D q(u_i) p_i (u_i - x) dx, \\ \dfrac{w_{N_u - 1}}{\Delta t} = -\dfrac{2}{r^2} \displaystyle\int_D q(u_{N_u}) p_{N_u} (u_{N_u} - x) dx. \end{cases} \tag{7.2.21}$$

The discrete derivatives with respect to the angle and first point are thus classically obtained. As for the differentiation with respect to the final time, the formula obtained in the continuous settings is kept.

Remark 7.4. Note that, in the discrete context, the "final time" condition on the adjoints is not equal to 0 anymore.

Update of the angle α

The discrete derivative with respect to the angle $\mathfrak{a}$ is given by

$$D_{\mathfrak{a}}\mathcal{L}_{\mathrm{ALM}}(d\mathfrak{a}) = \sum_{i=1}^{N_u-1} -\Delta t\tau\left(\mathfrak{a}_i\right) w_i d\mathfrak{a}_i. \tag{7.2.22}$$

To compute the gradient from this derivative, different choices exist. The first one consists into not regularizing the angle and using the scalar product related to the Hilbert space $\mathcal{A}$. In the discrete formulation, this amounts to choosing, $\forall i \in [\![1, N_u - 1]\!]$,

$$\left(\nabla_{\mathfrak{a}}\mathcal{L}_{\mathrm{ALM}}\right)_i = -\tau\left(\mathfrak{a}_i\right) w_i. \tag{7.2.23}$$

We propose here a second choice related to a H^1 regularization, which consists in solving the following minimization problem:

$$\nabla_{\mathfrak{a}}\mathcal{L}_{\mathrm{ALM}} = \underset{Q\in\mathbb{R}^{N_u-1}}{\arg\min} \sum_{i=1}^{N_u-2} \frac{\Delta t}{2}\left(\nu_\Gamma^2\left(\frac{\mathfrak{a}_{i+1}-\mathfrak{a}_i}{\Delta t}\right)^2 + \frac{\mathfrak{a}_i^2+\mathfrak{a}_{i+1}^2}{2}\right) - \Delta t\frac{\tau\left(\mathfrak{a}_i\right)w_i + \tau\left(\mathfrak{a}_{i+1}\right)w_{i+1}}{2}. \tag{7.2.24}$$

Of course, setting $\nu_\Gamma = 0$ amounts to applying (7.2.23).

Remark 7.5. Note that the H^1 formulation proposed actually corresponds to the discrete equation

$$\Delta t\nu_\Gamma^2\frac{-Q_{i+1}+2Q_i-Q_{i-1}}{\Delta t^2} + \Delta t Q_i = \tau\left(\mathfrak{a}_i\right)w_i \quad i \in [\![2, N_u-2]\!]. \tag{7.2.25}$$

Unlike the heat equation and the point equation, this formulation has been chosen centered to regularize "equally" each angle.

Following a gradient descent algorithm for the optimization variables, the angle is updated by, $\forall i \in \{1, ..., N_u - 1\}$

$$\mathfrak{a}_i^{k+1} = \mathfrak{a}_i^k - s_{\mathfrak{a}}^k\left(\nabla_{\mathfrak{a}}\mathcal{L}_{\mathrm{ALM}}\right)i,$$

with a positive step $s_{\mathfrak{a}}^k > 0$. To ensure the decrease of the objective function, the angles' variations must be kept small and the step $s_{\mathfrak{a}}^k$ is chosen so that

$$\|\mathfrak{a}^{k+1} - \mathfrak{a}^k\|_\infty = \mathfrak{a}_{ref}^k. \tag{7.2.26}$$

The reference angle $\mathfrak{a}_{ref}^k$ is initialized to $\mathfrak{a}_{ref}^0 = 3°$. At each iteration k, if the objective function satisfies $\mathcal{L}_{\mathrm{ALM}}(\mathfrak{a}^{k+1}) < tol^k * \mathcal{L}_{\mathrm{ALM}}(\mathfrak{a}^k)$, then,

$$\mathfrak{a}_{ref}^{k+1} = \min\left(\mathfrak{a}_{ref}^0, 1.2\mathfrak{a}_{ref}^k\right),$$

else,

$$\mathfrak{a}_{ref}^{k+1} = 0.6\mathfrak{a}_{ref}^k,$$

where $tol^k \geq 1$ is a tolerance which is initialized as 2 and multiplied by 0.9 every 50 iterations. The reference angle plays here the role of the step coefficient.

Update of the starting point $\tilde{u}$

Once the angles have been updated, the point equation is solved: from the point $\tilde{u}^k$ and following the angle vector $\left(\mathfrak{a}_i^{k+1}\right)$, the points $\left(\tilde{u}^k\right)_i$ are computed and the average displacement of those nodes corresponds to

$$\delta u^k = \frac{1}{N_u-1}\sum_{i=1}^{N_u-1}|\tilde{u}_i^k - u_i^k|,$$

where $|.|$ is the Euclidian distance in $\mathbb{R}^2$. The gradient with respect to the first point is $\nabla_{\tilde{u}}\mathcal{L}_{\text{ALM}} = w_1$. At each iteration k,

$$\tilde{u}^{k+1} = \tilde{u}^k + s_{\tilde{u}}^k \delta u^k \frac{w_1}{\|w_1\|},$$

with a positive step $s_{\tilde{u}}^k > 0$. This descent step is updated as follows: at each iteration k, if the objective function satisfies $\mathcal{L}_{\text{ALM}}(\tilde{u}^{k+1}) < tol^k * \mathcal{L}_{\text{ALM}}(\tilde{u}^k)$ (same tolerance than for the angles),

$$s_{\tilde{u}}^{k+1} = \min\left(s_{\tilde{u}}^0, 1.2 s_{\tilde{u}}^k\right),$$

else,

$$s_{\tilde{u}}^{k+1} = 0.6 s_{\tilde{u}}^k.$$

This update of the starting point is not done at each iteration but only every 3 iterations.

Once the updated first point $\tilde{u}^{k+1}$ has been computed, the trajectory equation is once again solved (actually, translating the points $\tilde{u}$ in the direction $\left(\tilde{u}^{k+1} - \tilde{u}^k\right)$ is enough).

Update of the final time t_{F}

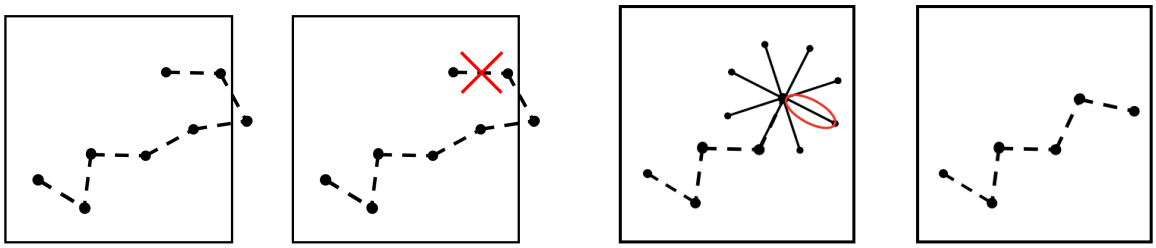
Recall that the final time t_{F} is not continuous anymore but a discrete variable linked to the number of segments of the path. We define a discrete descent step $s_{t_{\text{F}}}^k$ which is a positive integer number initialized at 5. This number $s_{t_{\text{F}}}^k$ determines the number of segments which are added or removed at the end of the discretized path for each iteration k. If the time derivative given by Proposition 7.1 is negative or if any nodal point of the line is out of the domain D, $s_{t_{\text{F}}}^k$ segments are removed. Else a straight line of $s_{t_{\text{F}}}^k$ segments (of size l) are added. The direction of the added segments can be chosen within 8 values which are in addition of $\mathfrak{a}_{N_u-1}^k$ (the direction of the last segment of the path at iteration n): 0°, 45°, 90°, 135°, 180°, 225°, 270°, 315°. The best one to optimize the problem is chosen (see Figure 7.1). The descent step $S_{t_{\text{F}}}^k$ is updated as follows: if after iteration n, the objective function satisfies $\mathcal{L}_{\text{ALM}}(\tilde{u}^{k+1}) < tol^k * \mathcal{L}_{\text{ALM}}(\tilde{u}^k)$ (same tolerance than for the angles), then

$$s_{t_{\text{F}}}^{k+1} = \min(s_{t_{\text{F}}}^k + 1, 5),$$

otherwise,

$$s_{t_{\text{F}}}^{k+1} = \max(s_{t_{\text{F}}}^k - 1, 0).$$

The final time t_{F} is not updated at each iteration but every 5 iterations. When the final time is updated, the angles and the starting point are kept unchanged.



(a) Removing a segment if one nodal point is out of the working domain

(b) Adding a segment

Figure 7.1: Final time t_{F} update process

7.2.4 Algorithmic details

For our numerical simulations, we use Algorithm 7.1. As in the steady state case, for simplicity, no stopping criterion has been implemented in this algorithm, except a prescribed maximal number of

iterations.
initialization of the line
resolution of the heat equation along the line and computation of the objective function and constraints
computation of the derivatives
for *each iteration* **do**
  update the tolerance
  update the line (angle, first point and final time modified depending on the iteration number)
  resolution of the heat equation, computation of the objective function and constraint
  if *improvement* $\left(\mathcal{L}_{\text{ALM}}^{k+1} < tol^k \mathcal{L}_{\text{ALM}}^k\right)$ **then**
    iteration accepted
    step coefficients increased $\left(\mathfrak{a}_{\text{ref}}^{k+1} = \min\left(\mathfrak{a}_{\text{ref}}^0, 1.2\mathfrak{a}_{\text{ref}}^k\right)\right.$, $\left.s_{\tilde{u}}^{k+1} = \min\left(s_{\tilde{u}}^0, 1.2 s_{\tilde{u}}^k\right)\right.$ if update of the starting point and $s_{t_{\text{F}}}^{k+1} = \min(s_{t_{\text{F}}}^k + 1, 5)$ if update of the final time)
    update of the variables
    computation of the derivatives
  end
  else
    iteration refused
    step coefficient refused $\left(\mathfrak{a}_{ref}^{k+1} = 0.6\mathfrak{a}_{ref}^k\right.$, $\left.s_{\tilde{u}}^{k+1} = 0.6 s_{\tilde{u}}^k\right.$ if update of the starting point and $\left.s_{t_{\text{F}}}^{k+1} = \max(s_{t_{\text{F}}}^k - 1, 0)\right.$ if update of the final time)
  end
end

Algorithm 7.1: Iterative algorithm to optimize the unsteady problem

7.2.5 Numerical results

The working domain for the transient case remains $D = [-0.7, 0.7] \times [-0.7, 0.7]$ (given in mm), with the physical mesh used in the calibration (Chapter 4) and described in Chapter 5. The path discretization is based on angles with $l = 1.4\Delta x$. This value is twice greater than in the steady state case and thus induces a loss of accuracy. However, it allows for computational time gains, crucial in the transient case.

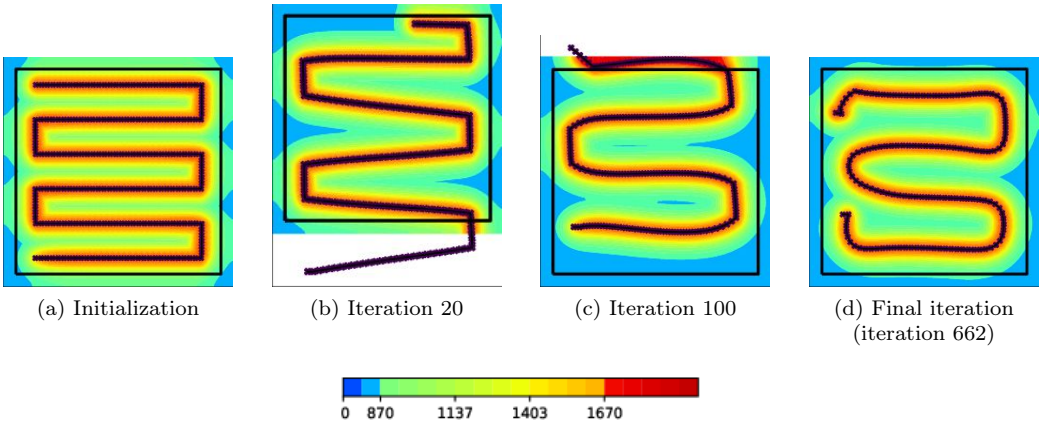
For each material, the physical properties (density, specific heat, conductivity and change of phase temperature) are given in the Nomenclature, as well as the source radius r, the source power P and the initial temperature y_{ini}. The absorption coefficient and the scaling parameters given by the calibration process are then applied. As in the steady state case, the maximum temperature inside the object is $1670K$ for the aluminum and $3400K$ for the titanium. Outside the object, the maximum temperature is fixed to $870K$ for the aluminum and $1800K$ for the titanium. The integer $\mathfrak{p}$ related to the phase constraint approximation is fixed to 64.
The tests run in section 7.2 remain simple tests without any further analysis such as in the steady state case (Chapter 6). If the angle regularization is tested, the path discretization remains fixed. The algorithm chosen is an Augmented Lagrangian method with $l_\phi^0 = 1$, $\mu_\phi = 10$, $l_{\text{M,s}}^0 = 1$, $\mu_{\text{M,s}} = 10$, $l_{\text{M,e}}^0 = 1$ and $\mu_{\text{M,e}} = 10$. Classic iterations consist in updating the angles. Every three iterations, the first point is also updated. Every five iterations, the final time is modified. The tolerance is initialized to 2 and multiplied by 0.9 every 50 iterations. The update coefficients are given by Algorithm 7.1. The code is stopped after the reference angle $\mathfrak{a}_{ref}$ defined by (7.2.26) gets smaller than 3e−8 ° and the number of iterations is bounded to 1000. The finite element computations are run with Freefem 3.56 [96] whereas the descent algorithm is run by Python 3.6.0. This problem is run on a MacBook laptop equipped with 2,3 GHz Intel Core i5 and a RAM of 16GB. No specific efforts for optimizing the Python code have been made but for a Cholesky decomposition of the rigidity matrix involved in the heat and adjoint equations.

For each of the aluminum and titanium cases, the building of a square is considered with four different algorithm initializations.

Remark 7.6. Since it is involved in the right hand side of the linear problems to solve, the Holder norm coefficient $\mathfrak{p}$ does not impact the computational time and can thus be chosen high. However, it impact the accuracy of the computations. No quantitative studies have been run but $\mathfrak{p} = 64$ seems to lead to accurate enough results.

Aluminium test case

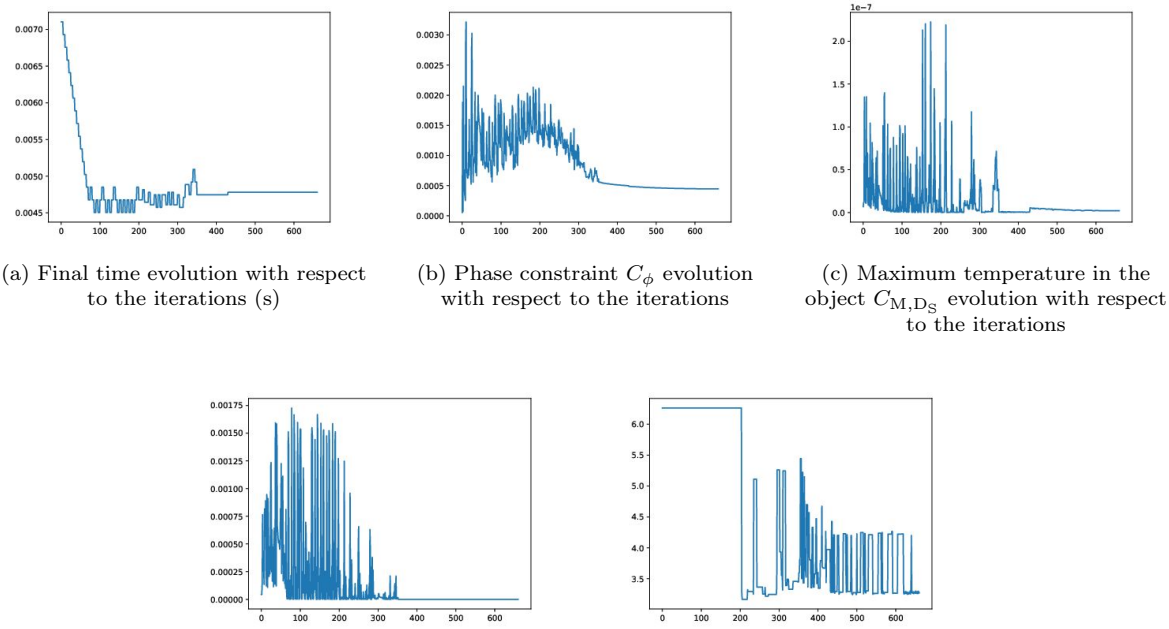
We first test this algorithm starting from a 6 lines zigzag. Figure 7.2 shows the evolution of the path and of the maximum temperature during the process with respect to iterations. The temperature colorbar used in the steady state context is kept here: the maximum temperature at each point within the square must belong to the segment $[870K, 1670K]$, avoiding the blue and red colors. Out of the domain, the maximum temperature should not overcome $870K$ and thus remain blue. Figure 7.3 gives the evolution of the final time, the phase constraint, the maximum temperature constraint within and without the square object with respect to iterations.



(a) Initialization (b) Iteration 20 (c) Iteration 100 (d) Final iteration (iteration 662)

(e) Temperature colorbar (K)

Figure 7.2: Path and maximum temperature ($\max_{t\in(0,t_{\mathrm{F}})} y(t)$) during the building with respect to the iterations, starting from a zigzag with 6 lines and using an angle-based path discretization (aluminum)



(a) Final time evolution with respect to the iterations (s)

(b) Phase constraint C_ϕ evolution with respect to the iterations

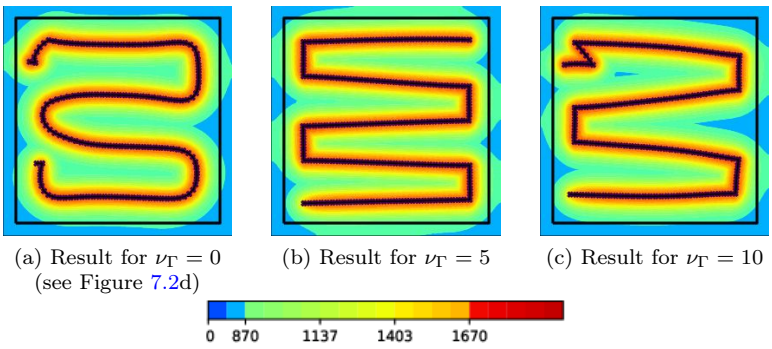
(c) Maximum temperature in the object $C_{\mathrm{M,D_S}}$ evolution with respect to the iterations

(d) Maximum temperature out of the object $C_{\mathrm{M,D\setminus D_S}}$ evolution with respect to the iterations

(e) Computational time per iteration (s)

Figure 7.3: Evolution of the final time, the phase constraint, the maximum temperature constraints with respect to the iterations, starting from a zigzag with 6 lines and using an angle-based path discretization (aluminum)

This first test for the aluminum initialized with a zigzag with 6 lines is very representative of the algorithm effects. The path is too long at initialization. In order to reduce the heat, the corner angles are amplified and the length reduced, leading the path to get out of the working domain. Because projecting is not possible in the angle based discretization, the final time reducing process proposed is applied until the path is included in the working domain. Thanks to this reduction, the path can finally adapt to the shape. This first results highlight once again the idea of a specific length adapted to the object to build. It also underlines the shrinking and lengthening process induced by the angle based discretization: the element must be added or removed at the end of the path before an adaptation of the angles to bring the source where required. Note that the final phase constraint is worst than the initial one. Further tests should favor this constraint to make sure to fulfill industrial requirements.
To illustrate the impact of the regularization, two additional tests have been run starting from the zigzag with 6 lines, respectively with $\nu_\Gamma = 5$ and $\nu_\Gamma = 10$. The resulting paths and temperatures are shown by Figure 7.4



(a) Result for $\nu_\Gamma = 0$ (see Figure 7.2d)

(b) Result for $\nu_\Gamma = 5$

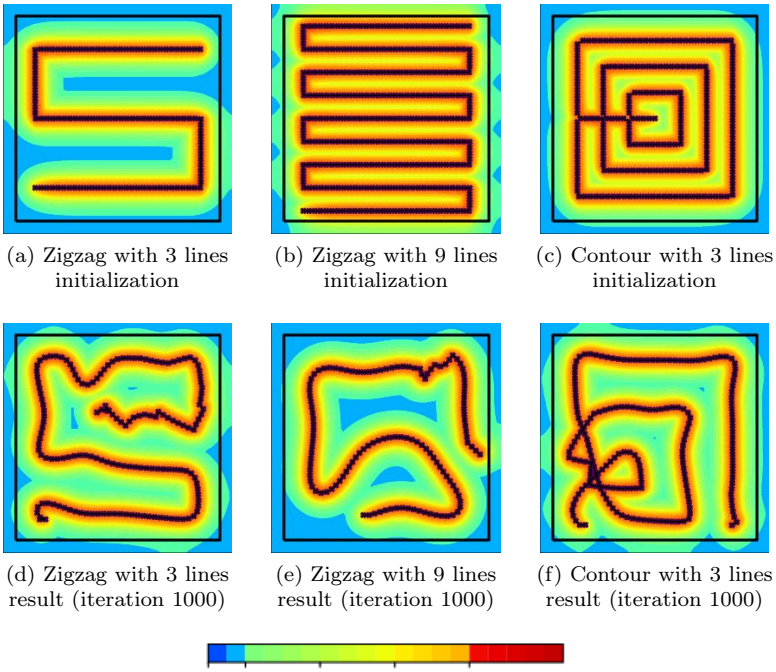
(c) Result for $\nu_\Gamma = 10$

Figure 7.4: Path and maximum temperature ($\max_{t\in(0,t_F)} y(t)$) during the building depending on the regularization ν_Γ, starting from a zigzag with 6 lines and using an angle-based path discretization (aluminum)

The regularization of the angle derivatives seems to have a poor impact on the final result. Indeed, the angles modification must, at least for the first iterations, impact mainly the zigzag corners. Regularizing decreases this derivative and mitigate the updates, preventing a correct minimization of the objective function. In the following, we choose $\nu_\Gamma = 0$.

To strengthen the analysis, three more initializations are tested: a 3 lines zigzag, a 9 lines zigzag and a contour with 3 lines. The initial and resulting paths and maximum temperatures are given by Figure 7.5. The resulting quantitative results are adimensionalized and summed up in Table 7.1:

$$\overline{C}_\phi = \frac{C_\phi}{|D|y_\phi^2}, \quad \overline{C}_{\mathrm{M,D_S}} = \frac{C_{\mathrm{M,D_S}}}{|D|y^2_{\mathrm{M,D_S}}}, \quad \overline{C}_{\mathrm{M,D\setminus D_S}} = \frac{C_{\mathrm{M,D\setminus D_S}}}{|D\setminus D_S|y^2_{\mathrm{M,D\setminus D_S}}}. \tag{7.2.27}$$



(a) Zigzag with 3 lines initialization (b) Zigzag with 9 lines initialization (c) Contour with 3 lines initialization

(d) Zigzag with 3 lines result (iteration 1000) (e) Zigzag with 9 lines result (iteration 1000) (f) Contour with 3 lines result (iteration 1000)

(g) Temperature colorbar (K)

Figure 7.5: Path and maximum temperature ($\max_{t\in(0,t_F)} y(t)$) during the building for the initial path (first line) and final path (second line), starting from three different initializations and using an angle-based path discretization (aluminum)

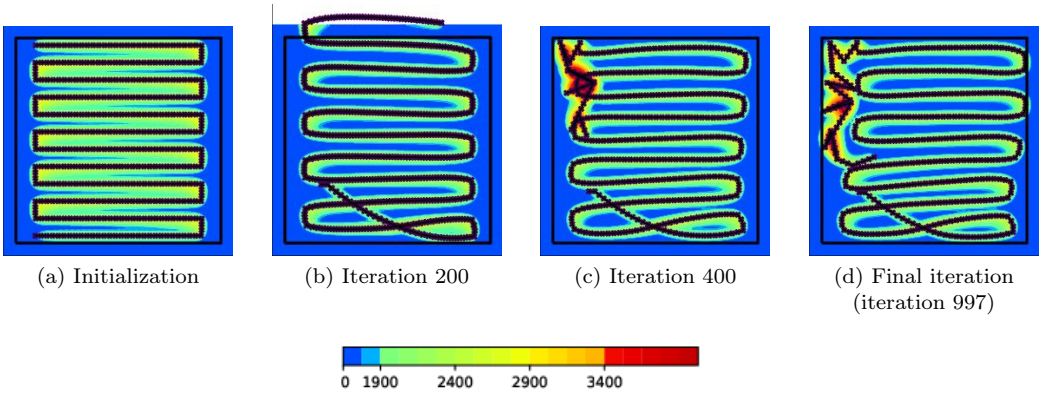
Initialization	t_F^{ini} (s)	t_F^{fin} (s)	$\overline{C}_\phi^{ini}$	$\overline{C}_\phi^{fin}$	$\overline{C}_{M,D_S}^{ini}$	$\overline{C}_{M,D_S}^{fin}$	$\overline{C}_{M,D\setminus D_S}^{ini}$	$\overline{C}_{M,D\setminus D_S}^{fin}$
zigzag 3 lines	3.881e−3	5.267e−3	1.32e−3	2.21e−4	5.91e−10	5.58e−9	2.26e−5	1.03e−5
zigzag 6 lines	7.103e−3	4.781e−3	4.46e−5	3.70e−4	1.55e−9	4.62e−10	1.55e−4	2.99e−6
zigzag 9 lines	1.019e−2	4.262e−3	1.15e−5	1.16e−3	3.49e−9	2.30e−9	6.00e−4	1.41e−5
contour 3 lines	7.865e−3	6.514e−3	1.39e−4	7.98e−5	1.71e−9	1.16e−9	8.57e−7	1.21e−5

Table 7.1: Comparison of the final results for the cost and the constraints of the final results (aluminum).

The zigzag with 3 lines is not long enough. The main modifications consist in adding path elements to satisfy the phase constraint. However, this addition leads to sharp angles decreasing the path regularity, which is a drawback for industrial applications. The two other cases seem to have encountered difficulties in the optimization. When starting from the zigzag with 9 lines, the algorithm seems to have decreased the length before increasing it again. The final path presents the irregularities coming from the final time management and does not satisfy the phase constraint. On the contrary, when starting from a contour with 3 lines, the algorithm has not managed to reduce enough the final time after the 1000 iterations allowed.

Titanium test case

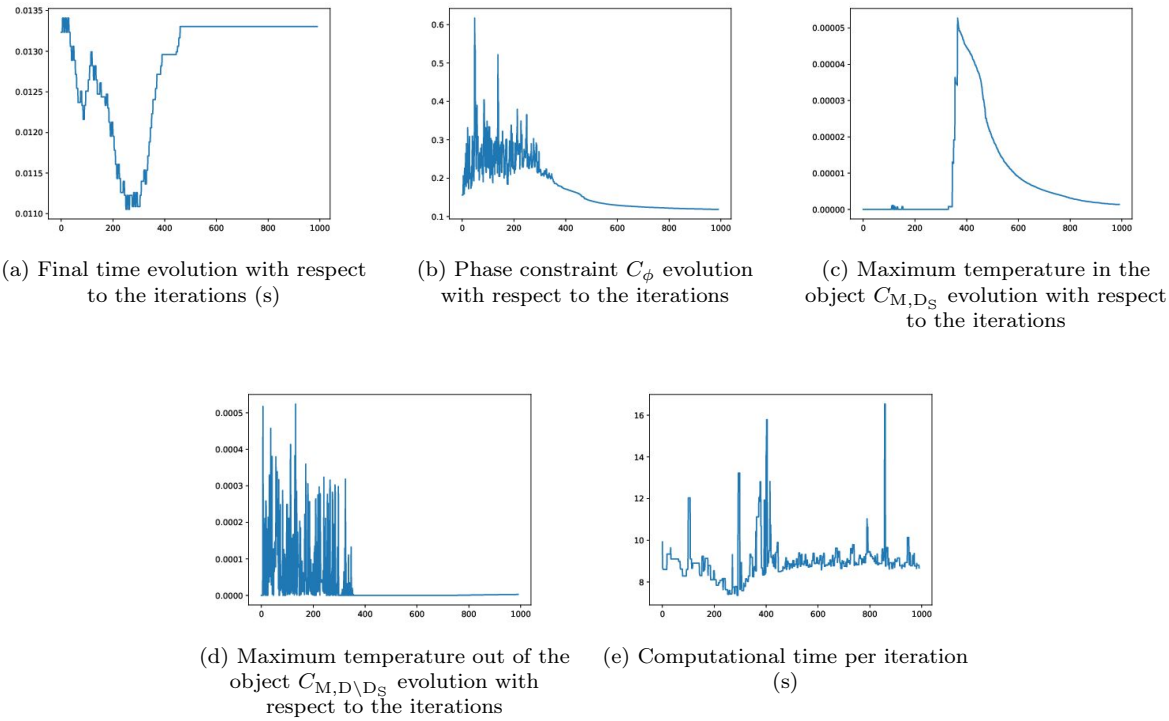
We first test this algorithm starting from a 12 lines zigzag. Figure 7.6 shows the evolution of the path and of the maximum temperature during the process with respect to iterations. In the titanium context, the maximum temperature at each point within the square must belong to the segment $[1900K, 3400K]$, avoiding the blue and red colors. Figure 7.7 gives the evolution of the final time, the phase constraint, the maximum temperature constraint within and without the square object with respect to iterations.



(a) Initialization (b) Iteration 200 (c) Iteration 400 (d) Final iteration (iteration 997)

(e) Temperature colorbar (K)

Figure 7.6: Path and maximum temperature ($\max_{t\in(0,t_F)} y(t)$) during the building with respect to the iterations, starting from a zigzag with 12 lines and using an angle-based path discretization (titanium)



(a) Final time evolution with respect to the iterations (s)

(b) Phase constraint C_ϕ evolution with respect to the iterations

(c) Maximum temperature in the object $C_{\mathrm{M,D_S}}$ evolution with respect to the iterations

(d) Maximum temperature out of the object $C_{\mathrm{M,D\setminus D_S}}$ evolution with respect to the iterations

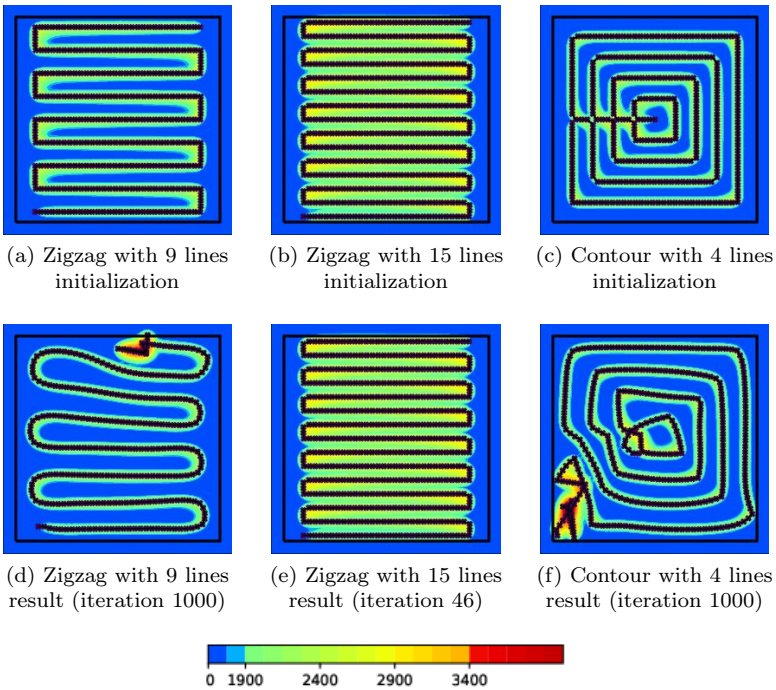
(e) Computational time per iteration (s)

Figure 7.7: Evolution of the final time, the phase constraint, the maximum temperature constraints with respect to the iterations, starting from a zigzag with 12 lines and using an angle-based path discretization (titanium).

In this new physical context, the conductivity is a lot lower and the final time is a lot higher than for the aluminum case. The initialization with a zigzag with 12 lines seems to cover most of the square but its left and right sides. On the other hand, the sharp corners related to the zigzag structure induce high temperatures. Thus, the first iterations aim at smoothing the path, getting it out of the working domain. This implies a length reduction and thus on the final iterations, the need for the increase of the final time. This optimization test highlights very well the main drawback of the algorithm: final time management.

To strengthen the analysis, three more iterations are tested: a 9 lines zigzag, a 15 lines zigzag and a contour with 4 lines. The initial and resulting paths and maximum temperatures are given by Figure

7.8. The resulting quantitative results are summed up in Table 7.2.



(a) Zigzag with 9 lines initialization (b) Zigzag with 15 lines initialization (c) Contour with 4 lines initialization

(d) Zigzag with 9 lines result (iteration 1000) (e) Zigzag with 15 lines result (iteration 46) (f) Contour with 4 lines result (iteration 1000)

(g) Temperature colorbar (K)

Figure 7.8: Path and maximum temperature ($\max_{t\in(0,t_F)} y(t)$) during the building for the initial path (first line) and final path (second line), starting from three different initializations and using an angle-based path discretization (titanium).

Initialization	t_F^{ini} (s)	t_F^{fin} (s)	$\overline{C}_\phi^{ini}$	$\overline{C}_\phi^{fin}$	$\overline{C}_{M,D_S}^{ini}$	$\overline{C}_{M,D_S}^{fin}$	$\overline{C}_{M,D\setminus D_S}^{ini}$	$\overline{C}_{M,D\setminus D_S}^{fin}$
zigzag 9 lines	1.019e−2	8.974e−3	4.12e−2	5.72e−2	0.00	2.02e−8	0.00	1.37e−6
zigzag 12 lines	1.324e−2	1.330e−2	2.73e−2	2.07e−2	0.00	6.99e−8	0.00	2.16e−6
zigzag 15 lines	1.608e−2	1.608e−2	2.57e−2	2.57e−2	0.00	0.00	4.69e−7	4.69e−7
contour 4 lines	1.019e−2	1.133e−2	5.01e−2	3.49e−2	0.00	4.76e−7	0.00	7.17e−8

Table 7.2: Comparison of the final results for the cost and the constraints of the final results (titanium).

But for the zigzag with 15 lines, each of the different initializations does not lead to an acceptable result. Indeed, the conductivity being much lower in the titanium case, the shrinking and lengthening processes are crucial in the path optimization. In addition to providing irregular paths, the angle based discretization used here does not manage to fulfill the constraints. It is obviously not adapted and must be modified.

7.2.6 Conclusion

The angle based discretization, if leading to an adaptation of the path to the shape, does not work correctly. Given the correct length, the path may be optimized. However, the optimization of the final time is not sufficient. The test of the square building starting from a zigzag with 12 lines in the titanium context illustrates very well the issue: intuitively, to improve the path, the horizontal lines of the zigzags should have been extended. In order to do so, the algorithm has first to add some elements at the end of the path and to then modify each angle to bring the source at the correct place. This makes the getting of admissible results either impossible or very long to converge. The lack of improved results in a few iterations prevent this method to be coupled with a shape optimization process and must be

modified. A first track consists in changing the final time update. In topology optimization processes (further developed in Chapter 8), the sensitivity to a topology change is computed so that the topology can be modified by adding or removing part of the object at the correct place. One could imagine the same kind of techniques to compute the sensitivity to the path to the addition of "beak" between two points instead of the classic segment (see Figure 7.9). This would allow the lengthening of the path to be done at the correct place while keeping the constant length segments properties that allows for computation time reductions. This technique has however not been further considered and remains part of the perspectives. A second idea, consisting in changing the path discretization from angle based to point based, is developed in Section 7.3.

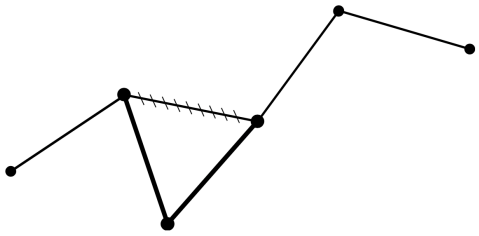


Figure 7.9: Lengthening the path at specific locations

7.3 PATH OPTIMIZATION BASED ON A POINTS PARAMETRIZATION

If the angle based path discretization allows for the optimization, it highly complicates the final time management. To deal with this issue, a second discretization is chosen, based on points. Very close to the steady state case, the discretization nodes are now the optimization variables and the line is rediscretized along the process. This increases the freedom in the design. Yet, it removes the constant element length property leading to heavier resolutions of the partial differential problem involved (see Chapter 5, Section 5.5).

Equation (7.3.1) gives the optimization problem in its discretized formulation. Because of the re-discretization process, note that the number of path nodes (and thus of optimization variables) is not fixed along the resolution. At each iteration, we consider the path as a sequence of N_u points $u = (u_1, ..., u_{N_u})$. The heat equation is discretized with an implicit Euler scheme following these points with the time step defined by $\forall i \in [\![1, N_u - 1]\!]$, $\Delta t_i = \|u_{i+1} - u_i\|/V$ (and V fixed to $1ms^{-1}$ in this work). A first time step Δt_0 is fixed. The heat equation then becomes $\forall i \in [\![1, N_u]\!]$,

$$\begin{cases} \rho c_p \dfrac{y_{i+1} - y_i}{\Delta t_i} - \nabla\left(\lambda \nabla y_{i+1}\right) + \beta\left(y_{i+1} - y_{\text{ini}}\right) = P \exp\left(-\dfrac{(x - u_{i+1})^2}{r^2}\right) = q_{i+1} & x \in D, \\ \lambda \partial_n y_{i+1} = 0 & x \in \partial D, \\ y_0 = y_{ini}. \end{cases} \tag{7.3.1}$$

The final time conforms to (the fixed first time step Δt_0 is not considered in the function)

$$t_{\text{F}} = \sum_{i=1}^{N_u - 1} \Delta t_i. \tag{7.3.2}$$

The constraints are discretized through an implicit scheme corresponding to the heat equation scheme:

$$C_\phi = \int_{D_S} \left[(y_\phi - N_{\mathfrak{p}})^+\right]^2 dx = \int_{D_S} \left[\left(y_\phi - \left(\frac{1}{t_{\text{F}}} \sum_{i=0}^{N_u - 1} \Delta t_i y_{i+1}^{\mathfrak{p}}\right)^{1/\mathfrak{p}}\right)^+\right]^2 dx. \tag{7.3.3}$$

and

$$\begin{cases} C_{\mathrm{M,D_S}} = \dfrac{1}{t_{\mathrm{F}}} \displaystyle\int_{D_S} \sum_{i=0}^{N_u-1} \Delta t_i \left[\left(y_{i+1} - y_{\mathrm{M,D_S}} \right)^+ \right]^2 dx, \\ C_{\mathrm{M,D\setminus D_S}} = \dfrac{1}{t_{\mathrm{F}}} \displaystyle\int_{D\setminus D_S} \sum_{i=0}^{N_u-1} \Delta t_i \left[\left(y_{i+1} - y_{\mathrm{M,D\setminus D_S}} \right)^+ \right]^2 dx. \end{cases} \tag{7.3.4}$$

The maximum constraints being split into a constraint in the object and a constraint out of the object, the optimization problem finally amounts to

$$\min_{\Gamma} t_{\mathrm{F}}, \quad \text{such that} \begin{cases} C_\phi = 0, \\ C_{\mathrm{M,D_S}} = 0, \\ C_{\mathrm{M,D\setminus D_S}} = 0, \\ y \in H^1(D) \text{ solution of (7.3.2).} \end{cases} \tag{7.3.5}$$

Remark 7.7. Remark 7.3, given in the angle based discretization, is still valid: centered schemes could be used for the constraints discretization.

7.3.1 Descent direction computation

Derivatives computation

We compute for each function the derivative with respect to points. To set the dependence of the steps Δt with respect to the points u, the definition of the discrete tangents is recalled:

$$\forall i \in [\![1, N_u - 1]\!],\ \tau_i = \frac{(u_{i+1} - u_i)}{\|u_{i+1} - u_i\|}. \tag{7.3.6}$$

Proposition 7.2. *Let $u = (u_i)_{i\in[\![1,N_u]\!]} \in D^{N_u}$ the nodal points of the path Γ. We have, $\forall v \in \mathbb{R}^{N_u\times 2}$,*

$$\begin{cases} D(\Delta t_i)(u)(v) = \tau_i \cdot (v_{i+1} - v_i) \quad \forall i \in [\![1, N_u - 1]\!], \\ D(\Delta t_0)(u)(v) = 0 \end{cases} \tag{7.3.7}$$

PROOF.
This comes from the definition of Δt: $\forall i \in [\![1, N_u - 1]\!]$, $\Delta t_i = \|u_{i+1} - u_i\|$ and Δt_0 is fixed. □

Proposition 7.3 gives the derivatives of each function with respect to the points.

Proposition 7.3. *Let $u = (u_i)_{i\in[\![1,N_u}\!] \in D^{N_u}$ the nodal points of the path Γ. Then, the final time and both constraints are differentiable at u and, $\forall\, v \in \mathbb{R}^{N_u\times 2}$,*

$$Dt_{\mathrm{F}}(u)(v) \quad = \tau_{N_u-1} v_{N_u} + \sum_{i=2}^{N_u-1} \left((\tau_{i-1} - \tau_i) \cdot v_i \right) - \tau_1 \cdot v_1, \tag{7.3.8}$$

with $\tau \cdot v = \tau_x v_x + \tau_y v_y$. For the phase constraint C_ϕ,

$$\begin{aligned} DC_\phi(u)(v) = &\left[- \left(\mathrm{HE}_{\phi,2} + \mathrm{C}_{\phi,2} \right) \tau_1 - \Delta t_0 \int_D \frac{2}{r^2} p_{\phi,1} q_1 \left(x - u_1 \right) dx \right] \cdot v_1 \\ &+ \sum_{i=2}^{N_u-1} \left[\left(\mathrm{HE}_{\phi,\mathrm{i}} + \mathrm{C}_{\phi,\mathrm{i}} \right) \tau_{i-1} - \left(\mathrm{HE}_{\phi,\mathrm{i+1}} + \mathrm{C}_{\phi,\mathrm{i+1}} \right) \tau_i - \Delta t_{i-1} \int_D \frac{2}{r^2} p_{\phi,i} q_i \left(x - u_i \right) dx \right] \cdot v_i \\ &+ \left[\left(\mathrm{HE}_{\phi,\mathrm{N_u}} + \mathrm{C}_{\phi,\mathrm{N_u}} \right) \tau_{N_u-1} - \Delta t_{N_u-1} \int_D \frac{2}{r^2} p_{\phi,N_u} q_{N_u} \left(x - u_{N_u} \right) dx \right] \cdot v_{N_u} \end{aligned} \tag{7.3.9}$$

with $\forall i \in [\![2, N_u]\!]$,

$$\begin{cases} \mathrm{HE}_{\phi,\mathrm{i}} = \displaystyle\int_{\mathrm{D}} \lambda \nabla \mathrm{y_i} \nabla \mathrm{p}_{\phi,\mathrm{i}} + \beta \left(\mathrm{y_i} - \mathrm{y_{ini}} \right) \mathrm{p}_{\phi,\mathrm{i}} - \mathrm{P} \exp\left(- \frac{\left(\mathrm{x} - \mathrm{u_i} \right)^2}{\mathrm{r}^2} \right) \mathrm{p}_{\phi,\mathrm{i}} \mathrm{dx}, \\ C_{\phi,i} = \displaystyle\int_D \frac{2}{\mathfrak{p} t_{\mathrm{F}}} \left(y_\phi - N_{\mathfrak{p}} \right)^+ \left[N_{\mathfrak{p}} - N_{\mathfrak{p}}^{1-\mathfrak{p}} y_i^{\mathfrak{p}} \right] \mathbb{1}_{D_S} dx, \end{cases} \tag{7.3.10}$$

and with

$$\begin{cases} \begin{cases} \rho c_p \dfrac{p_{\phi,N_u}}{\Delta t_{N_u-1}} - \nabla\left(\lambda \nabla p_{\phi,N_u}\right) + \beta p_{\phi,N_u} = \dfrac{2}{t_{\mathrm{F}}}\left(y_\phi - N_{\mathfrak{p}}\right)^+ N_{\mathfrak{p}}^{1-\mathfrak{p}} y_{N_u}^{\mathfrak{p}-1} \mathbb{1}_{D_S} & in\, D \\ \lambda \partial_n p_{\phi,N_u} = 0 & on\, \partial D, \end{cases} \\ \begin{cases} \rho c_p \dfrac{p_{\phi,i} - p_{\phi,i+1}}{\Delta t_{i-1}} - \nabla\left(\lambda \nabla p_{\phi,i}\right) + \beta p_{\phi,i} = \dfrac{2}{t_{\mathrm{F}}}\left(y_\phi - N_{\mathfrak{p}}\right)^+ N_{\mathfrak{p}}^{1-\mathfrak{p}} y_i^{\mathfrak{p}-1} \mathbb{1}_{D_S} & in\, D, \\ \lambda \partial_n p_{\phi,i} = 0 & on\, \partial D. \end{cases} \end{cases} \tag{7.3.11}$$

Both maximum temperature constraints gets the same type of derivatives and we consider here $C_{\mathrm{M,X}}$ *with* $X = D_S$ *or* $X = D \setminus D_S$*. depending on within the object derivative is given by*

$$\begin{aligned} DC_{\mathrm{M,X}}(u)(v) = & \left[-\left(\mathrm{HE}_{\mathrm{M_X},2} + \mathrm{C}_{\mathrm{M_X},2}\right)\tau_1 - \Delta t_0 \int_D \frac{2}{r^2} p_{\mathrm{M_X},1} q_1(x-u_1)dx \right] \cdot v_1 \\ & + \sum_{i=2}^{N_u-1} \left[\left(\mathrm{HE}_{\mathrm{M_X,i}} + \mathrm{C}_{\mathrm{M_X,i}}\right)\tau_{i-1} - \left(\mathrm{HE}_{\mathrm{M_X,i+1}} + \mathrm{C}_{\mathrm{M_X,i+1}}\right)\tau_i - \Delta t_{i-1} \int_D \frac{2}{r^2} p_{\mathrm{M_X,i}} q_i(x-u_i)dx \right] \cdot v_i \\ & + \left[\left(\mathrm{HE}_{\mathrm{M_X,N_u}} + \mathrm{C}_{\mathrm{M_X,N_u}}\right)\tau_{N_u-1} - \Delta t_{N_u-1} \int_D \frac{2}{r^2} p_{\mathrm{M_X,N_u}} q_{N_u}(x-u_{N_u})dx \right] \cdot v_{N_u} \end{aligned} \tag{7.3.12}$$

with $\forall i \in [\![2, N_u]\!]$*,*

$$\begin{cases} \mathrm{HE}_{\mathrm{M_X,i}} = \displaystyle\int_{\mathrm{D}} \lambda \nabla \mathrm{y_i} \nabla \mathrm{p_{M_X,i}} + \beta\left(\mathrm{y_i} - \mathrm{y_{ini}}\right)\mathrm{p_{M_X,i}} - \mathrm{P}\exp\left(-\frac{\left(\mathrm{x}-\mathrm{u_i}\right)^2}{\mathrm{r}^2}\right)\mathrm{p_{M_X,i}}\mathrm{dx}, \\ C_{\mathrm{M_X,i}} = \dfrac{1}{t_{\mathrm{F}}}\left[\displaystyle\int_D \left[\left(y_i - y_{\mathrm{M}}\right)^+\right]^2 \mathbb{1}_X dx - C_{\mathrm{M,X}}\right] \end{cases} \tag{7.3.13}$$

and with

$$\begin{cases} \begin{cases} \rho c_p \dfrac{p_{\mathrm{M_X,N_u}}}{\Delta t_{N_u-1}} - \nabla\left(\lambda \nabla p_{\mathrm{M_X,N_u}}\right) + \beta p_{\mathrm{M_X,N_u}} = -\dfrac{2}{t_{\mathrm{F}}}\left(y_{N_u} - y_{\mathrm{M}}\right)^+ \mathbb{1}_X & in\, D \\ \lambda \partial_n p_{\mathrm{M_X,N_u}} = 0 & on\, \partial D, \end{cases} \\ \begin{cases} \rho c_p \dfrac{p_{\mathrm{M_X,i}} - p_{\mathrm{M_X,i+1}}}{\Delta t_{i-1}} - \nabla\left(\lambda \nabla p_{\mathrm{M_X,i}}\right) + \beta p_{\mathrm{M_X,i}} - \dfrac{2}{t_{\mathrm{F}}}\left(y_i - y_{\mathrm{M}}\right)^+ \mathbb{1}_X & in\, D \\ \lambda \partial_n p_{\mathrm{M_X,i}} = 0 & on\, \partial D. \end{cases} \end{cases} \tag{7.3.14}$$

PROOF.
The differentiation of t_{F} is naturally derived from Proposition 7.2. Following the method of Céa, three Lagrangian functions, $\mathcal{L}_\phi$ related to the phase constraint, $\mathcal{L}_{\mathrm{M_{D_S}}}$ to the maximum temperature inside the object and $\mathcal{L}_{\mathrm{M_{D\setminus D_S}}}$ to the maximum temperature outside the object, are introduced as

$$\begin{cases} \mathcal{L}_\phi\left(\Gamma, y, p\right) = C_\phi(\Gamma, y) + \mathrm{HE}(\Gamma, \mathrm{y}, \mathrm{p}), \\ \mathcal{L}_{\mathrm{M_{D_S}}}\left(\Gamma, y, p\right) = C_{\mathrm{M,D_S}}(\Gamma, y) + \mathrm{HE}(\Gamma, \mathrm{y}, \mathrm{p}), \\ \mathcal{L}_{\mathrm{M_{D\setminus D_S}}}\left(\Gamma, y, p\right) = C_{\mathrm{M,D\setminus D_S}}(\Gamma, y) + \mathrm{HE}(\Gamma, \mathrm{y}, \mathrm{p}), \end{cases} \tag{7.3.15}$$

with

$$\begin{aligned} \mathrm{HE} = \sum_{\mathrm{i}=0}^{\mathrm{N_u}-1} \int_{\mathrm{D}} \Big[& \rho \mathrm{c_p}\left(\mathrm{y_{i+1}} - \mathrm{y_i}\right)\mathrm{p_{i+1}} + \lambda \Delta \mathrm{t_i} \nabla \mathrm{y_{i+1}} \nabla \mathrm{p_{i+1}} + \beta \Delta \mathrm{t_i}\left(\mathrm{y_{i+1}} - \mathrm{y_{ini}}\right)\mathrm{p_{i+1}} \\ & - P \Delta t_i \exp\left(-\frac{\left(x - u_{i+1}\right)^2}{r^2}\right) p_{i+1} \Big] dx. \end{aligned} \tag{7.3.16}$$

This quantity can be differentiated with respect to the temperature leading to, $\forall i \in [\![1, N_u]\!]$, $\forall \Phi \in H^1(D)$,

$$\begin{cases} \partial_{y_{N_u}}(\mathrm{HE})(\Gamma, \mathrm{y}, \mathrm{p})(\Phi) = \displaystyle\int_{\mathrm{D}} \rho \mathrm{c_p} \mathrm{p_{N_u}} \Phi + \lambda \Delta \mathrm{t_{N_u-1}} \nabla \mathrm{p_{N_u}} \nabla \Phi + \beta \Delta \mathrm{t_{N_u-1}} \mathrm{p_{N_u}} \Phi \mathrm{dx}, \\ \partial_{y_i}(\mathrm{HE})(\Gamma, \mathrm{y}, \mathrm{p})(\Phi) = \displaystyle\int_{\mathrm{D}} \rho \mathrm{c_p}\left(\mathrm{p_i} - \mathrm{p_{i+1}}\right)\Phi + \lambda \Delta \mathrm{t_{i-1}} \nabla \mathrm{p_i} \nabla \Phi + \beta \Delta \mathrm{t_{i-1}} \mathrm{p_i} \Phi \mathrm{dx}, \quad \forall i \in [\![1, N_u - 1]\!]. \end{cases} \tag{7.3.17}$$

as well as the phase constraint:

$$\partial_{y_i} C_\phi(y)(\Phi) = \int_D -2\left(y_\phi - N_\mathfrak{p}\right)^+ N_\mathfrak{p}^{1-\mathfrak{p}} \frac{\Delta t_{i-1}}{t_\mathrm{F}} y_i^{\mathfrak{p}-1} \mathbb{1}_{D_S} dx, \tag{7.3.18}$$

and the maximum temperature constraint:

$$\begin{aligned} \partial_{y_i} C_{\mathrm{M,D_S}}(y)(\Phi) &= \int_D \frac{2\Delta t_{i-1}}{t_\mathrm{F}} \left(y_i - y_{\mathrm{M,D_S}}\right)^+ \mathbb{1}_{D_S} dx, \\ \partial_{y_i} C_{\mathrm{M,D\setminus D_S}}(y)(\Phi) &= \int_D \frac{2\Delta t_{i-1}}{t_\mathrm{F}} \left(y_i - y_{\mathrm{M,D\setminus D_S}}\right)^+ \mathbb{1}_{D\setminus D_S} dx. \end{aligned} \tag{7.3.19}$$

The phase adjoint p_ϕ and the maximum constraint adjoints $p_{\mathrm{M_{D_S}}}$ and $p_{\mathrm{M_{D\setminus D_S}}}$ respectively satisfy, $\forall i \in [\![1, N_u]\!]$, $\forall \phi \in H^1(D)$,

$$\begin{cases} \partial_{y_i} \mathrm{HE}(\Gamma, \mathrm{y}, \mathrm{p}_\phi)(\Phi) + \partial_{\mathrm{y_i}} \mathrm{C}_\phi(\mathrm{y})(\Phi) = 0, \\ \partial_{y_i} \mathrm{HE}(\Gamma, \mathrm{y}, \mathrm{p}_{\mathrm{M_{D_S}}})(\Phi) + \partial_{\mathrm{y_i}} \mathrm{C}_{\mathrm{M,D_S}}(\mathrm{y})(\Phi) = 0, \\ \partial_{y_i} \mathrm{HE}(\Gamma, \mathrm{y}, \mathrm{p}_{\mathrm{M_{D\setminus D_S}}})(\Phi) + \partial_{\mathrm{y_i}} \mathrm{C}_{\mathrm{M,D\setminus D_S}}(\mathrm{y})(\Phi) = 0. \end{cases} \tag{7.3.20}$$

This leads to the adjoint functions.

The derivatives with respect to the points come from the differentiation of $\mathcal{L}_\phi$ and $\mathcal{L}_{\max}$ with respect to the points. On the one hand, the heat equation function HE gives, $\forall v \in \mathbb{R}^{N_u \times 2}$:

$$\begin{aligned} \partial_u \mathrm{HE(u)(v)} &= \sum_{i=0}^{N_u-1} \left(\partial_{\Delta t_i} \mathrm{HE}\right) D(\Delta t_i)(u)(v) - \sum_{i=0}^{N_u-1} v_{i+1} \int_D \Delta t_i \frac{2}{r^2} q_{i+1} \left(x - u_{i+1}\right) p_{i+1} dx \\ &= \sum_{\mathbf{i=1}}^{N_u-1} \left(\partial_{\Delta t_i} \mathrm{HE}\right) D(\Delta t_i)(u)(v) - \sum_{i=0}^{N_u-1} v_{i+1} \int_D \Delta t_i \frac{2}{r^2} q_{i+1} \left(x - u_{i+1}\right) p_{i+1} dx. \end{aligned} \tag{7.3.21}$$

We set $(\mathrm{HE})_{i \in [\![2, N_u]\!]}$ such that $\forall i \in [\![2, N_u]\!]$,

$$\mathrm{HE_i} = \partial_{\Delta \mathrm{t_{i-1}}} \mathrm{HE} = \int_\mathrm{D} \left(\lambda \nabla \mathrm{y_i} \nabla \mathrm{p_i} + \beta \left(\mathrm{y_i} - \mathrm{y_{ini}}\right) \mathrm{p_i} - \mathrm{q_i p_i}\right) \mathrm{dx}. \tag{7.3.22}$$

This finally leads to

$$\begin{aligned} \partial_u \mathrm{HE(u)(v)} &= \mathrm{HE_{N_u} \tau_{N_u-1} \cdot v_{N_u}} + \sum_{\mathrm{i=2}}^{\mathrm{N_u-1}} \left(\mathrm{HE_i \tau_{i-1} - HE_{i+1} \tau_i}\right) \cdot \mathrm{v_i} - \mathrm{HE_2 \tau_1 v_1} \\ &\quad - \sum_{i=1}^{N_u} v_i \Delta t_{i-1} \int_D \frac{2}{r^2} q_i \left(x - u_i\right) p_i dx \end{aligned} \tag{7.3.23}$$

Of course, the adjoint p must be chosen related to the phase (p_ϕ) or to the maximum temperature constraint (p_M)

The partial derivative of the phase constraint is

$$\begin{aligned} \partial_u C_\phi(u)(v) &= \int_D 2\left(y_\phi - N_\mathfrak{p}\right)^+ \frac{N_\mathfrak{p}^{1-\mathfrak{p}}}{\mathfrak{p} t_\mathrm{F}} \left[N_\mathfrak{p}^\mathfrak{p} D(t_\mathrm{F})(u)(v) - \sum_{i=0}^{N_u-1} D(\Delta t_i)(u)(v) y_{i+1}^\mathfrak{p}\right] \mathbb{1}_{D_S} dx \\ &= \int_D 2\left(y_\phi - N_\mathfrak{p}\right)^+ \frac{N_\mathfrak{p}^{1-\mathfrak{p}}}{\mathfrak{p} t_\mathrm{F}} \sum_{i=1}^{N_u-1} \left[N_\mathfrak{p}^\mathfrak{p} - y_{i+1}^\mathfrak{p}\right] \mathbb{1}_{D_S} D(\Delta t_i)(u)(v) dx \\ &= \sum_{i=1}^{N_u-1} \underbrace{\int_D 2\left(y_\phi - N_\mathfrak{p}\right)^+ \frac{N_\mathfrak{p}^{1-\mathfrak{p}}}{\mathfrak{p} t_\mathrm{F}} \left[N_\mathfrak{p}^\mathfrak{p} - y_{i+1}^\mathfrak{p}\right] \mathbb{1}_{D_S} dx}_{C_{\phi,i+1}} D(\Delta t_i)(u)(v) \\ &= C_{\phi,N_u} \tau_{N_u-1} \cdot v_{N_u} + \sum_{i=2}^{N_u-1} \left(C_{\phi,i} \tau_{i-1} - C_{\phi,i+1} \tau_i\right) \cdot v_i - C_{\phi,2} \tau_1 \cdot v_1 \end{aligned} \tag{7.3.24}$$

This finally leads to the result. The partial derivative of the maximum constraint in the object is

$$\begin{aligned}\partial C_{\mathrm{M,D_S}}(u)(v) &= \sum_{i=1}^{N_u-1} \underbrace{\frac{1}{t_\mathrm{F}}\left(\int_D \left[(y_{i+1}-y_\mathrm{M})^+\right]^2 \mathbb{1}_{D_S} dx - C_{\mathrm{M,D_S}}\right)}_{C_{\mathrm{M_{D_S}},\mathrm{i+1}}} D(\Delta t_i)(u)(v) \\ &= C_{\mathrm{M_{D_S}},\mathrm{N_u}}\tau_{N_u-1}\cdot v_{N_u} + \sum_{i=2}^{N_u-1}\left(C_{\mathrm{M_{D_S}},\mathrm{i}}\tau_{i-1} - C_{\mathrm{M_{D_S}},\mathrm{i+1}}\tau_i\right)\cdot v_i - C_{\mathrm{M_{D_S}},2}\tau_1\cdot v_1,\end{aligned} \tag{7.3.25}$$

and outside the object,

$$\begin{aligned}\partial C_{\mathrm{M,D\setminus D_S}}(u)(v) &= \sum_{i=1}^{N_u-1} \underbrace{\frac{1}{t_\mathrm{F}}\left(\int_D \left[(y_{i+1}-y_\mathrm{M})^+\right]^2 \mathbb{1}_{D\setminus D_S} dx - C_{\mathrm{M,D\setminus D_S}}\right)}_{C_{\mathrm{M_{D\setminus D_S}},\mathrm{i+1}}} D(\Delta t_i)(u)(v) \\ &= C_{\mathrm{M_{D\setminus D_S}},\mathrm{N_u}}\tau_{N_u-1}\cdot v_{N_u} + \sum_{i=2}^{N_u-1}\left(C_{\mathrm{M_{D\setminus D_S}},\mathrm{i}}\tau_{i-1} - C_{\mathrm{M_{D\setminus D_S}},\mathrm{i+1}}\tau_i\right)\cdot v_i - C_{\mathrm{M_{D\setminus D_S}},2}\tau_1\cdot v_1,\end{aligned} \tag{7.3.26}$$

□

Regularization

Alike the steady state case, a scalar product must be chosen to compute the gradients from the derivatives. A $H^1(\Gamma)$ scalar product is once again elected (see (7.2.24) and Remark 7.5). Each derivative can be written under the form

$$Df(u)(v) = \sum_{i=1}^{N_u} G_i \cdot v_i, \tag{7.3.27}$$

and the gradient computation amounts to solving the following minimization problem:

$$\nabla_u f = \underset{Q\in\mathbb{R}^{N_u\times 2}}{\arg\min} \sum_{i=1}^{N_u-1}\frac{\Delta t_i}{2}\left(\nu_\Gamma^2\left(\frac{Q_{i+1}-Q_i}{\Delta t_i}\right)^2 + \frac{Q_i^2+Q_{i+1}^2}{2}\right) - \sum_{i=1}^{N_u-1}\Delta t_i\frac{G_iQ_i+G_{i+1}Q_{i+1}}{2}. \tag{7.3.28}$$

7.3.2 Numerical Augmented Lagrangian algorithm

Like in the steady state case, the Augmented Lagrangian method can be used. Introducing the multipliers l_ϕ, $l_{\mathrm{M,s}}$ and $l_{\mathrm{M,e}}$ as well as the multipliers μ_ϕ, $\mu_{\mathrm{M,s}}$ and $\mu_{\mathrm{M,e}}$, the new objective function is

$$\begin{aligned}\mathcal{L}_{\mathrm{ALM}}(u, l_\phi, l_{\mathrm{M,s}}, l_{\mathrm{M,e}}) = t_\mathrm{F} + l_\phi C_\phi + \frac{\mu_\phi}{2}C_\phi^2 \quad &+ l_{\mathrm{M,s}}C_{\mathrm{M,D_S}} + \frac{\mu_{\mathrm{M,s}}}{2}C_{\mathrm{M,D_S}}^2 \\ &+ l_{\mathrm{M,e}}C_{\mathrm{M,D\setminus D_S}} + \frac{\mu_{\mathrm{M,e}}}{2}C_{\mathrm{M,D\setminus D_S}}^2.\end{aligned} \tag{7.3.29}$$

The update of the multipliers are

$$\begin{cases} l_\phi^{k+1} = l_\phi^k + c^k C_\phi^k \\ l_{\mathrm{M,s}}^{k+1} = l_{\mathrm{M,s}}^k + \mu_{\mathrm{M,s}}^k C_{\mathrm{M,D_S}}^k \\ l_{\mathrm{M,e}}^{k+1} = l_{\mathrm{M,e}}^k + \mu_{\mathrm{M,e}}^k C_{\mathrm{M,D\setminus D_S}}^k \end{cases} \tag{7.3.30}$$

As for the derivative with respect to the points, it is given by

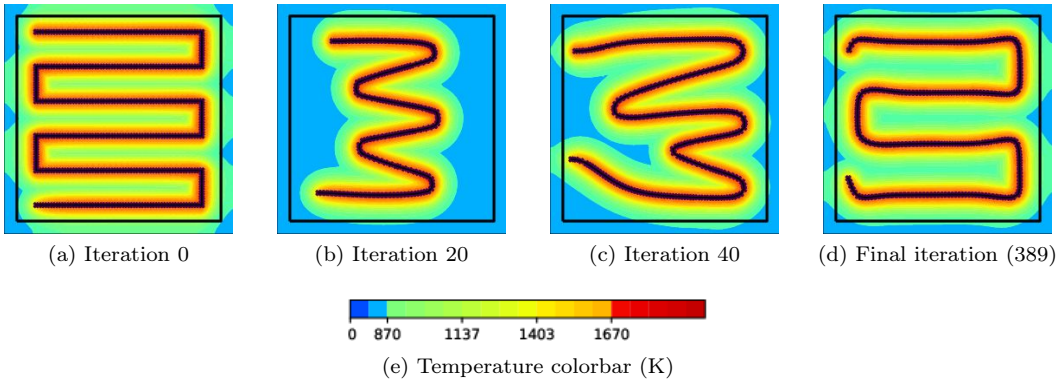
$$\begin{aligned}D_u\mathcal{L}_{\mathrm{ALM}}(u)(v) = D_u t_\mathrm{F}(u)(v) \quad &+ \left(l_\phi + \mu_\phi C_\phi\right) D_u C_\phi(u)(v) \\ &+ \left(l_{\mathrm{M,s}} + \mu_{\mathrm{M,s}}C_{\mathrm{M,D_S}}\right) D_u C_{\mathrm{M,D_S}}(u)(v) \\ &+ \left(l_{\mathrm{M,e}} + \mu_{\mathrm{M,e}}C_{\mathrm{M,D\setminus D_S}}\right) D_u C_{\mathrm{M,D\setminus D_S}}(u)(v).\end{aligned} \tag{7.3.31}$$

Remark 7.8. Note that, as Section 7.2 and unlike the steady state context, the functions are not normalized in the definition of $\mathcal{L}_{\mathrm{ALM}}$ (see Remark 7.2).

7.3.3 First results

As before, the domain considered is $D = [-0.7, 0.7] \times [-0.7, 0.7]$ (given in mm). The accuracy of the discretized path is chosen such that $d_{\mathrm{l}} = 0.7\Delta x \leq \Delta u \leq 1.4\Delta x = d_{\mathrm{u}}$. This distance is chosen bigger than in the steady state context to decrease the computational time. Indeed, because in this discretization choice, a linear problem must be inverted at each time step, at each iteration, the computational time is much higher than in the steady state case and in the angle-based discretization. Finally, the regularization coefficient is chosen as $\nu_\Gamma = 20 d_{\mathrm{l}}$. The physical coefficients are the same than in Section 7.2. As for the coefficient Δt_0, it is fixed to d_{u}. The finite element analysis is once again run with Freefem 3.56 and Python 3.6.0. The resolution of the different linear problems is done using conjugate gradient methods (library scipy.sparse.linalg.cg in Python, with a tolerance set to 10^{-10} for the convergence of the iteration method used for the linear system resolution).

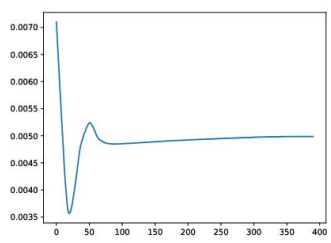
The first results correspond to the building of the square object (see Figure 6.1). In order to understand how the algorithm modifies the path, the optimization has been run with the Augmented Lagrangian method with $l^0_\phi = l^0_{\mathrm{M,s}} = l^0_{\mathrm{M,e}} = 0$ and $\mu_\phi = 10$, $\mu_{\mathrm{M,s}} = 10$ and $\mu_{\mathrm{M,e}} = 10$. The regularization coefficient is fixed to 20. First in the aluminum context, the optimization is run starting from a zigzag with 6 lines (distance between two horizontal lines 2.10e$-4m$). The maximum temperature over the scanning as well as the path are displayed for different iterations in Figure 7.10. This maximum temperature must remain within $[870K, 1670K]$ in the object, corresponding to green to orange colors. The evolution of the final time t_{F}, of the phase, of the maximum temperature constraints and the mean time per iteration are displayed in Figure 7.11.



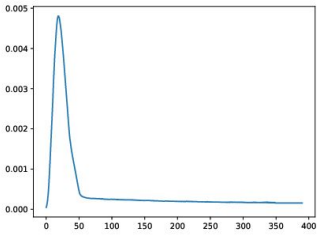
(a) Iteration 0 (b) Iteration 20 (c) Iteration 40 (d) Final iteration (389)

(e) Temperature colorbar (K)

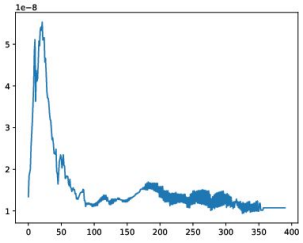
Figure 7.10: Path and maximum temperature $\left(\max_{t\in(0,t_{\mathrm{F}})} y(t)\right)$ during the building with respect to the iterations, starting from a zigzag with 6 lines and using an point-based path discretization (aluminum)



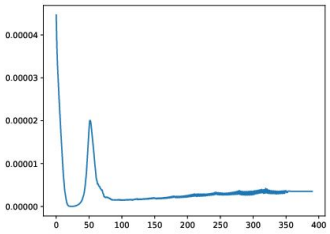
(a) Length t_{F} (s)



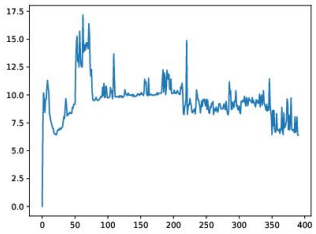
(b) Phase constraint C_{ϕ}



(c) Maximum temperature constraint inside $C_{\mathrm{M,D_S}}$



(d) Maximum temperature constraint outside $C_{\mathrm{M,D\setminus D_S}}$

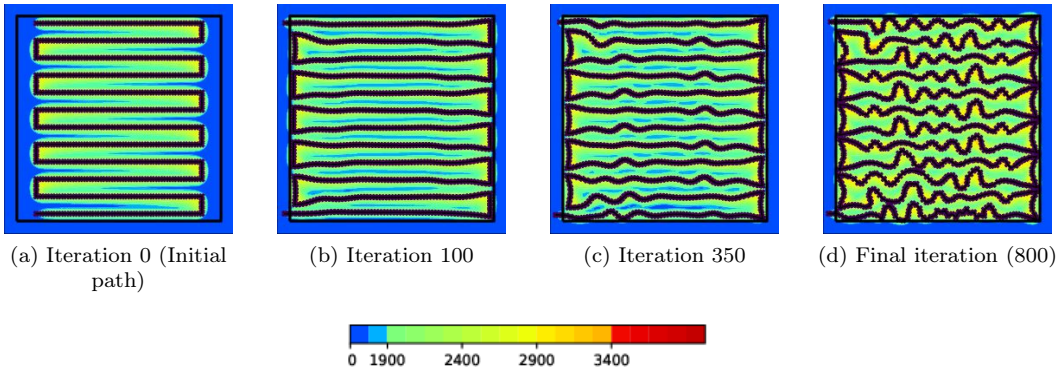


(e) Computational time per iteration (s)

Figure 7.11: Evolution of the final time, the phase constraint, the maximum temperature constraints with respect to the iterations, starting from a zigzag with 6 lines and using an point-based path discretization (aluminum)

First of all, the final path is very similar to the path found using an angle-based discretization (Figure 7.2). Since both algorithms converge to the same result, we can reasonably assume that the techniques used are relevant to address the optimization problem. However, the optimization iterations to get there are very different. Indeed, in this point-based case, the final time is first drastically reduced and then increased again, setting the new source points in accordance with the problem requirements. Unlike in the angle-based algorithm, the final time is modified in a smooth way which infers that the optimization process is smoother. The resolution time is however higher than for the angle based discretization, going from a mean time per iteration of 4s to a mean time per iteration of 8s. Indeed, the distance between points being fixed for the angle-based discretization, only one linear system must be inverted. Solving the heat equation then simply amounts to a matrix product. In the point based discretization, this is not possible anymore which increases the computational time. The conjugate gradient method used in the point-based problem to solve linear problems could be accelerated by adding a preconditioner. This remains part of the perspectives.

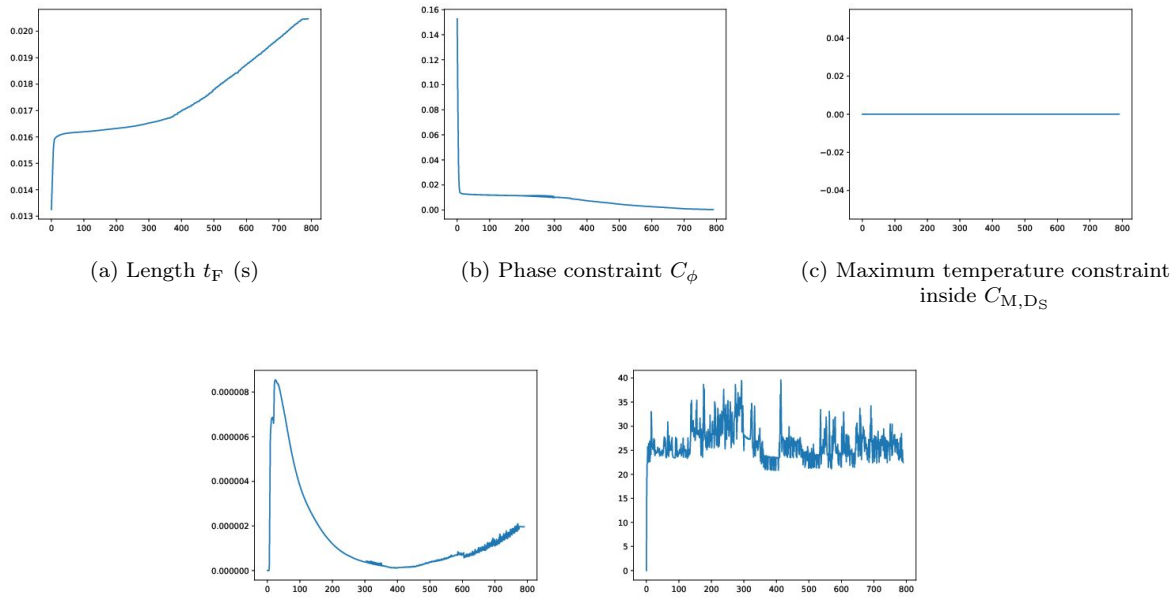
This first result must be supplemented by a test with the titanium. Indeed, in the angle based discretization, the optimization was complicated by the decrease of the conductivity. Thus, the titanium case is considered with an optimization process starting from a zigzag with 12 lines. Figure 7.12 shows the path and the resulting maximal temperature that must remain in $[1900K, 3400K]$ (green to orange again). The different quantities evolution are given by Figure 7.13.



(a) Iteration 0 (Initial path) (b) Iteration 100 (c) Iteration 350 (d) Final iteration (800)

(e) Temperature colorbar (K)

Figure 7.12: Path and maximum temperature $\left(\max_{t\in(0,t_F)} y(t)\right)$ during the building with respect to the iterations, starting from a zigzag with 12 lines and using an point-based path discretization (titanium).



(a) Length t_F (s) (b) Phase constraint C_ϕ (c) Maximum temperature constraint inside C_{M,D_S}

(d) Maximum temperature constraint outside $C_{M,D\setminus D_S}$ (e) Computational time per iteration (s)

Figure 7.13: Evolution of the final time, the phase constraint, the maximum temperature constraints with respect to the iterations, starting from a zigzag with 12 lines and using an point-based path discretization (titanium).

Unlike the angle based discretization, the algorithm is able to lengthen the path (and thus the final time since the source velocity is constant) on the left and right sides (Figure 7.12(b)). Then, to remove the unmelted parts between the horizontal lines, perturbations are introduced to fill the holes (Figure 7.12(c)). These perturbations are deeply related with the regularization process. Even though it is much more expensive in terms of computational cost (mean time per iteration of 27s versus 10s for the angle based discretization), this point based discretization is clearly more efficient.

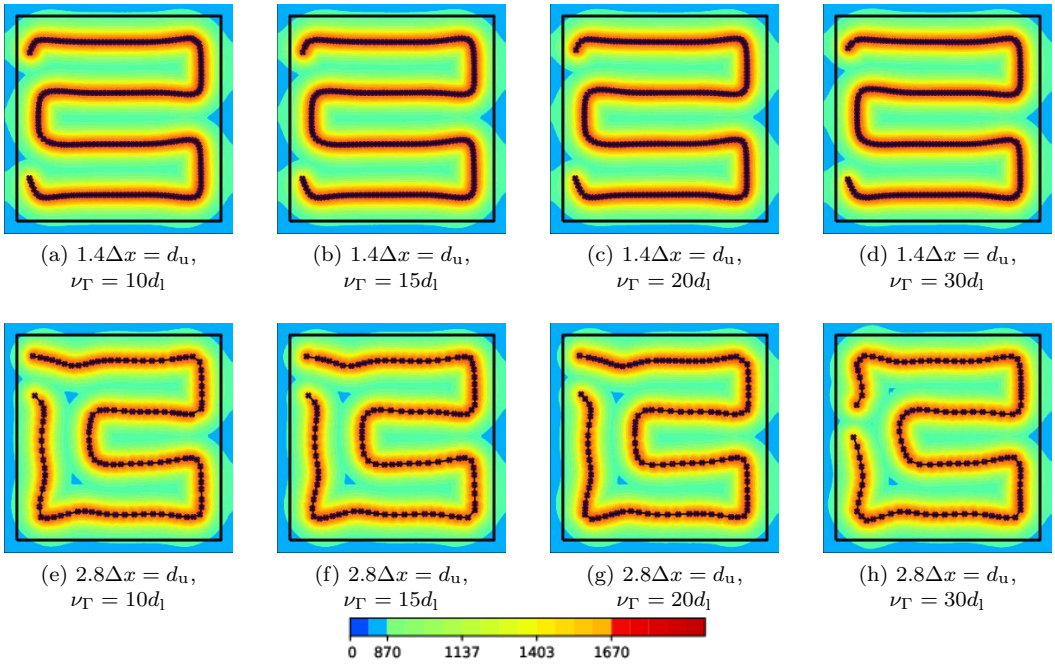
7.3.4 Impact of the numerical choices

As in the steady state context, numerical and algorithm choices impact the result. This paragraph analyzes three of them: the path discretization accuracy, the regularization process and the algorithm

choices.

Impact of the path discretization and of the regularization

As mentioned in the first results analysis, the regularization process seems to be crucial. Indeed, especially for low conductivity, small path variations are required. Yet, industrial constraints prevent them to be too sharp. The accuracy of the discretization (and thus the size of path elements) as well as the regularization impact these variations. Four different regularizations are considered: $\nu_\Gamma/d_\mathrm{l} \in \{10, 15, 20, 30\}$ and two discretization coefficients $d_\mathrm{l} = 0.7\Delta x \leq \Delta u \leq 1.4\Delta x = d_\mathrm{u}$ and $d_\mathrm{l} = 1.4\Delta x \leq \Delta u \leq 2.8\Delta x = d_\mathrm{u}$. In the aluminum case, the final iteration path and maximal temperature are given by Figure 7.14 and the adimensionalized final values defined by (7.2.27) are summed up in Table 7.3. As for the titanium, the results are given by Figure 7.15 and Table 7.4.



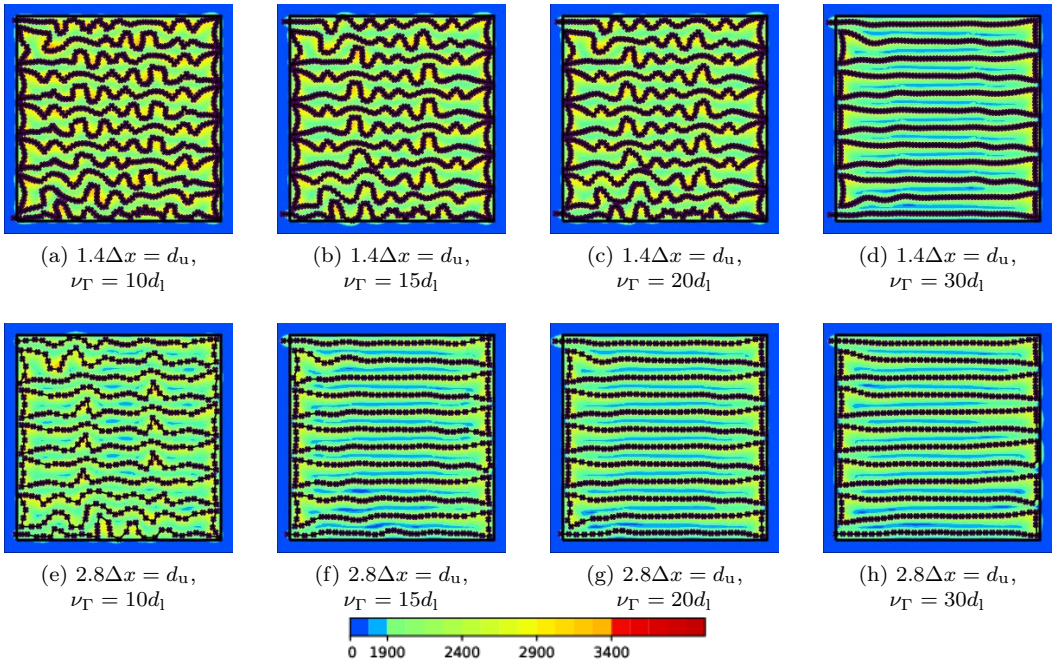
(a) $1.4\Delta x = d_\mathrm{u}$, $\nu_\Gamma = 10d_\mathrm{l}$ (b) $1.4\Delta x = d_\mathrm{u}$, $\nu_\Gamma = 15d_\mathrm{l}$ (c) $1.4\Delta x = d_\mathrm{u}$, $\nu_\Gamma = 20d_\mathrm{l}$ (d) $1.4\Delta x = d_\mathrm{u}$, $\nu_\Gamma = 30d_\mathrm{l}$

(e) $2.8\Delta x = d_\mathrm{u}$, $\nu_\Gamma = 10d_\mathrm{l}$ (f) $2.8\Delta x = d_\mathrm{u}$, $\nu_\Gamma = 15d_\mathrm{l}$ (g) $2.8\Delta x = d_\mathrm{u}$, $\nu_\Gamma = 20d_\mathrm{l}$ (h) $2.8\Delta x = d_\mathrm{u}$, $\nu_\Gamma = 30d_\mathrm{l}$

Figure 7.14: Final path and maximum temperature $\left(\max_{t\in(0,t_\mathrm{F})} y(t)\right)$ during the building depending on the discretization accuracy and on the regularization chosen (aluminum)

Case	t_F (s)	$\overline{C}_\phi$	$\overline{C}_{\mathrm{M,D_S}}$	$\overline{C}_{\mathrm{M,D\setminus D_S}}$
Initialization	7.098e−3	3.99e−5	3.02e−9	1.58e−4
$1.4\Delta x = d_\mathrm{u}$, $\nu_\Gamma = 10d_\mathrm{l}$	4.994e−3	1.25e−4	2.57e−9	1.33e−5
$1.4\Delta x = d_\mathrm{u}$, $\nu_\Gamma = 15d_\mathrm{l}$	4.987e−3	1.27e−4	2.68e−9	1.38e−5
$1.4\Delta x = d_\mathrm{u}$, $\nu_\Gamma = 20d_\mathrm{l}$	4.986e−3	1.28e−4	2.43e−9	1.25e−5
$1.4\Delta x = d_\mathrm{u}$, $\nu_\Gamma = 30d_\mathrm{l}$	5.002e−3	1.24e−4	2.99e−9	1.47e−5
$2.8\Delta x = d_\mathrm{u}$, $\nu_\Gamma = 10d_\mathrm{l}$	4.999e−3	1.43e−4	3.65e−11	8.38e−6
$2.8\Delta x = d_\mathrm{u}$, $\nu_\Gamma = 15d_\mathrm{l}$	4.999e−3	1.42e−4	2.95e−11	8.65e−6
$2.8\Delta x = d_\mathrm{u}$, $\nu_\Gamma = 20d_\mathrm{l}$	5.027e−3	1.33e−4	1.81e−11	1.06e−5
$2.8\Delta x = d_\mathrm{u}$, $\nu_\Gamma = 30d_\mathrm{l}$	5.083e−3	1.27e−4	5.03e−11	1.08e−5

Table 7.3: Comparison of the final cost and the constraints of the final results (aluminum).



(a) $1.4\Delta x = d_{\mathrm{u}}$, $\nu_\Gamma = 10 d_{\mathrm{l}}$ (b) $1.4\Delta x = d_{\mathrm{u}}$, $\nu_\Gamma = 15 d_{\mathrm{l}}$ (c) $1.4\Delta x = d_{\mathrm{u}}$, $\nu_\Gamma = 20 d_{\mathrm{l}}$ (d) $1.4\Delta x = d_{\mathrm{u}}$, $\nu_\Gamma = 30 d_{\mathrm{l}}$

(e) $2.8\Delta x = d_{\mathrm{u}}$, $\nu_\Gamma = 10 d_{\mathrm{l}}$ (f) $2.8\Delta x = d_{\mathrm{u}}$, $\nu_\Gamma = 15 d_{\mathrm{l}}$ (g) $2.8\Delta x = d_{\mathrm{u}}$, $\nu_\Gamma = 20 d_{\mathrm{l}}$ (h) $2.8\Delta x = d_{\mathrm{u}}$, $\nu_\Gamma = 30 d_{\mathrm{l}}$

Figure 7.15: Final path and maximum temperature $\left(\max_{t\in(0,t_{\mathrm{F}})} y(t)\right)$ during the building depending on the discretization accuracy and on the regularization chosen (titanium)

Case	t_{F} (s)	$\overline{C}_\phi$	$\overline{C}_{\mathrm{M,D_S}}$	$\overline{C}_{\mathrm{M,D}\setminus\mathrm{D_S}}$
Initialization	1.325e−2	2.66e−2	0.00	0.00
$1.4\Delta x = d_{\mathrm{u}}$, $\nu_\Gamma = 10 d_{\mathrm{l}}$	2.158e−2	3.99e−7	0.00	1.27e−6
$1.4\Delta x = d_{\mathrm{u}}$, $\nu_\Gamma = 15 d_{\mathrm{l}}$	2.075e−2	5.08e−5	0.00	2.73e−6
$1.4\Delta x = d_{\mathrm{u}}$, $\nu_\Gamma = 20 d_{\mathrm{l}}$	2.047e−2	6.31e−5	0.00	1.62e−6
$1.4\Delta x = d_{\mathrm{u}}$, $\nu_\Gamma = 30 d_{\mathrm{l}}$	1.632e−2	1.89e−3	0.00	1.12e−6
$2.8\Delta x = d_{\mathrm{u}}$, $\nu_\Gamma = 10 d_{\mathrm{l}}$	1.842e−2	6.28e−4	0.00	6.47e−7
$2.8\Delta x = d_{\mathrm{u}}$, $\nu_\Gamma = 15 d_{\mathrm{l}}$	1.632e−2	2.01e−3	0.00	6.77e−7
$2.8\Delta x = d_{\mathrm{u}}$, $\nu_\Gamma = 20 d_{\mathrm{l}}$	1.624e−2	2.04e−3	0.00	9.74e−7
$2.8\Delta x = d_{\mathrm{u}}$, $\nu_\Gamma = 30 d_{\mathrm{l}}$	1.612e−2	2.21e−3	0.00	2.09e−6

Table 7.4: Comparison of the final cost and the constraints of the final results (titanium).

We start by commenting the results in the titanium context. The comments are close to the steady state case ones. Indeed, from the left to the right, the regularization is increased. Thus the path is required to be smoother and the slight and rapid path variations present on the left columns disappear with the increase of ν_Γ. These also disappear when increasing the mean distance between discretization nodes. Indeed, loosening the path description accuracy leads to loosing freedom and thus getting rid of the small variations. These analysis are translated in terms of quantitative results: less regularization and more accuracy leads to bigger final times and smaller constraints. For the aluminum, whatever regularization is chosen, the results seem to be similar (but for a small variations for Figure 7.14(h)). Indeed, the conductivity being bigger, small path variations are not required.

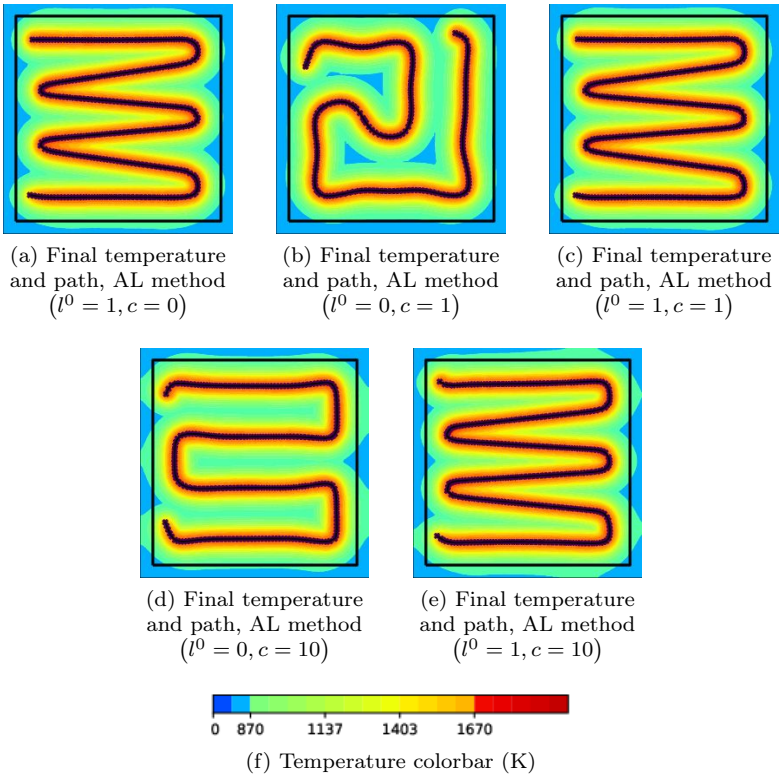
In the following, the regularization is fixed to $\nu_\Gamma = 20 d_{\mathrm{l}}$ and the accuracy to $d_{\mathrm{u}} = 1.4\Delta x$.

Impact of the optimization algorithm

As in the steady state case, the optimization algorithm is then considered, with height tests. Five of them consist in Augmented Lagrangian methods with different coefficients: the first one does not penalize ($c = \mu_{\mathrm{M,s}} = \mu_{\mathrm{M,e}} = 0$) and the Lagrange multipliers remain fixed during the process ($l_\phi = l_{\mathrm{M,s}} = l_{\mathrm{M,e}} = 1$). In the second and third tests, the penalization is increased to $c = \mu_{\mathrm{M,s}} = \mu_{\mathrm{M,e}} = 1$. In the third test, the Lagrange multipliers are initialized to 0 ($l_\phi = l_{\mathrm{M,s}} = l_{\mathrm{M,e}} = 0$) whereas they are initialized to 1 in the third test. The second test should thus lead to a larger time reduction since the constraints are not taken into account straight at the beginning of the process. In the fourth and fifth test, the penalization is again increased to $c = \mu_{\mathrm{M,s}} = \mu_{\mathrm{M,e}} = 10$. Once again, the multipliers in the fourth test are initialized to 0 whereas they are initialized to 1 in the fifth test with the same consequences. The different tests are recapped as

- Augmented Lagrangian method with $l^0 = 1, c = 0$,
- Augmented Lagrangian method with $l^0 = 0, c = 1$,
- Augmented Lagrangian method with $l^0 = 1, c = 1$,
- Augmented Lagrangian method with $l^0 = 0, c = 10$,
- Augmented Lagrangian method with $l^0 = 1, c = 10$,

The results in the aluminum case initialized by a zigzag with 6 lines (Figure 7.10(a)) are given by Figure 7.16 and the adimensionalized constraints in Table 7.5 whereas the results in the titanium case initialized by a zigzag with 12 lines (Figure 7.12) are given by Figure 7.17 and Table 7.6.



(a) Final temperature and path, AL method ($l^0 = 1, c = 0$)

(b) Final temperature and path, AL method ($l^0 = 0, c = 1$)

(c) Final temperature and path, AL method ($l^0 = 1, c = 1$)

(d) Final temperature and path, AL method ($l^0 = 0, c = 10$)

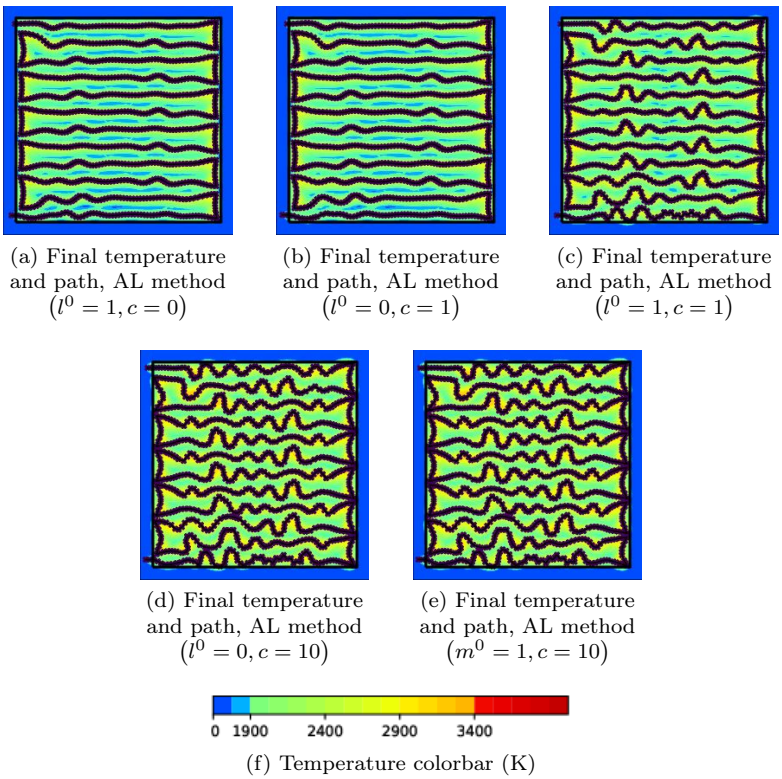
(e) Final temperature and path, AL method ($l^0 = 1, c = 10$)

(f) Temperature colorbar (K)

Figure 7.16: Path and maximum temperature $\left(\max_{t\in(0,t_{\mathrm{F}})} y(t)\right)$ during the building depending on the iterations (aluminum)

Case	t_F (s)	$\overline{C}_\phi$	$\overline{C}_{M,D_S}$	$\overline{C}_{M,D\setminus D_S}$
Initialization	7.098e−3	3.99e−5	3.02e−9	1.58e−4
AL met. test 1 ($l^0 = 1$, $c = 0$)	6.138e−3	3.39e−4	8.21e−9	1.85e−6
AL met. test 2 ($l^0 = 0$, $c = 1$)	4.321e−3	4.98e−4	2.24e−9	4.48e−7
AL met. test 3 ($l^0 = 1$, $c = 1$)	6.214e−3	2.82e−4	8.81e−9	2.28e−6
AL met. test 4 ($l^0 = 0$, $c = 10$)	4.986e−3	1.28e−4	2.43e−9	1.25e−5
AL met. test 5 ($l^0 = 1$, $c = 10$)	6.475e−3	1.38e−4	7.27e−9	6.46e−6

Table 7.5: Comparison of the cost and the constraints of the final results (aluminum)



(a) Final temperature and path, AL method $(l^0 = 1, c = 0)$

(b) Final temperature and path, AL method $(l^0 = 0, c = 1)$

(c) Final temperature and path, AL method $(l^0 = 1, c = 1)$

(d) Final temperature and path, AL method $(l^0 = 0, c = 10)$

(e) Final temperature and path, AL method $(m^0 = 1, c = 10)$

(f) Temperature colorbar (K)

Figure 7.17: Path and maximum temperature $\left(\max_{t\in(0,t_F)} y(t)\right)$ during the building depending on the iterations (titanium)

Case	t_F (s)	$\overline{C}_\phi$	$\overline{C}_{M,D_S}$	$\overline{C}_{M,D\setminus D_S}$
Initialization	1.325e−2	2.66e−2	0.00	0.00
AL met. test 1 ($l^0 = 1$, $c = 0$)	1.643e−2	1.64e−3	0.00	3.62e−8
AL met. test 2 ($l^0 = 0$, $c = 1$)	1.657e−2	1.60e−3	0.00	1.20e−7
AL met. test 3 ($l^0 = 1$, $c = 1$)	1.836e−2	5.44e−4	0.00	3.55e−7
AL met. test 4 ($l^0 = 0$, $c = 10$)	2.047e−2	6.31e−5	0.00	1.62e−6
AL met. test 5 ($l^0 = 1$, $c = 10$)	2.065e−2	4.85e−5	0.00	1.71e−6

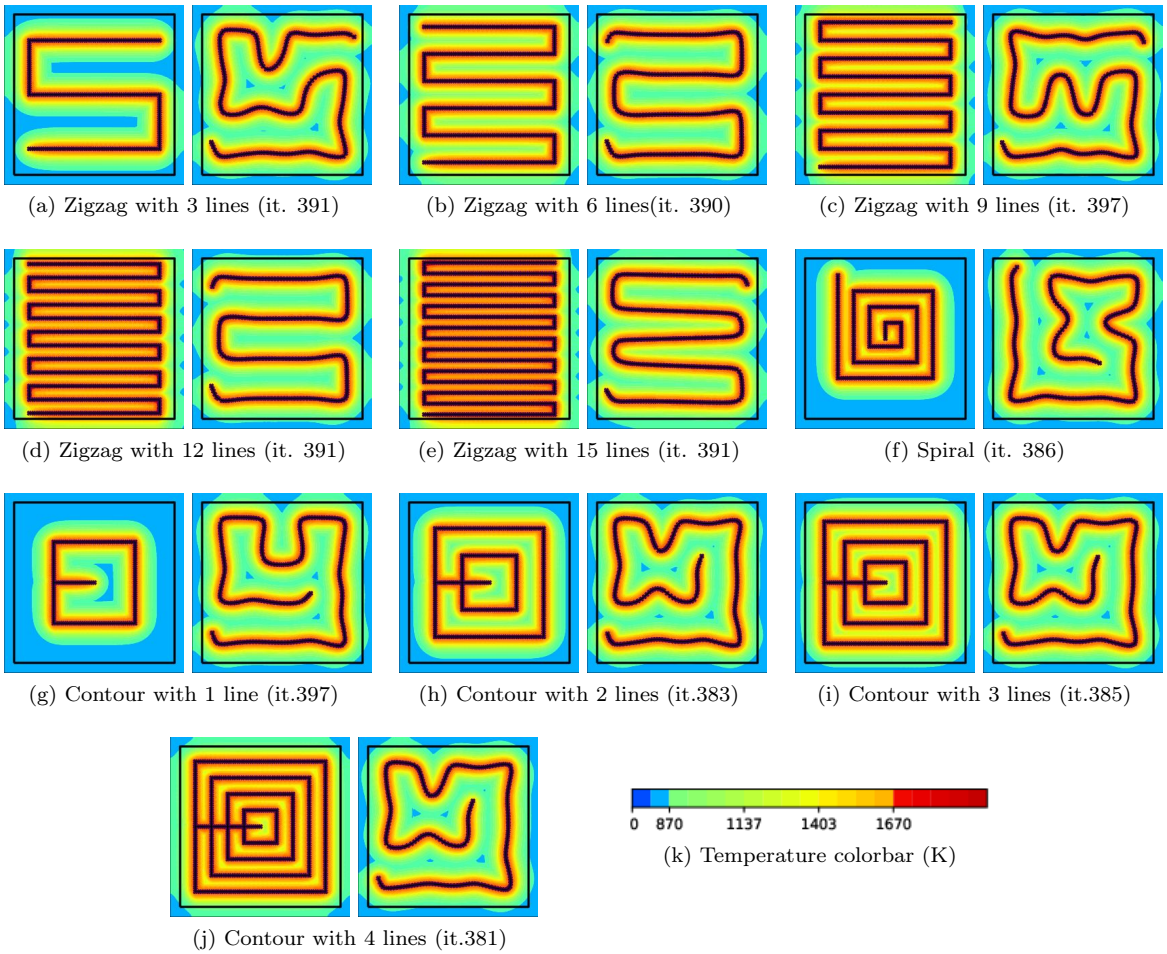
Table 7.6: Comparison of the cost and the constraints of the final results (titanium)

As in the steady state case, the algorithm highly impact the results. In the aluminum case first, three of the Augmented Lagrangian results are very similar. They correspond to initial Lagrange multiplier $l^0 = 1$, algorithms for which the constraints are taken into account since the very first iterations. Consequently, the final time for these paths are higher than the two others whereas the final constraints are similar. The diversity in the results is thus more restrained than in the steady state case: there might be less local minima in the transient case than in the steady state case. Testing different initializations should give more information about this. In the titanium case, the conclusions are slightly different. Indeed, whereas the important parameters seemed to be the initial Lagrange multiplier for the aluminum, the penalization coefficient c appears to have more impact for the titanium. Once again, three of the results seem to be quite similar, leading to the same assumption about local minima in the case of low conductivity material. The null space gradient could also be adapted here but this remains part of the perspectives.

In the following, the Augmented Lagrangian method with the multipliers initialized to 0 and the penalization set to 10 is chosen (Figures 7.10(d) and 7.12(d)).

7.3.5 Aluminium material test case

As in the steady state case, different initializations are now tested in the aluminum physical test case. The initializations and results are presented in Figure 7.18, the final adimensionalized values summed up in Table 7.7 and the graph of the final time with respect to the aggregated adimensionalized constraint (see (7.2.27)) $\overline{C} = \overline{C}_\phi + \overline{C}_{\mathrm{M,D_S}} + \overline{C}_{\mathrm{M,D\setminus D_S}}$ in each case is given in Figure 7.19.



(a) Zigzag with 3 lines (it. 391)
(b) Zigzag with 6 lines(it. 390)
(c) Zigzag with 9 lines (it. 397)
(d) Zigzag with 12 lines (it. 391)
(e) Zigzag with 15 lines (it. 391)
(f) Spiral (it. 386)
(g) Contour with 1 line (it.397)
(h) Contour with 2 lines (it.383)
(i) Contour with 3 lines (it.385)
(j) Contour with 4 lines (it.381)
(k) Temperature colorbar (K)

Figure 7.18: Initial and final path and maximum temperature $\left(\max_{t\in(0,t_\mathrm{F})} y(t)\right)$ during the building depending on the initialization (aluminum)

Case	$t_{\mathrm{F}}^{\mathrm{ini}}$ (s)	$t_{\mathrm{F}}^{\mathrm{fin}}$ (s)	$\overline{C}_{\phi}^{\mathrm{ini}}$	$\overline{C}_{\phi}^{\mathrm{fin}}$	$\overline{C}_{\mathrm{M,D_S}}^{\mathrm{ini}}$	$\overline{C}_{\mathrm{M,D_S}}^{\mathrm{fin}}$	$\overline{C}_{\mathrm{M,D\setminus D_S}}^{\mathrm{ini}}$	$\overline{C}_{\mathrm{M,D\setminus D_S}}^{\mathrm{fin}}$
zigzag 3 lines	3.864e−3	4.958e−3	1.30e−3	1.71e−4	1.07e−9	3.18e−9	2.39e−5	1.05e−5
zigzag 6 lines	7.098e−3	4.986e−3	3.99e−5	1.28e−4	3.02e−9	2.43e−9	1.58e−4	1.25e−5
zigzag 9 lines	1.019e−2	5.030e−3	9.85e−6	1.53e−4	6.24e−9	4.59e−9	6.06e−4	8.70e−6
zigzag 12 lines	1.325e−2	5.025e−3	4.18e−6	1.15e−4	1.22e−8	2.63e−9	9.34e−4	1.64e−5
zigzag 15 lines	1.630e−2	5.765e−3	1.99e−6	1.06e−4	1.59e−8	3.06e−9	1.12e−3	1.95e−5
spiral	5.635e−3	4.930e−3	3.16e−3	1.75e−4	4.51e−9	4.95e−9	0.00	3.98e−6
contour 1 line	2.835e−3	4.981e−3	4.31e−3	1.48e−4	4.74e−9	3.68e−9	0.00	1.44e−5
contour 2 lines	5.460e−3	4.916e−3	1.02e−3	1.80e−4	5.57e−9	4.35e−9	0.00	3.83e−6
contour 3 lines	8.033e−3	4.910e−3	1.12e−4	1.83e−4	5.43e−9	3.86e−9	9.30e−7	4.20e−6
contour 4 lines	1.058e−2	4.909e−3	1.06e−5	1.82e−4	4.58e−9	4.10e−9	4.46e−5	4.36e−6

Table 7.7: Comparison of the final cost and constraints of the final results (aluminum)

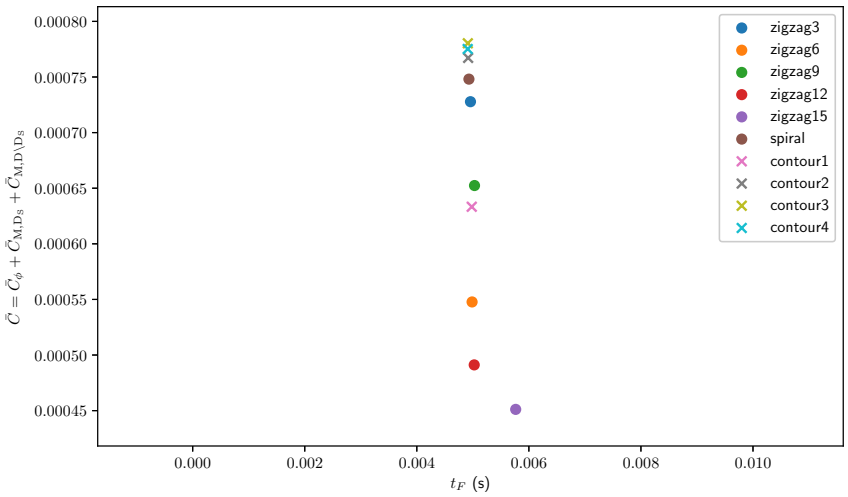


Figure 7.19: Constraint $\overline{C} = \overline{C}_{\phi} + \overline{C}_{\mathrm{M,D_S}} + \overline{C}_{\mathrm{M,D\setminus D_S}}$ with respect to the final time t_{F} (s) for the final results (aluminum)

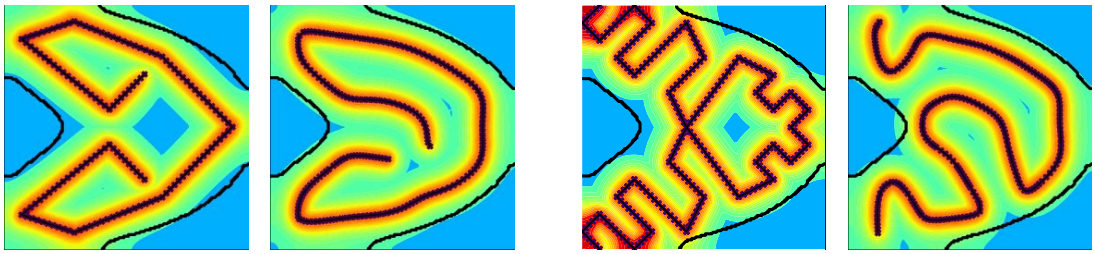
The first conclusion to these different tests is that the algorithm point based gives quite good results, far better than the angle based algorithm. Indeed, but for the result coming out from the zigzag with 15 lines, which final time is higher, each of the others appear quite intuitive and the final quantities are quite alike. In particular, the final time belongs mostly to $[4.9ms, 5.1ms]$. Thus, in the transient case again, the notion of specific final time or energy can be introduced. On the contrary to the steady state case, the final shapes are not all different. Indeed, the contour with 2, 3 and 4 lines lead to the same final path, which can be compared the result from the spiral to which a rotation of $\pi/2$ would have been applied. The results from the zigzag with 6 and 12 lines also appear the same. Finally, the results from the contour with 1 line, the zigzag with 3 lines and the zigzag 9 lines lead to different results however in line with the ones found with the other initializations (see Figure 7.19). The number of local minima seems to be smaller than in the steady state case but still well reached by the algorithm.

In each of these cases, the phase constraint appears not fully satisfied: for all the paths found, it remains unmelted zones. This is proscribed by industrial applications which would rather increase the maximum temperature inside the part to build than leave powder unmelted. Thus the algorithm chosen should favor the phase constraint. Yet, this must be done carefully. Indeed, the phase constraint is only

approximated and undervalued: it is already slightly favored. Unbalancing the different constraints could then result in large increase of the maximum temperature constraints. This could be improved by using another optimization algorithm than the Augmented Lagrangian method (null space gradient algorithm for example) but this further study remains part of the perspectives.

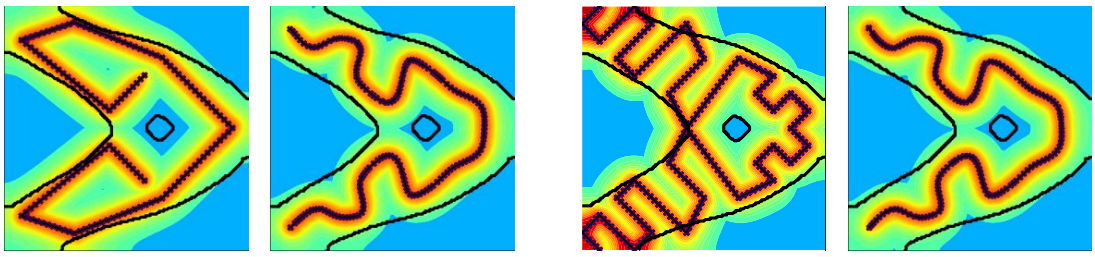
Scanning complex geometries

Finally, to get intuition on design criteria adapted to scanning constraints, for each of the object resulting from the shape optimization detailed in Section 3.3.2 and given in Figures 3.6, 3.7 and 3.8, the optimization is run for two different initializations. The results for each object are presented by Figures 7.20, 7.21, 7.22, the graph shown in Figure 7.23 and Table 7.8.



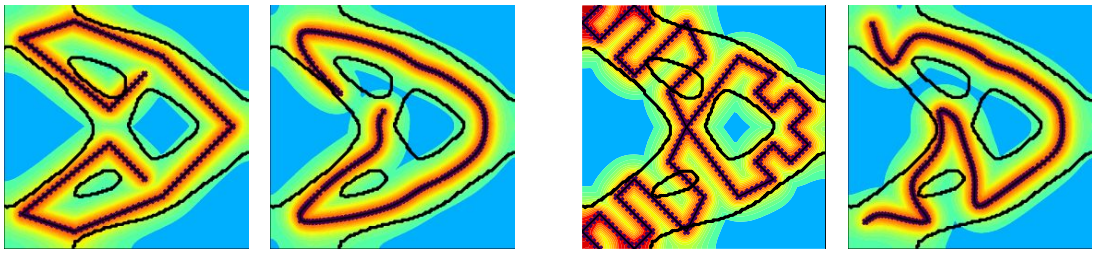
(a) Initialization 1 (it. 379) (b) Initialization 2 (it. 395)

Figure 7.20: Initial and final path and maximum temperature $\left(\max_{t\in(0,t_F)} y(t)\right)$ during the building depending on the initialization for the zero hole object (aluminum)



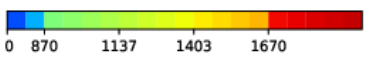
(a) Initialization 1 (it. 409) (b) Initialization 2 (it. 410)

Figure 7.21: Initial and final path and maximum temperature $\left(\max_{t\in(0,t_F)} y(t)\right)$ during the building depending on the initialization for the one hole object (aluminum)



(a) Initialization 1 (it. 386) (b) Initialization 2 (it. 397)

Figure 7.22: Initial and final path and maximum temperature $\left(\max_{t\in(0,t_F)} y(t)\right)$ during the building depending on the initialization for the three holes object (aluminum)



Temperature colobar

Case	$t_{\mathrm{F}}^{\mathrm{ini}}$ (s)	$t_{\mathrm{F}}^{\mathrm{fin}}$ (s)	$\overline{C}_{\phi}^{\mathrm{ini}}$	$\overline{C}_{\phi}^{\mathrm{fin}}$	$\overline{C}_{\mathrm{M,D_S}}^{\mathrm{ini}}$	$\overline{C}_{\mathrm{M,D_S}}^{\mathrm{fin}}$	$\overline{C}_{\mathrm{M,D\backslash D_S}}^{\mathrm{ini}}$	$\overline{C}_{\mathrm{M,D\backslash D_S}}^{\mathrm{fin}}$
Zero hole								
initialization 1	4.687e−3	4.449e−3	6.75e−4	1.40e−4	1.46e−8	3.61e−9	1.34e−6	1.68e−7
initialization 2	7.433e−3	4.664e−3	4.23e−4	1.62e−4	4.63e−6	5.24e−9	1.59e−5	1.17e−5
One hole								
initialization 1	4.687e−3	3.534e−3	1.47e−4	2.54e−4	1.88e−8	7.32e−9	2.45e−4	6.52e−7
initialization 2	7.433e−3	3.510e−3	4.50e−6	2.65e−4	5.95e−6	6.90e−9	3.39e−4	5.04e−7
Three holes								
initialization 1	4.687e−3	3.787e−3	2.48e−5	1.48e−4	1.97e−8	8.21e−9	5.51e−4	9.47e−5
initialization 2	7.433e−3	3.918e−3	2.81e−5	2.17e−4	6.23e−6	1.90e−8	7.97e−4	2.80e−4

Table 7.8: Comparison of the final cost and constraints of the final results for the zero, one and three holes objects (aluminum)

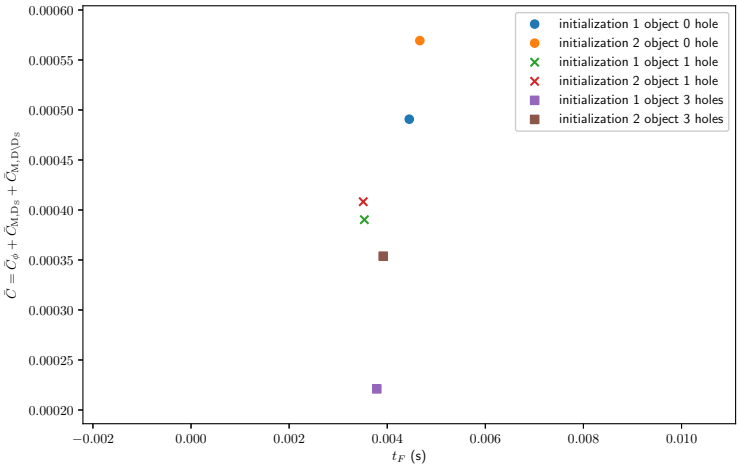
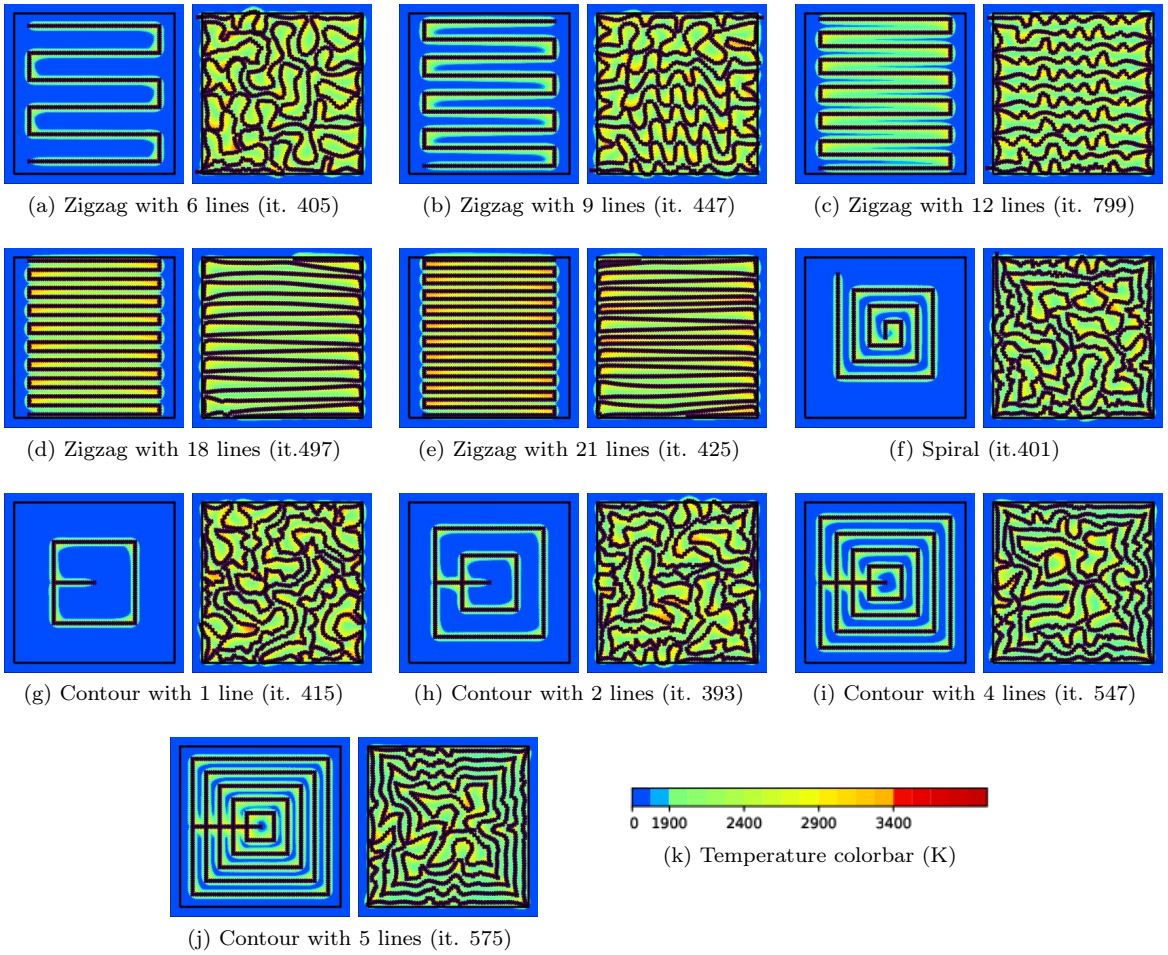


Figure 7.23: Constraint $\overline{C} = \overline{C}_{\phi} + \overline{C}_{\mathrm{M,D_S}} + \overline{C}_{\mathrm{M,D\backslash D_S}}$ with respect to the final time t_{F} (s) for the final results for the zero, one and three hole objects (aluminum)

The results from path optimization for a specific geometry confirm the previous remarks. First of all, at least for the two first objects, the algorithm leads to correct results and the final time values for the different initializations are similar. As for the third object, the optimization is very hard: the aluminum conductivity is high and the shape appears very difficult to build while not melting the holes. In the steady state case, this optimization was possible (see Figure 6.28). This steady state case is calibrated to be the maximal temperature over the building. Since the optimization is possible in this steady state case and not in the transient one, the calibration process should be improved. Indeed, if leading to results that can be interpreted in a physical way, it remains not perfect and based on model full of assumptions. This algorithm could then be applied to an improved model. The development of this model as well as the acceleration of the algorithm this will require are part of the perspectives.

7.3.6 Titanium material test case

We now consider the titanium test case, which lower conductivity complicates the optimization process. The initializations and results are presented in Figure 7.24, the final values summed up in Table 7.9 and the graph of the final time with respect to the aggregated adimensionalized constraint $\overline{C}$ shown in Figure 7.25.



(a) Zigzag with 6 lines (it. 405)
(b) Zigzag with 9 lines (it. 447)
(c) Zigzag with 12 lines (it. 799)
(d) Zigzag with 18 lines (it.497)
(e) Zigzag with 21 lines (it. 425)
(f) Spiral (it.401)
(g) Contour with 1 line (it. 415)
(h) Contour with 2 lines (it. 393)
(i) Contour with 4 lines (it. 547)
(j) Contour with 5 lines (it. 575)
(k) Temperature colorbar (K)

Figure 7.24: Initial and final path and maximum temperature $\left(\max_{t\in(0,t_F)} y(t)\right)$ during the building depending on the initialization (titanium)

Case	t_F^{ini} (s)	t_F^{fin} (s)	$\overline{C}_\phi^{ini}$	$\overline{C}_\phi^{fin}$	$\overline{C}_{M,D_S}^{ini}$	$\overline{C}_{M,D_S}^{fin}$	$\overline{C}_{M,D\setminus D_S}^{ini}$	$\overline{C}_{M,D\setminus D_S}^{fin}$
zigzag 6 lines	7.098e−3	2.235e−2	8.37e−2	2.06e−8	0.00	0.00	0.00	1.48e−5
zigzag 9 lines	1.019e−2	2.277e−2	4.06e−2	9.17e−7	0.00	0.00	0.00	1.60e−5
zigzag 12 lines	1.325e−2	2.047e−2	2.66e−2	6.31e−5	0.00	0.00	0.00	1.62e−6
zigzag 18 lines	1.933e−2	2.254e−2	2.24e−2	3.80e−6	0.00	0.00	6.30e−6	1.53e−5
zigzag 21 lines	2.237e−2	2.530e−2	2.20e−2	3.29e−6	0.00	0.00	1.80e−5	9.75e−6
spiral	5.635e−3	2.317e−2	1.65e−1	3.46e−5	0.00	0.00	0.00	3.25e−6
contour 1 line	2.835e−3	2.329e−2	2.25e−1	1.41e−8	0.00	0.00	0.00	1.11e−5
contour 2 lines	5.460e−3	2.332e−2	1.41e−1	7.44e−6	0.00	0.00	0.00	1.33e−5
contour 4 lines	1.058e−2	2.353e−2	4.80e−2	8.40e−5	0.00	0.00	0.00	2.86e−6
contour 5 lines	1.312e−2	2.316e−2	2.72e−2	1.40e−4	0.00	0.00	0.00	1.31e−6

Table 7.9: Comparison of the final cost and constraints of the final results (titanium)

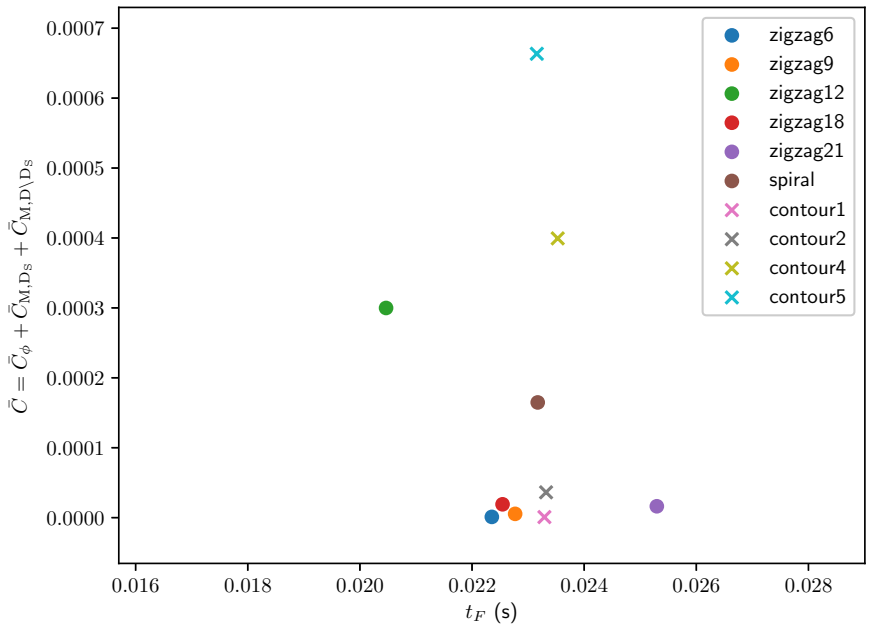


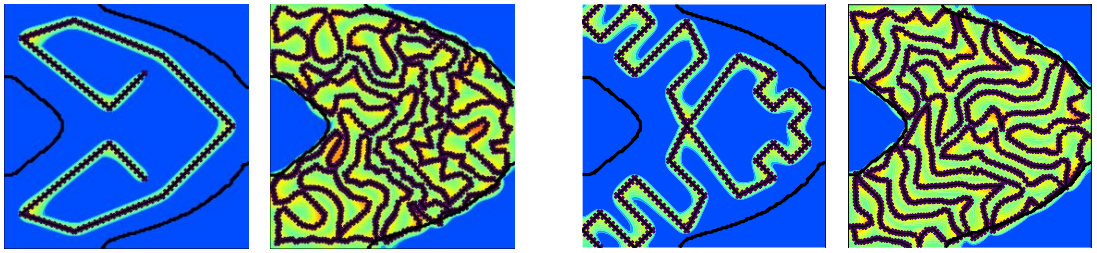
Figure 7.25: Constraint $\overline{C} = \overline{C}_\phi + \overline{C}_{\mathrm{M,D_S}} + \overline{C}_{\mathrm{M,D\backslash D_S}}$ with respect to the final time t_{F} (s) for the final results (titanium)

The results in the titanium test case are a lot more intricate than in the aluminum case, and the decrease of the number of local minima can not be concluded here. Yet, the optimization process seems to work pretty well with constraints satisfied and final times that get quite close to one another even if starting from very different initializations. Indeed, but for 2 of them, the final final times all belong to $[22.3ms, 23.5ms]$. The two exceptions are for the zigzag with 21 lines and the zigzag with 12 lines. The first one has a very high initial final time and the algorithm does not manage to reduce enough the path length. The second initialization seems quite close to a good optimum and does not require much path modifications. Thus, the idea of specific energy is still valid when considering a material with lower conductivity.

In order to facilitate the optimization and obtain results that could be applied in the industry, the optimization should be run on smaller zones, thus reducing the number of local minima. We could then work on the expansion of these results for small domains to larger by concurrent optimization between several domains or symmetry conditions. These developments are part of the perspectives.

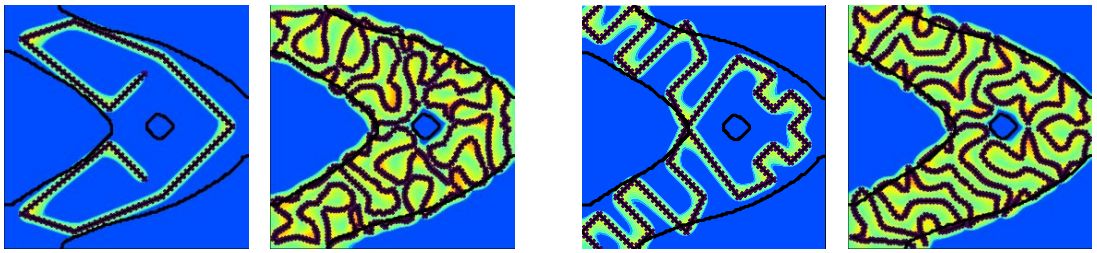
Scanning complex geometries

Finally, as for the aluminum, for each of the objects given in Figures 3.6, 3.7 and 3.8, the optimization is run for two different initializations. The results for each object are presented by Figures 7.26, 7.27, 7.28, by the graph shown in Figure 7.29 and Table 7.10.



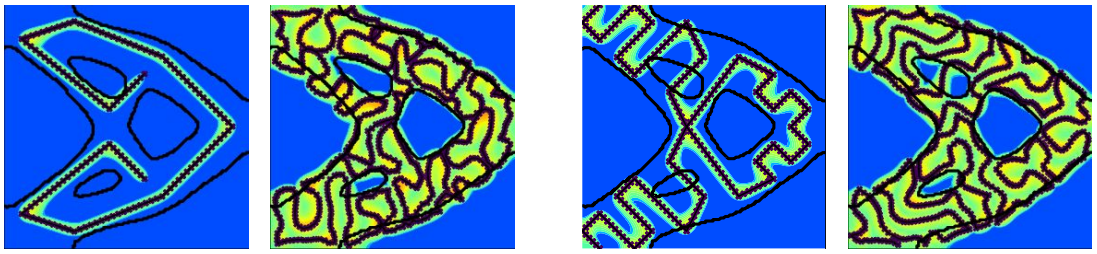
(a) Initialization 1 (it. 499) (b) Initialization 2 (it. 458)

Figure 7.26: Path and maximum temperature $\left(\max_{t\in(0,t_F)} y(t)\right)$ during the building depending on the initialization for the zero hole object (titanium)



(a) Initialization 1 (it. 419) (b) Initialization 2 (it. 466)

Figure 7.27: Path and maximum temperature $\left(\max_{t\in(0,t_F)} y(t)\right)$ during the building depending on the initialization for the one hole object (titanium)



(a) Initialization 1 (it. 417) (b) Initialization 2 (it. 459)

Figure 7.28: Initial and final path and maximum temperature $\left(\max_{t\in(0,t_F)} y(t)\right)$ during the building depending on the initialization for the three holes object (titanium)

Temperature colobar

Case	$t_{\mathrm{F}}^{\mathrm{ini}}$ (s)	$t_{\mathrm{F}}^{\mathrm{fin}}$ (s)	$\overline{C}_{\phi}^{\mathrm{ini}}$	$\overline{C}_{\phi}^{\mathrm{fin}}$	$\overline{C}_{\mathrm{M,D_S}}^{\mathrm{ini}}$	$\overline{C}_{\mathrm{M,D_S}}^{\mathrm{fin}}$	$\overline{C}_{\mathrm{M,D\backslash D_S}}^{\mathrm{ini}}$	$\overline{C}_{\mathrm{M,D\backslash D_S}}^{\mathrm{fin}}$
Zero hole								
initialization 1	4.687e−3	2.213e−2	1.55e−1	1.06e−9	0.00	0.00	0.00	1.12e−5
initialization 2	7.433e−3	2.046e−2	1.00e−1	2.14e−8	0.00	0.00	0.00	1.36e−5
One hole								
initialization 1	4.687e−3	1.650e−2	1.19e−1	1.03e−8	0.00	0.00	0.00	1.01e−5
initialization 2	7.433e−3	1.540e−2	5.59e−2	2.99e−7	0.00	0.00	2.74e−7	1.47e−5
Three holes								
initialization 1	4.687e−3	1.658e−2	1.13e−1	3.73e−9	0.00	0.00	7.17e−6	4.38e−5
initialization 2	7.433e−3	1.545e−2	5.84e−2	3.40e−7	0.00	0.00	2.08e−5	5.45e−5

Table 7.10: Comparison of the final cost and constraints of the final results for the zero, one and three holes objects (titanium)

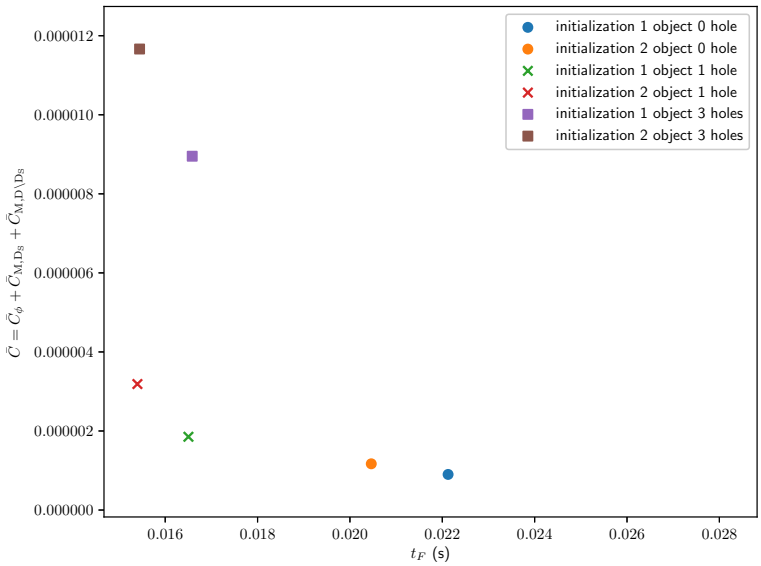


Figure 7.29: Constraint $\overline{C} = \overline{C}_{\phi} + \overline{C}_{\mathrm{M,D_S}} + \overline{C}_{\mathrm{M,D\backslash D_S}}$ with respect to the final time t_{F} (s) for the final results (titanium)

The results for each of the objects are better than for the aluminum. Indeed, the shape is better respected while keeping the maximum temperature constraint low. This conclusion was expected. Since the titanium's conduction is lower than the aluminum's, building thin bars is easier which explains the results obtained for the object with three holes. However, the final paths are once again too complicated to be directly applied in the industry: the curvature should be controlled and we could split the object to work on smaller domains thus reducing the number of local minima. These adaptations to further industrial requirements are part of the perspectives.

7.3.7 Conclusion

The modification of the path discretization leads to better results. Indeed, the issue of final time is easier to deal with: because the path elements can have different lengths, the points can simply move in the working domain, allowing for the line lengthening and shrinking at correct locations. The information is fully contained by the derivatives which constitutes a huge improvement compared to the element

addition in the angle based formulation. The code thus converges in a reasonable number of iterations (usually less than 500). Yet, the main drawback is the computational time (Figures 7.11(e) and 7.13(e)). Indeed, this process is close to brute force algorithms, requiring the inversion of as many linear problems than points in the path discretization.

7.4 Conclusion

Among the two discretizations, the point based one seems to give results far better than the angle based one. Indeed, the final time issue is naturally dealt with and no artificial methods are required to lengthen or shrink the path. Yet, the method then gets closer to a brute force process, increasing the computational costs. Indeed, the constant element size property cannot be kept and the simulation of the building requires the inversion of the number of discretization elements linear systems.

Besides the computational costs, the point based discretization optimization gives interesting results. In the aluminum case first, the path adapts very well to the shape to build. The final paths all induce the same final times and constraints, expanding the notion of optimal energy to the transient case. As for the diversity of the optimized path, it seems reduced with respect to the steady state case. Yet, no conclusion can be taken regarding a potential decrease of the number of local minima in the transient case. The model has been calibrated for the titanium and might require some further modifications in the aluminum case. Indeed, less energy seems to be required in the transient case than in the steady state case even though the steady state case considers the maximum temperature over the building. As for the titanium context, the point based discretization seems quite efficient. Unlike the angle based discretization, the final algorithm managed to adapt the path in order to build the square with final quantitative values confirming the notion of optimal energy. Yet, the results could be improved for the building of more complicated objects with a deeper study of the different optimization algorithms available. Finally, the model once again seems to impact the results. Yet, on the contrary to the aluminum, more energy seems to be required in the transient case than in the steady state one. The results obtained remain dependent on the initialization but the modifications induced by the optimization are large enough to make the results really different from the initialing pattern. This method is thus interesting to produce new path shapes and get intuition on the notion of "good path".
Besides the improvements of the optimization algorithms and of the model, these first results lead to different perspectives. Following the remarks stated in the steady state case, it would be very interesting to allow several path connected components. Indeed, especially for intricate objects, this could help the path design to remain within the part without overheating. Then, this study could be carried on to better understand the link between the part shape and the path.

CHAPTER 8

MODIFICATION OF THE PATH'S TOPOLOGY

Contents

8.1 Introduction . . . 179
8.2 Controlling the power . . . 180
8.2.1 Relaxation penalization methods . . . 180
8.2.2 Controlling the jumps . . . 181
8.2.3 Optimization algorithm . . . 183
8.3 Application of the power control in the steady state context . . . 184
8.3.1 Path discretization . . . 185
8.3.2 Derivative computations . . . 186
8.3.3 Optimization algorithm . . . 189
8.3.4 Numerical results: optimization of the power . . . 190
8.3.5 Numerical results of the relaxation penalization problem with jump control . . . 192
8.3.6 Conclusion . . . 197
8.4 Topology optimization of the path (steady state context) . . . 198
8.4.1 A notion of topological derivative for lines in the plane . . . 198
8.4.2 Numerical algorithm to introduce path splittings . . . 202
8.4.3 Numerical algorithm to add paths connected components . . . 206
8.4.4 Numerical results . . . 210
8.5 Further comparisons between the coupled power and path optimization and the topology optimization methods . . . 214
8.5.1 Square geometry, aluminium context . . . 214
8.5.2 Square geometry, titanium context . . . 218
8.5.3 Comparison between the coupled power and path optimization and the topology optimization methods for complex geometries . . . 221
8.6 Conclusion . . . 228

8.1 Introduction

As shown in the steady state framework, the efficiency of the optimal path depends on the number of curves composing the path, and allowing points at which the source is switched off could improve the scanning. Even though its slows down the process (see Chapter 1), this option is already involved in many scanning strategies (raster paths for example, see Chapter 1). This feature is clearly relevant at an industrial level and extends the set of admissible paths. In this chapter, we propose strategies to vary both the path shape and topology, that is its number of connected components. Based on the algorithms already developed, two different ideas are presented.

The first one is based on the real process modeling. A new optimization variable, written as *power variable*, is added. Multiplied to the power, it can take only two values: 0 when the source is off and 1 otherwise. Section 8.3 presents the theoretical tools required to include this feature to the algorithms we developed. First, a modified optimization problem is stated to include the additional power variable. However, the set $\{0, 1\}$ is discrete which complicates the optimization. To address this issue, a relaxation penalization method (very similar to the SIMP method in topology optimization [32]) is proposed leading to a new optimization algorithm. If these settings allow for switching off the source, they do not control the number of switching points at which this must be done. To prevent degenerated paths with an very large number of switching, we complement this algorithm by a control of the power variable oscillation. To this end, a total variation constraint, inspired from image segmentation issues, is added. In Section 8.3, we apply these tools to the path optimization algorithm proposed in the steady state context (Chapter 6). We conduct numerical computations that enable for a critical analysis of the approach.

The work presented in these sections results from a joined work with Tonia-Maria ALAM and Serge NICAISE, from the Laboratoire de Mathématiques et leurs Applications de Valenciennes (LAMAV), in Valenciennes, France. Based on two different path optimization approaches, a common work has been conducted on the different available techniques to optimize the power, force its bang-bang properties and control its number of variations. A theoretical study of these tools' ability to provide an efficient control has then been kept on developing by the LAMAV team, which results can be found in [6]. The work presented here focuses on the numerical implementation. For further information about the theory and to consider a second numerical approach for path optimization, the reader is warmly invited to further look into the LAMAV's work and especially [7] and [6].

The second idea consists in modifying the path topology itself, in the steady state context. Topology optimization, mentioned in Chapter 3, aims at modifying the topology of the optimized structure. As for shape differentiation, the sensitivity of the problem to the creation of an infinitesimal hole is computed. This sensitivity is then transformed into a descent direction and the structure is modified. In Section 8.4, we apply this to path optimization.

8.2 Controlling the power

The objective is to include in the optimization the ability to switch the source on and off. To this aim, an additional optimization variable $\zeta : [0, t_{\mathrm{F}}] \to \{0, 1\}$ is introduced. The source is off whenever $\zeta = 0$, on for $\zeta = 1$.

The optimization problem now involves two different variables: the path $\Gamma \in \mathcal{G}$, and a power variable $\zeta \in L^2([0, t_{\mathrm{F}}], \{0, 1\})$. Thus, we can rewrite the heat equation (4.2.6) as:

$$\begin{cases} \rho c_p \partial_t y - \nabla(\lambda \nabla y) + \beta(y - y_{\mathrm{ini}}) = \zeta P \exp\left(-\dfrac{(x-u)^2}{r^2}\right) & (t,x) \in [0,t_{\mathrm{F}}] \times D \\ \lambda \partial_n y = 0 & (t,x) \in [0,t_{\mathrm{F}}] \times \partial D \\ y(0,x) = y_{\mathrm{ini}}(x) & x \in D, \end{cases} \tag{8.2.1}$$

where u describes the path Γ on $[0, t_{\mathrm{F}}]$. The new optimization problem is then (see Chapters 4 and 6),

$$\min_{\Gamma \in \mathcal{G}, \zeta \in L^2([0,t_{\mathrm{F}}],\{0,1\})} L_{\mathrm{F}} = V t_{\mathrm{F}}, \quad \text{such that} \begin{cases} C_\phi(\Gamma, \zeta) = C_{\mathrm{M}}(\Gamma, \zeta) = 0, \\ y \in L^2\left([0,t_{\mathrm{F}}], H^1(D)\right) \text{ solution of (8.2.1)} \\ \text{dependence on } \Gamma \text{ and } \zeta \text{ in the source term} \end{cases} \tag{8.2.2}$$

Remark 8.1. Note that both optimization variables are independent. Indeed, the power variable ζ is actually a one dimensional function than can be defined on the segment $[0, t_{\mathrm{Final}}]$, with $t_{\mathrm{Final}} > t_{\mathrm{F}}$. This function only depends on time and not on the path point. Because the source $q(t > t_{\mathrm{F}}) = 0$, $\zeta(t > t_{\mathrm{F}})$ is not involved in the optimization process. This independence is crucial for differentiation properties and the introduction of total variation constraints into the optimization problem (see Section 8.2.2).

8.2.1 Relaxation penalization methods

Because the variable ζ cannot vary continuously ($\zeta \in \{0, 1\}$), differentiating is complicated. In order to deal with this, a relaxation-penalization approach is chosen. The relaxation process corresponds to enlarging the admissible values for the variable ζ through the replacement of the set $\{0, 1\}$ by the segment $[0, 1]$. To force the variable $\zeta(t)$ to its extremal values 0 and 1 during the optimization process, intermediate values are penalized.

Relaxation-penalization processes arise in many different fields. In shape and topology optimization, the domain can be represented by a density variable ρ where $\rho(x) = 0$ in case of void and $\rho(x) = 1$ if solid (this domain representation is an alternative choice to the level set method detailed in Chapter 3). A relaxation-penalization method is applied. In a first approach, the density is first relaxed to the interval $[0, 1]$ until convergence is reached. A filter is then applied to penalize the intermediate densities (for example, one could take $\rho_{\mathrm{filtered}} = \frac{1-\cos(\pi\rho_{\mathrm{opt}})}{2}$ [9]). This technique is well described and documented in

[8, 9].

In a second approach, called the SIMP method, the density ρ is still the optimization variable. However, the actual density involved in the state partial differential equation is ρ^q ($q > 1$), squeezing any density smaller than 1 to 0. A volume constraint is added to the problem and usually, to ease the optimization, continuation algorithms, where the power q is increased along the process are used. The SIMP method is fully detailed in [32].

To keep it simple, we choose here to penalize the intermediate values by adding to the optimization problem the following constraint $\mathcal{P}$ on the variable ζ (equality constraint $\mathcal{P} = 0$):

$$\mathcal{P} = \frac{1}{t_{\mathrm{F}}} \int_0^{t_{\mathrm{F}}} f_{\mathrm{pen}}(\zeta(t))dt. \tag{8.2.3}$$

The function f_{pen} must penalize the intermediate values of ζ. The most natural choice is:

$$f_{\mathrm{pen},q_{\mathcal{P}}} : \left\{ \begin{array}{ccc} [0,1] & \to & \mathbb{R} \\ \zeta & \mapsto & \left(\zeta\left(1-\zeta\right)\right)^{\frac{1}{q_{\mathcal{P}}}}, \end{array} \right. \tag{8.2.4}$$

with $q_{\mathcal{P}} \geq 1$ (Figure 8.1). To ensure the differentiability at 0 and 1 (when $q_{\mathcal{P}} > 1$), this function is replaced by $f_{\mathrm{pen},q_{\mathcal{P}}}(\zeta) = \left(\left(\zeta + \epsilon_{\mathcal{P}}\right)\left(1 + \epsilon_{\mathcal{P}} - \zeta\right)\right)^{\frac{1}{q_{\mathcal{P}}}}$, with $\epsilon_{\mathcal{P}}$ arbitrarily fixed to $\epsilon_{\mathcal{P}} = 1\mathrm{e}{-6}$. Such a function however presents two main drawbacks. First, the derivative of the integrand at $\frac{1}{2}$ equals 0 and the algorithm can be stuck there. Then, the derivative tends to push any value under 0.5 to 0 whereas any value above 0.5 goes to 1. The first iterations of the algorithm are then crucial since alternating between both zones is complicated. A continuation method could be used to limit this drawback.

To address these issues, a second function, inspired by the SIMP method [32], is considered:

$$f_{\mathrm{SIMP}} : \left\{ \begin{array}{ccc} [0,1] & \to & \mathbb{R} \\ \zeta & \mapsto & \zeta^{\frac{1}{q_{\mathcal{P}}}}, \end{array} \right. \tag{8.2.5}$$

with $q_{\mathcal{P}} \geq 1$ (Figure 8.2). Again, this function is modified to $\left(\zeta + \epsilon_{\mathcal{P}}\right)^{\frac{1}{q_{\mathcal{P}}}}$ to insure the differentiability at 0. This function is very steep at 0 and almost flat at 1. The penalization cost is thus similar at $x = 1$ or at intermediate value. Assume that the problem only involves a phase constraint and no maximum temperature constraints. Then, if energy has to be introduced at a point x, it is worth setting the power variable to 1. Indeed, the penalization function is almost the same for any power variable above 0 and it is more efficient to set the power variable at its maximum to decrease the phase constraint as much as possible. Yet, in the problem we consider here, there is a maximum temperature constraint that could act against this penalization.

Both functions have been numerically evaluated in the steady state context. However, it appears that the second one does not provide interesting result and in all the numerical tests presented in this chapter, the penalization amounts to add the constraint

$$\mathcal{P} = \frac{1}{t_{\mathrm{F}}} \int_0^{t_{\mathrm{F}}} f_{\mathrm{pen}}(\zeta(t))dt = 0. \tag{8.2.6}$$

Remark 8.2. Note that this new penalization constraint is "artificial", namely a purely numerical trick that is not related to the physics of the problem. Correctly involving it in the optimization process requires a careful choice of the optimization parameters.

8.2.2 Controlling the jumps

If switching off and on the source may improve the scanning strategy, it is also time consuming and the number of times it happens (written as power jumps in the following) must be limited. To prevent these oscillations, the variations of ζ must be properly defined and controlled.

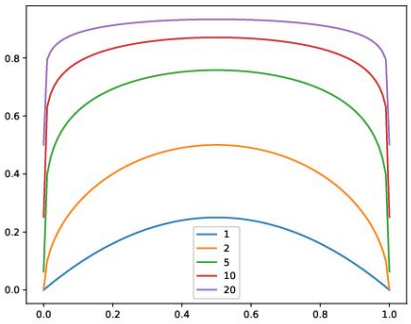


Figure 8.1: Symmetric penalization $f_{\text{pen}} : \zeta \to \left((\zeta + \epsilon_{\mathcal{P}})\left(1 + \epsilon_{\mathcal{P}} - \zeta\right)\right)^{1/q_{\mathcal{P}}}$

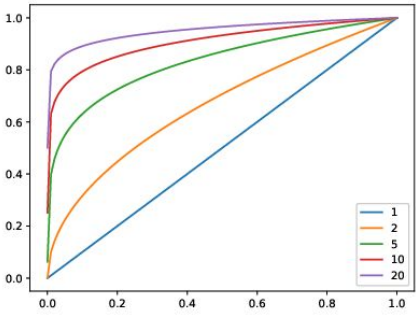


Figure 8.2: SIMP penalization $f_{\text{SIMP}} : \zeta \to (\zeta + \epsilon_{\mathcal{P}})^{1/q_{\mathcal{P}}}$

This switchings control issue is close to the denoising or segmentation problem in image processing. Consider indeed a black and white image characterized by each pixel's grey level ρ: black if $\rho = 1$ and white if $\rho = 0$. A clear image, called the target, must be recovered from a noisy image, called the data, by denoising processes: the variations of the pixels must be controlled to recover clear black and white zones. In this work, the target image term plays the role of the temperature constraints. The variation control remains the same and the techniques applied in image analysis are adapted to the power context.

Denoising methods

Fully based on [49], we introduce in this paragraph the denoising problem in the image analysis context. Let's recall the denoising problem. Consider a target and a data images, defined on a square D by their pixel value $g_{\text{target}}, g_{\text{data}} : D \to [0, 1]$ (in the continuous version). Based on the data image, the goal is the image guess g_{guess} the closest to the target. This problem is composed of two different objectives. First, the recovered image must remain close from the data image (least square term). Then, a penalization F is added to limit the pixels variations, leading to the following problem:

$$\text{find } g_{\text{guess}} = \underset{u \in L^2}{\arg\min}\, \lambda F(u) + \|u - g_{\text{data}}\|^2_{L^2([0,1]^2)}. \tag{8.2.7}$$

In image analysis, this penalization chosen is the function TV, called the total variation of the function and defined as

$$TV : u \mapsto \int_{[0,1]^2} |\nabla u| dx. \tag{8.2.8}$$

This seems to require the function u to belong to $W^{1,1}$ (or even more C^1) but it turns out not to be the case and u can be discontinuous (the correct space to consider is the bounded variation space, no theoretical details are given here). The total variation exactly corresponds to the number of jumps for a piecewise constant function.

Remark 8.3. Other penalizations could have been thought of. The most intuitive are

$$F_1(u) = \|u\|_{L^2([0,1]^2)} = \int_{[0,1]^2} |u|^2 dx, \qquad F_2(u) = \|u\|_{H^1([0,1]^2)} = \int_{[0,1]^2} |\nabla u|^2 dx,$$

respectively corresponding to a L^2 and H^1 norm. The first function F_1 does not reduce the noise. Indeed, it only requires the function u to belong to L^2 without any impact on its variations. On the other hand, the function F_2 requires the variable u to belong to H^1 which implies its continuity. As a consequence, such a function cannot represent sharp discontinuous variations leaving the recovered image still blurred. In the powder bed fusion context of this work, these penalizations correspond to the scalar product choices to determine the optimization problem gradient from the derivatives (this process has been detailed in Section 3.2.4 for a shape variable and can be extended to the power variable).

Application to the power optimization

In the path optimization context this work focuses on, the least square function from image analysis corresponds to the minimization of the final time subjected to temperature constraints. Inspired by

image analysis problems, the new optimization problem gathering the relaxation penalization (addition of the constraint $\mathcal{P}=0$, see (8.2.6)) and the jump control is now rewritten as

$$\min_{\Gamma\in\mathcal{G},\zeta\in L^2([0,t_{\rm F}],[0,1])} L_{\rm F} = Vt_{\rm F} + l_{\rm TV}\int_0^{t_{\rm F}} |\mathrm{d}_t\zeta| dt,$$
$$\text{such that } \begin{cases} C_\phi(\Gamma,\zeta) = C_{\rm M}(\Gamma,\zeta) = 0, \\ \mathcal{P}(\zeta) = 0, \\ y\in L^2\left([0,t_{\rm F}],H^1(D)\right) \text{ solution of (8.2.1)} \\ \text{that depends on } \Gamma \text{ and } \zeta. \end{cases} \tag{8.2.9}$$

8.2.3 Optimization algorithm

The new optimization problem must now be solved with an appropriate numerical algorithm. However, the total variation function is not differentiable and the gradient based approach previously developed must be replaced by a proximal gradient algorithm as studied in Section 2.6. The total variation function is indeed convex and the problem thus fits the requirements presented in Section 2.6.

The constraints related to indicator functions are added in problem (8.2.9) as follows

$$\min_{\Gamma,\zeta\in L^2([0,t_{\rm F}],\mathbb{R})} L_{\rm F} = Vt_{\rm F} + l_{\rm TV}\int_0^{t_{\rm F}} |\partial_t\zeta| dt + \chi_{\Gamma\subset D} + \chi_{\zeta\in[0,1]},$$
$$\text{such that } \begin{cases} C_\phi(\Gamma,\zeta) = C_{\rm M}(\Gamma,\zeta) = 0, \\ \mathcal{P}(\zeta) = 0, \\ y\in L^2\left([0,t_{\rm F}],H^1(D)\right) \text{ solution of (8.2.1)}. \end{cases} \tag{8.2.10}$$

where $\chi_{\Gamma\subset D}$ is the indicator function of $\Gamma\subset D$ and $\chi_{\zeta\in[0,1]}$ the indicator function that the variable ζ remains in $[0,1]$. To solve this problem, we combine the Augmented Lagrangian and proximal gradient methods (see Section 2.6.5) with μ_ϕ, $\mu_{\rm M}$, c_ζ penalization coefficients and

$$\begin{aligned}\mathcal{L}_{\rm ALM}(\Gamma,\zeta,l_\phi,l_{\rm M},\lambda_\zeta) \quad &= \underbrace{Vt_{\rm F} + l_\phi C_\phi + \frac{\mu_\phi}{2}C_\phi^2 + l_{\rm M}C_{\rm M} + \frac{\mu_{\rm M}}{2}C_{\rm M}^2 + \lambda_\zeta\mathcal{P} + \frac{c_\zeta}{2}\mathcal{P}^2}_{f_{\rm D}} \\ &\quad + \underbrace{l_{\rm TV}\frac{1}{t_{\rm F}}\int_0^{t_{\rm F}} |\partial_t\zeta| dt + \chi_{\Gamma\subset D} + \chi_{\zeta\in[0,1]}}_{f_{\rm ND}}.\end{aligned} \tag{8.2.11}$$

Recall that the constraints C_ϕ and $C_{\rm M}$ depends on the temperature y solution of (8.2.1). The function $f_{\rm D}$ correspond to the differentiable terms for which a gradient can be determined. The function $f_{\rm ND}$ is non differentiable but convex. The update of the Lagrange multipliers follows:

$$l_\phi^{k+1} = l_\phi^k + \mu_\phi C_\phi^k, \quad l_{\rm M}^{k+1} = l_{\rm M}^k + \mu_{\rm M}C_{\rm M}^k, \quad \lambda_\zeta^{k+1} = \lambda_\zeta^k + c_\zeta\mathcal{P}^k. \tag{8.2.12}$$

The proximal of the non differentiable function must be made explicit. Since the variable ζ is independent from the path Γ (see Remark 8.1, the function $f_{\rm ND}$ is the sum $f_{\rm ND}^\zeta$ depending on the variable ζ only and $f_{\rm ND}^\Gamma$ depending on the path Γ:

$$\begin{cases} f_{\rm ND}^\zeta = l_{\rm TV}\dfrac{1}{t_{\rm F}}\displaystyle\int_0^{t_{\rm F}} |\partial_t\zeta| dt + \chi_{\zeta\in[0,1]}, \\ f_{\rm ND}^\Gamma = \chi_{\Gamma\subset D}. \end{cases} \tag{8.2.13}$$

The proximal operators corresponding to each function are then independently applied to each variable. For the path, we have:

$$\mathrm{prox}_{\lambda f_{\rm ND}^\Gamma}(\tilde{\Gamma}) = \mathbb{P}_D\left(\tilde{\Gamma}\right), \tag{8.2.14}$$

where $\mathbb{P}_D$ is the orthogonal projection on the domain D. Because the variable ζ is a one dimensional variable and since $f_{\rm ND}^\zeta$ is the sum of a convex function and a characteristic function, the corresponding proximal is the composition of the proximal to each function (see [207]):

$$\mathrm{prox}_{\lambda f_{\rm ND}^\zeta}(\tilde{\zeta}) = \mathbb{P}_{\zeta\in[0,1]}\left(\mathrm{prox}_{\lambda TV}(\tilde{\zeta})\right). \tag{8.2.15}$$

The proximal of the total variation function is computed using toolbox "proxTV" of Python [24, 25]. The proximal steps are chosen equal to the gradient steps. The process follows a line search and is further detailed in the application to the steady state context. Finally, Algorithm 8.1 is applied.

Initialize (Γ^0, ζ^0) and l_ϕ^0, l_{M}^0, λ_ζ^0
Compute the objective function
Compute the gradient with respect to the path $\nabla_\Gamma \mathcal{L}_{\mathrm{ALM}}(\Gamma^0, \zeta^0, l_\phi^0, l_{\mathrm{M}}^0, \lambda_\zeta^0)$
Compute the gradient with respect to the power variable $\nabla_\zeta \mathcal{L}_{\mathrm{ALM}}(\Gamma^0, \zeta^0, l_\phi^0, l_{\mathrm{M}}^0, \lambda_\zeta^0)$
while *the stopping criterion is not reached* **do**
    Compute the update steps s_Γ^n and s_ζ^n
    Compute $\tilde{\Gamma}^{n+1} = \Gamma^n - s_\Gamma^n \nabla_\Gamma f_{\mathrm{D}}(\Gamma^n, \zeta^n, l_\phi^n, l_{\mathrm{M}}^n, \lambda_\zeta^n)$
    Compute $\tilde{\zeta}^{n+1} = \zeta^n - s_\zeta^n \nabla_\zeta f_{\mathrm{D}}(\Gamma^0, \zeta^n, l_\phi^n, l_{\mathrm{M}}^n, \lambda_\zeta^n)$
    Path projection $\Gamma^{n+1} = \mathbb{P}_D(\Gamma^n)$
    Application of the proximal related to ζ: $\zeta^{n+1} = \mathbb{P}_{\zeta\in[0,1]}\left(\mathrm{prox}_{s_\zeta^n l_{\mathrm{TV}} TV(\tilde{\zeta}^n)}\right)$
    Compute the new objective function
    if *the new objective function is smaller than before (up to a tolerance)* **then**
        Iteration accepted
        Update of the Lagrange multipliers and recompute the objective function
        Compute the shape and power derivatives
        Increase the step coefficients (line search)
    end
    else
        Reject the iteration
        Decrease the coefficients: line only for a line iteration, power only for a power iteration and both if both
    end
end

Algorithm 8.1: Power and shape optimization algorithm

Remark 8.4. Note that if the temperature constraints are split into a phase constraint and a maximum temperature constraints, they could also be split into three different ones (phase, maximum temperature in the object and maximum temperature out of the object) or aggregated into an only one (like in the steady case).

8.3 Application of the power control in the steady state context

The numerical computations are conducted in the steady case. Indeed, the faster computations and the availability of shape differentiation simplify the optimization. We then can use the algorithm analysis it provides to spare tests in the transient case. In this Section, after specifying the optimization problem in the steady state context, the different gradients and proximal operators are computed. Then, the optimization algorithm is made precise by detailing the update step choices. Finally numerical results are provided: the impact of the power variable is first tested by fixing the path and running the relaxed problem, the relaxation-penalization method is then analyzed to elect one penalization function and the jump control strategy is finally applied.

In this steady state context, the Augmented Lagrangian function (8.2.11) becomes, with $L \geq L_{\mathrm{F}}$ fixed (see Remarque 8.1):

$$\mathcal{L}_{\mathrm{ALM}} = \underbrace{L_{\mathrm{F}} + C + \frac{c}{2}C^2 + \lambda_\zeta \mathcal{P} + \frac{c_\zeta}{2}\mathcal{P}^2}_{f_{\mathrm{D}}} + \underbrace{l_{\mathrm{TV}} \int_\Gamma |\mathrm{d}_s \zeta| ds + \chi_{\Gamma\subset D} + \chi_{\zeta\in[0,1]}}_{f_{\mathrm{ND}}}, \tag{8.3.1}$$

with $C = C_\phi + C_{\mathrm{M}}$, $\mathcal{P} = \dfrac{1}{L_{\mathrm{F}}} \displaystyle\int_\Gamma f_{\mathrm{pen}}(\zeta(s))\, ds$ and the temperature y stepping in the constraint C is given as $y \in H^1(D)$ solution of, $\forall \phi \in H^1(D)$,

$$\int_D \lambda \nabla y \cdot \nabla \phi + \beta (y - y_{\mathrm{ini}})\, \phi dx - \int_\Gamma P\zeta\phi ds = 0. \tag{8.3.2}$$

Remark 8.5. Note that in the steady case, the total variation function depends on both the path Γ and the power variable ζ. This complicates the proximal computation further detailed in the following.

8.3.1 Path discretization

The path discretization must now carry the variable $\zeta(s)$ and the broken line approach detailed in Section 5.3 must be adapted.

The broken line approach is kept and the variable ζ is defined on the segments, leading to the numerical variable $(\zeta_i)_{i\in[\![1,N_u-1]\!]}$. If the computation of the normal, tangent and curvature (Section 5.3.3) and the differentiation and integration of $P0-$functions can be applied to the variable ζ (Section 5.3.5), the discretization coherence process and the source computation must be modified.

Discretization coherence

If is was previously possible to delete and add segments, it is not anymore. Imagine for example that a segment S_i must be deleted. However, the value ζ_i is high whereas the values on the neighboring segments are not. A simple delete would thus remove most of the energy of this part of the path. To remedy this problem, all the addition and deletion must keep $\int_\Gamma \zeta$ constant.

Let's first consider the adding process. Let S be a segment carrying the value ζ_S, with length l such that $l > d_\mathrm{u}$. This segment is split into $\left[\frac{l}{d_\mathrm{u}}\right]+1$ elements of size $\tilde{l} = \frac{l}{\left[\frac{l}{d_\mathrm{u}}\right]+1}$, each of them carrying $\tilde{\zeta} = \zeta_S$. Thus,

$$l\zeta = \sum_{i=1}^{\left[\frac{l}{d_\mathrm{u}}\right]+1} \tilde{l}\tilde{\zeta}. \tag{8.3.3}$$

The deleting process follows the same structure than in Section 5.3.2, yet not simply removing the points. Assume that $\|u_i - u_{i-1}\| \leq d_\mathrm{l}$ and carries ζ_{i-1}. In the backward process previously defined, the new segment to consider was $u_i - u_{i-2}$ and point u_{i-1} was removed. In this new process, the function ζ must be defined on the resulting segment (introducing the corresponding variable $\tilde{\zeta}$). To do so, a quantity E is defined in relation to the three points (u_i, u_{i-1}, u_{i-2}) as well as a second one $\tilde{E}$ related to the two points (u_i, u_{i-2}):

$$E = \|u_{i-1} - u_{i-2}\|\zeta_{i-1} + \|u_i - u_{i-1}\|\zeta_i, \qquad \tilde{E} = \|u_i - u_{i-2}\|\tilde{\zeta}.$$

To keep the $\tilde{E} = E$, one should choose

$$\tilde{\zeta} = \frac{\|u_{i-1} - u_{i-2}\|\zeta_{i-1} + \|u_i - u_{i-1}\|\zeta_i}{\|u_i - u_{i-2}\|}.$$

However, ζ is bounded and must remain with $[0, 1]$. If the new function value $\tilde{\zeta} \leq 1$, then the process can go on (Figure 8.3 (a-b)). Else, the point u_{i-1} can not be removed and is simply moved (Figure 8.3 (c)). This point must be pushed away from u_i and should not be brought closer to u_{i-2} (in the backward process), leading to an optimization problem:

$$\min_{u\in\mathbb{R}^2} \|u_{i-1} - u\| \qquad \text{such that} \begin{cases} \|u - u_i\| - d_\mathrm{l} \geq 0 \\ \|u - u_{i-2}\| - \|u_{i-1}u_{i-2}\| \geq 0 \end{cases}$$

If the point u found is in the working domain D, u_{i-1} is replaced by u and the function ζ is adjusted to maintain the same quantity E (Figure 8.3 (d-e)). Else, a set of points v is defined as

$$\mathcal{V} = \{v \in D; \|v - u_i\| = d_\mathrm{l} \text{ and } \|v - u_{i-2}\| = \|u_{i-1}u_{i-2}\|\} \cup \{v \in D; \|v - u_i\| = d_\mathrm{l} \text{ and } v \in \partial D\}.$$

and u is taken as $\min_{G\in\mathcal{G}} \|v - u_{i-1}\|$, with adaptation of the function ζ (Figure 8.4).

Source computation

The process creating the $P0-$source from the path is also slightly modified from the one presented in Section 5.3.4. Indeed, when a segment S carrying the value ζ_S crosses a physical mesh cell, the value attributed to the mesh is not the classic length but the length multiplied by ζ.

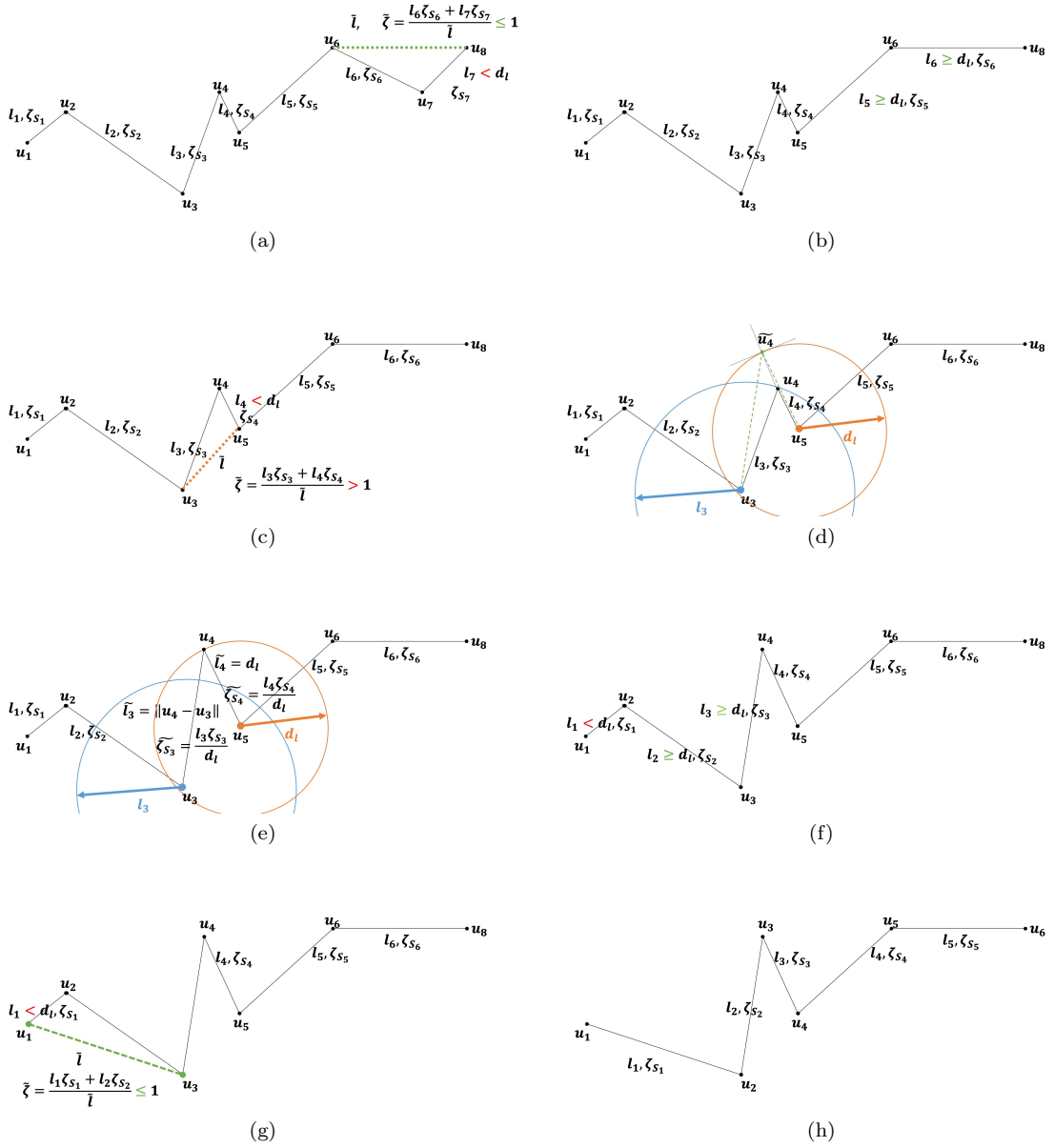


Figure 8.3: Deleting process: (a)Starting from the end point, the last segment is too small but removing it does not make ζ overcome 1 and the segment can be deleted. (b)The segment resulting from the delete is large enough as well as the one starting form u_5. (c)The length of element 4 is too small. Yet, the segment cannot be deleted because the resulting function ζ would overcome 1. (d-e)The closest point to u_4 is found so that $\|\tilde{u_4} - u_5\| = d_l$ and $\|\tilde{u_4} - u_3\| \geq \|u_4 - u_3\|$. The corresponding length and function value are updated. (f)Each segment but the first is large enough. The first point cannot be removed. (g)Process started from u_1. (h)Resulting path.

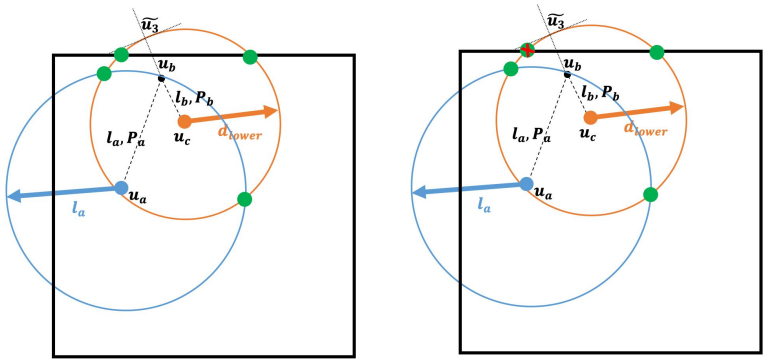
8.3.2 Derivative computations

The differentiation with respect to the multipliers gives, at iteration k, the following update steps:

$$\begin{cases} l_C^{k+1} = l_C^k + cC^k \\ \lambda_\zeta^{k+1} = \lambda_\zeta^k + c_\zeta \mathcal{P}^k. \end{cases} \tag{8.3.4}$$

Proposition 8.1 gives the derivatives of the derivative of f_{D} with respect to both variables.

Proposition 8.1. *Let $\Gamma \in \mathcal{G}$ and $\zeta \in L^2([0, L], \mathbb{R})$. Then, the function f_{D} is differentiable at (Γ, ζ). With respect to the path, $\forall \theta \in \mathcal{C}^2\left(\overline{D}, \mathbb{R}^2\right)$,*



(a) The closest point to u_b out of the circles is out of the box

(b) Intersection between the circle of center u_b with the other circle and the box

Figure 8.4: Deleting process with $\tilde{\zeta} > 1$, choosing the new point: the point u_b is not in the working domain D. The intersection between both circles and the circle centered in u_c with the boundary ∂D are computed. The closest to u_b belonging to D and out of the circle centered in u_a is chosen.

$$
\begin{aligned}
D_\Gamma f_{\mathrm{D}}(\Gamma,\zeta)(\theta) = &\int_\Gamma \kappa\theta\cdot n ds + (\theta\cdot\tau)(B) - (\theta\cdot\tau)(A) \\
&+\frac{\lambda_\zeta + c_\zeta \mathcal{P}}{L_{\mathrm{F}}}\left[\left(\int_\Gamma (f(\zeta)-\mathcal{P})\,\kappa\theta\cdot n - f'(\zeta)\partial_\tau\zeta\theta\cdot\tau ds\right) + ((f(\zeta)-\mathcal{P})\,\theta\cdot\tau)(B) - ((f(\zeta)-\mathcal{P})\,\theta\cdot\tau)(A)\right] \\
&-\int_\Gamma \left[P\zeta\,(p\kappa+\partial_n p)\,\theta\cdot n - Pp\partial_\tau\zeta\theta\cdot\tau\right] ds - (P\zeta p\theta\cdot\tau)(B) + (P\zeta p\theta\cdot\tau)(A)
\end{aligned}
\tag{8.3.5}
$$

and with respect to the power variable ζ, $\forall\delta\zeta \in L^2([0,L],\mathbb{R})$,

$$
D_\zeta f_{\mathrm{D}}(\Gamma,\zeta)(d\zeta) = \int_\Gamma \left(-Pp + \frac{\lambda_\zeta + c_\zeta\mathcal{P}}{L_{\mathrm{F}}} f'_{\mathrm{pen}}(\zeta)\right)\delta\zeta ds. \tag{8.3.6}
$$

The function $p \in H^1(D,\mathbb{R})$ *is solution of* (8.3.7) *(actually corresponding to the adjoint in Proposition 6.1 in Chapter 6):*

$$
\begin{cases}
-\nabla(\lambda p) + \beta p = (l_C + cC)\left(2l_\phi\,(y_\phi - y)^+ - 2l_{\mathrm{M}}\,(y - y_{\mathrm{M}})^+\right) & in\, D \\
\lambda\partial_n p = 0 & on\, \partial D.
\end{cases}
\tag{8.3.7}
$$

PROOF.
The proof of this proposition consists in applying the method of Céa to the function f_{D}. Note that this derivative includes the transport of the power variable. Indeed, since the power variable ζ is only defined on the path Γ, we must apply the formula related to J_2 and J_3 in Proposition 3.4 (see Remark 3.4). □

For numerical applications, these derivatives must be transformed into discrete gradients. Following Section 3.2.4 (Chapter 3) and Section 6.3.2 (Chapter 6), the following subsections give the gradients with respect to the path and the power variable. In the discretization process the discretization of the functions ζ and ζ matches. Indeed, they are given one value per path discretization element ($P0-$function along the path, see Chapter 5, Section 5.3.5), that can be related to the corresponding element or to the length crossed from the beginning of the path to this element. Since the path is discretized with N_u nodes $(u_i)_{i\in[\![1,N_u]\!]}$, the power variable vector is constituted of $N_u - 1$ elements: $(\zeta)_{i\in[\![1,N_u-1]\!]} = (\zeta)_{i\in[\![1,N_u-1]\!]}$.

Discrete gradient related to the path

Following (3.2.19), the gradient with respect to the path is given by $\nabla_\Gamma f_{\mathrm{D}} = \partial_\tau^\Gamma f_{\mathrm{D}} \tau + \partial_n^\Gamma f_{\mathrm{D}} n H^1(\Gamma, \mathbb{R}^2)$, with

$$\begin{cases} \partial_\tau^\Gamma f_{\mathrm{D}}(\Gamma) = \underset{W \in H^1(\Gamma,\mathbb{R})}{\arg\min} \overbrace{\int_\Gamma \left(\frac{\nu_\Gamma^2}{2} (\partial_\tau W)^2 + \frac{1}{2} W^2 \right) ds}^{LHS} - \overbrace{\int_\Gamma v_\tau W ds - v(B)W(B) + v(A)W(A)}^{RHS^\tau}, \\ \partial_n^\Gamma f_{\mathrm{D}}(\Gamma) = \underset{W \in H^1(\Gamma,\mathbb{R})}{\arg\min} \underbrace{\int_\Gamma \left(\frac{\nu_\Gamma^2}{2} (\partial_\tau W)^2 + \frac{1}{2} W^2 \right) ds}_{} - \underbrace{\int_\Gamma v_n W ds}_{RHS^n}, \end{cases} \tag{8.3.8}$$

and

$$\begin{cases} v_\tau(s) = Pp\partial_\tau\zeta - \dfrac{\lambda_\zeta + c_\zeta \mathcal{P}}{L_{\mathrm{F}}} f'(\zeta)\partial_\tau\zeta \\ v_\tau(B) = 1 - P\zeta(B)p(B) + \dfrac{\lambda_\zeta + c_\zeta \mathcal{P}}{L_{\mathrm{F}}} \left(f(\zeta(B)) - \mathcal{P} \right) \\ v_\tau(A) = -1 + P\zeta(A)p(A) - \dfrac{\lambda_\zeta + c_\zeta \mathcal{P}}{L_{\mathrm{F}}} \left(f(\zeta(A)) - \mathcal{P} \right) \\ v_n(s) = \kappa - P\zeta \left(p\kappa + \partial_n p \right) + \dfrac{\lambda_\zeta + c_\zeta \mathcal{P}}{L_{\mathrm{F}}} \kappa \left(f(\zeta) - \mathcal{P} \right) \end{cases} \tag{8.3.9}$$

The discretization of the left hand side of each equation is given by (6.3.6) (Chapter 6). As for the right hand side it holds:

$$\begin{aligned} RHS^\tau &= \int_\Gamma v_\tau W_\tau ds + v_\tau(B)W_\tau(B) - v_\tau(A)W_\tau(A) \\ &\approx -\left[1 - P\zeta_1 p_1 + \frac{\lambda_\zeta + c_\zeta \mathcal{P}}{L_{\mathrm{F}}} \left(f(\zeta_1 - \mathcal{P}) \right) \right] W_{\tau,1} \\ &\quad + \sum_{i=1}^{N_u - 1} \left[Pl_i \frac{p_i \partial_\tau \zeta_i W_{\tau,i} + p_{i+1} \partial_\tau \zeta_{i+1} W_{\tau,i+1}}{2} - \frac{\lambda_\zeta + c_\zeta \mathcal{P}}{L_{\mathrm{F}}} l_i f'(\zeta_i) \frac{\partial_\tau \zeta_i W_{\tau,i} + \partial_\tau \zeta_{i+1} W_{\tau,i+1}}{2} \right] \\ &\quad + \left[1 - P\zeta_{N_u - 1} p_{N_u} + \frac{\lambda_\zeta + c_\zeta \mathcal{P}}{L_{\mathrm{F}}} \left(f(\zeta_{N_u - 1}) - \mathcal{P} \right) \right] W_{\tau,N_u} \end{aligned} \tag{8.3.10}$$

$$\begin{aligned} RHS^n &= \int_\Gamma v_n W_n ds \\ &\approx \sum_{i=1}^{N_u - 1} \left[l_i \frac{\kappa_i W_{n,i} + \kappa_{i+1} W_{n,i+1}}{2} - l_i \zeta_i P \frac{\left(p_i \kappa_i + \partial_n p_i \right) W_{n,i} + \left(p_{i+1} \kappa_{i+1} + \partial_n p_{i+1} W_{n,i+1} \right)}{2} \right. \\ &\qquad \left. + \frac{\lambda_\zeta + c_\zeta \mathcal{P}}{L_{\mathrm{F}}} l_i \left(f(\zeta_i) - \tilde{\mathcal{P}} \right) \frac{\kappa_i W_{n,i} + \kappa_{i+1} W_{n,i+1}}{2} \right] \end{aligned} \tag{8.3.11}$$

Discrete gradient related to the power variable

As stated in Section 8.2, the power variable is expected to be only in L^2. Nevertheless, it is interesting to compare the H^1 regularization effect to the total variation penalization and both scalar products are given (see Remark 8.3).

First the classic L^2 scalar product requires $\zeta \in L^2([0, L])$ and $\nabla_\zeta f_{\mathrm{D}} \in L^2([0, L])$ to satisfy,

$$\int_\Gamma \nabla_\zeta f_{\mathrm{D}} W ds = D_\zeta \mathcal{L}_{\mathrm{ALM}}(W) = \int_\Gamma v_\zeta W ds. \tag{8.3.12}$$

This leads to $\forall i \in [\![1, N_u - 1]\!]$

$$\left(\nabla_\zeta f_{\mathrm{D}} \right)_i = -P \frac{p_i + p_{i+1}}{2} + \frac{\lambda_\zeta + c_\zeta \mathcal{P}}{L_{\mathrm{F}}} f'(\zeta_i). \tag{8.3.13}$$

In case of a H^1 scalar product, $\zeta \in H^1([0,L])$ and $\nabla_\zeta f_{\mathrm{D}} \in H^1([0,L])$ must satisfy $\forall W \in H^1([0,L])$,

$$\int_\Gamma \nu_\zeta^2 \partial_s \left(\nabla_\zeta f_{\mathrm{D}}\right) \cdot \partial_s W + \left(\nabla_\zeta f_{\mathrm{D}}\right) W ds = D_\zeta \mathcal{L}_{\mathrm{ALM}}(W) = \int_\Gamma v_\zeta W ds. \tag{8.3.14}$$

This leads to minimizing the following discrete function:

$$\begin{aligned}\sum_{i=1}^{N_u-1} \Bigg[\frac{l_i}{2}\frac{\nu_\zeta^2}{2} \left(\left(\frac{l_i \left(\nabla_\zeta f_{\mathrm{D}}\right)_i - l_{i-1}\left(\nabla_\zeta f_{\mathrm{D}}\right)_{i-1}}{l_i + l_{i-1}} \right)^2 + \left(\frac{l_{i+1}\left(\nabla_\zeta f_{\mathrm{D}}\right)_{i+1} - l_i \left(\nabla_\zeta f_{\mathrm{D}}\right)_i}{l_{i+1}+l_i} \right)^2 \right) + \frac{l_i}{2}\left(\nabla_\zeta f_{\mathrm{D}}\right)_i^2 \\ + \left(-P\frac{p_i + p_{i+1}}{2} + \frac{\lambda_\zeta + c_\zeta \mathcal{P}}{L_{\mathrm{F}}} f'(\zeta_i) \right) \left(\nabla_\zeta f_{\mathrm{D}}\right)_i \Bigg]\end{aligned} \tag{8.3.15}$$

Proximal computation

The proximal operator of the non differentiable function involves both the path Γ and the power variable ζ. To simplify the problem we approximate this proximal considering that the total variation function does not depend on the path and that:

$$\mathrm{prox}_{\lambda f_{\mathrm{ND}}} = \mathrm{prox}_{\lambda \mathbb{1}_{\zeta \in [0,1]}} \circ \mathrm{prox}_{\lambda TV} \circ \mathrm{prox}_{\lambda \mathbb{1}_{\Gamma \subset D}}. \tag{8.3.16}$$

Since the proximal operator of a characteristic function is the orthogonal projection, this leads to

- first projecting the path Γ in the working domain D. This involves an approximation: whatever modification on the path discretization segments length this modification induces, the power variable on this segment remains unmodified).
- then applying the proximal operator related to the TV function. The discrete formulation of this function is

$$TV = \sum_{i=1}^{N_u-2} |\zeta_{i+1} - \zeta_i|, \tag{8.3.17}$$

  proximal operator is computed using the toolbox "proxTV" in Python (see [24, 25]. The path is not taken into account in this proximal operator. The step chosen for the proximal step is the same than for the gradient step related to the variable ζ.
- finally project the resulting power variable ζ in the domain $[0,1]$.

8.3.3 Optimization algorithm

Algorithm 8.1 can finally be applied, providing that the steps s_Γ and s_ζ are given. The step with respect to the shape follows the choices of Chapter 6:

$$s_\Gamma^n = \frac{C_{s,n}\Delta x}{\max_i \| \left(\nabla_\Gamma f_{\mathrm{D}}\right)_i \|}, \tag{8.3.18}$$

where $C_{s,0}$ fixed to 1, updated by $\max(C_{s,0}, \eta_{\mathrm{acc}}^s C_{s,n})$ when the iteration is accepted ($\eta_{\mathrm{acc}}^s = 1.2$) and by $\eta_{\mathrm{ref}}^s C_{s,n}$ when refused ($\eta_{\mathrm{ref}}^s = 0.6$).

The same technique is used for the power, with

$$s_\zeta^n = \frac{C_{s_\zeta,n}}{\max_i | \left(\nabla_\zeta f_{\mathrm{D}}\right)_i |}, \tag{8.3.19}$$

and $C_{s_\zeta,0} = 0.1$ and the same coefficient updates.

To get the numerical results, the normalization technique presented in Chapter 6 (Section 6.4.1) has been applied here and each of the optimization function and constraints have been normalized by their initial value. The total variation at the initial state often being equal to 0, the normalization is realized with the first iteration different from 0.

8.3.4 Numerical results: optimization of the power

In the first tests, we focus on the power variable only. The objective is to understand its ability to improve the constraints fulfillment. The path being fixed, the relaxed optimization problem is run with the same physical and discretization settings than for the numerical tests in Chapter 6.

Four different test cases are studied:

- in the aluminium case, the path consists in a zigzag with 9 lines and the power is initialized to $\zeta^0 = 0.5$ all along the path (Figure 8.5 (a)), referred to as "zigzag9-ALU",
- in the aluminium case, the path consists in the result from path optimization starting from a zigzag with 9 lines and the power is initialized to $\zeta^0 = 1$ all along the path (Figure 8.5 (b)), referred to as "zigzag9BIS-ALU"
- in the titanium case, the path consists in a zigzag with 12 lines and the power is initialized to $\zeta^0 = 0.5$ all along the path (Figure 8.5 (c)), referred to as "zigzag12-TI",
- in the titanium case, the path consists in the result from path optimization starting from a zigzag with 12 lines and the power is initialized to $\zeta^0 = 1$ all along the path (Figure 8.5 (d)), referred to as "zigzag12BIS-TI".

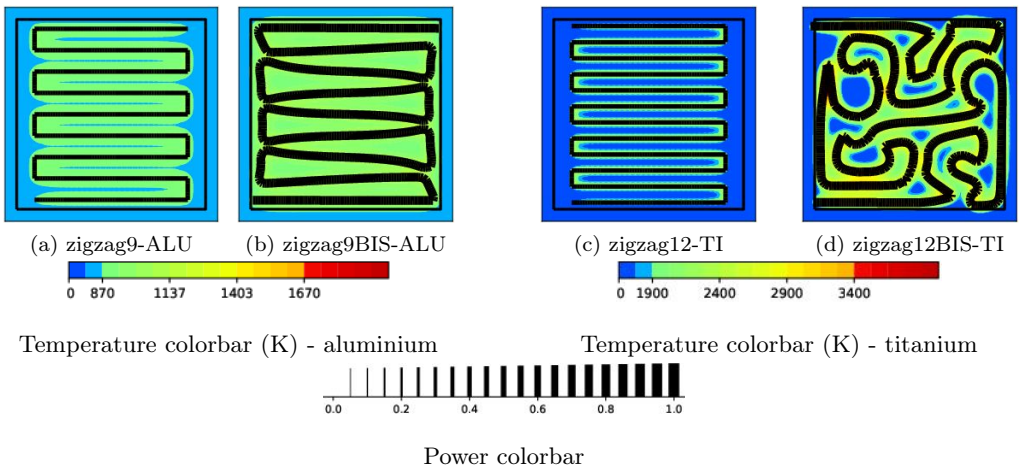
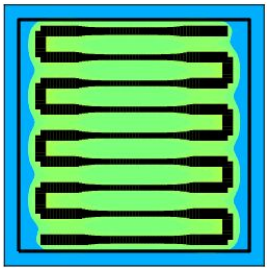


Figure 8.5: Temperature map for each initialization

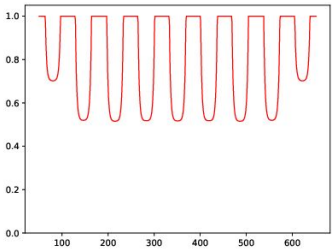
For each of these four tests, the results are presented in a Figure showing the resulting path, the power variable along the path at the final iteration as well as the constraint evolution (Figure 8.6, Figure 8.7, 8.8, 8.9). The numerical values are stored in two different tables: Table 8.1 for the aluminium and Table 8.2 for the titanium.

Case	$\overline{C}_\phi$	$\overline{C}_{\mathrm{M,D_S}}$	$\overline{C}_{\mathrm{M,D\backslash D_S}}$
zigzag9-ALU - initialization	$7.84e-4$	0.0	0.0
zigzag9-ALU - result	$3.32e-4$	0.0	$3.43e-9$
zigzag9BIS-ALU - initialization	$4.19e-7$	0.0	$2.83e-5$
zigzag9BIS-ALU - result	$4.93e-7$	0.0	$6.42e-8$

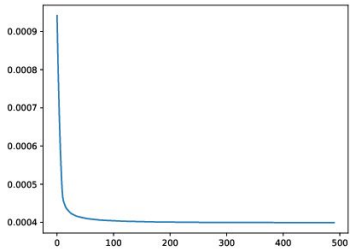
Table 8.1: Comparison of the cost and constraints for the power optimization (aluminium)



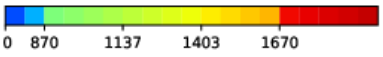
(a) Result (path and temperature)



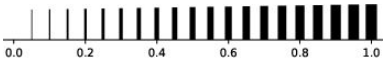
(b) Power variable ζ along the path



(c) Evolution of the constraint C with respect to the iterations

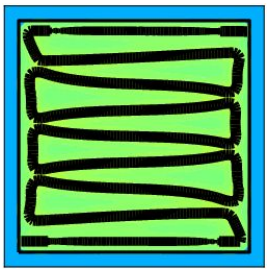


Temperature colorbar (K)

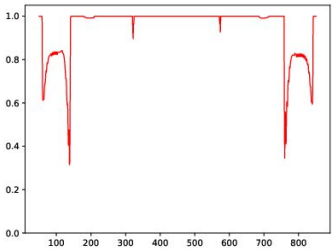


Power colorbar

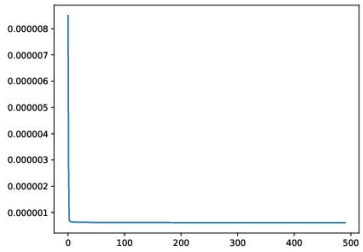
Figure 8.6: Power optimization starting from zigzag9-ALU (aluminium)



(a) Result (path and temperature)



(b) Power variabl ζ along the path

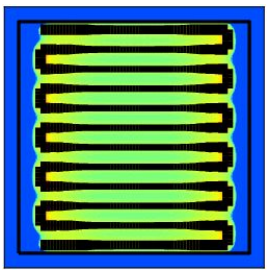


(c) Evolution of the constraint C with respect to the iterations

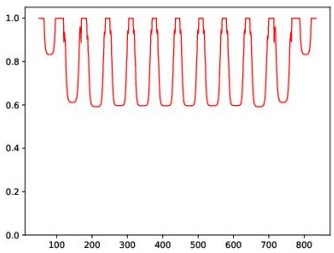
Temperature colorbar (K)

Power colorbar

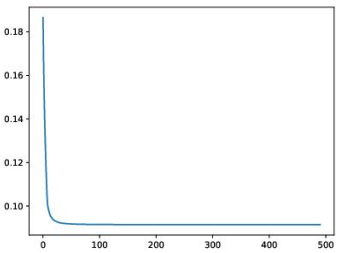
Figure 8.7: Power optimization starting from zigzag9BIS-ALU (aluminium)



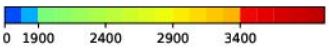
(a) Result (path and temperature)



(b) Power variabl ζ along the path



(c) Evolution of the constraint C with respect to the iterations



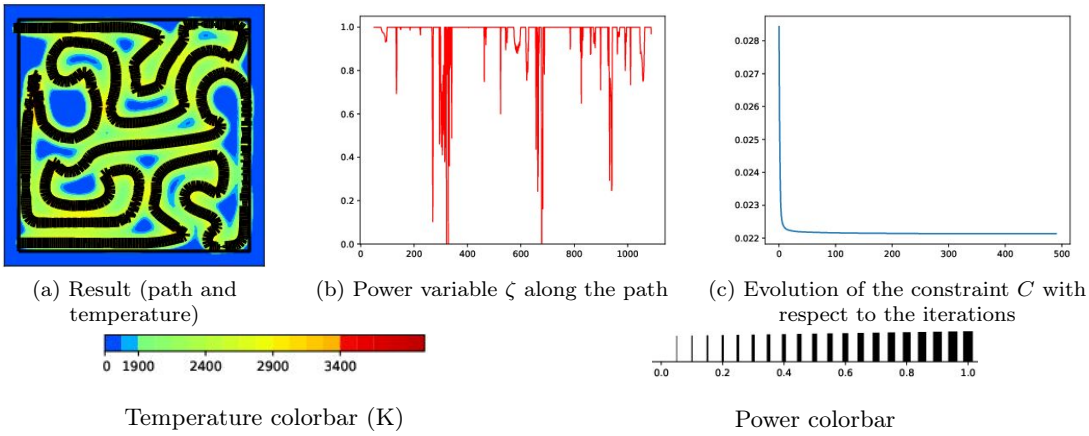
Temperature colorbar (K)

Power colorbar

Figure 8.8: Power optimization starting from zigzag12-TI (titanium)

Case	$\overline{C}_{\phi}$	$\overline{C}_{\mathrm{M,D_S}}$	$\overline{C}_{\mathrm{M,D\setminus D_S}}$
zigzag12-TI - initialization	$3.25e-2$	0.0	0.0
zigzag12-TI - result	$1.58e-2$	$6.27e-5$	0.0
zigzag12BIS-TI - initialization	$3.26e-3$	$3.37e-4$	$2.95e-3$
zigzag12BIS-TI - result	$3.66e-3$	$2.41e-5$	$5.46e-4$

Table 8.2: Comparison of the cost and constraints for the power optimization (titanium)



(a) Result (path and temperature)

(b) Power variable ζ along the path

(c) Evolution of the constraint C with respect to the iterations

Temperature colorbar (K)

Power colorbar

Figure 8.9: Power optimization starting from zigzag12BIS-TI (titanium)

We first consider the initialization with the non optimized zigzags (Figures 8.6 and 8.8). In both the aluminium and titanium cases, the power is increased in the corners of the zigzag, leading to a reduction of the phase constraint and an increase, in the titanium case, of the maximum temperature constraint. Note first that this result might be dependent on the optimization algorithm chosen. Then, in both results, the power is increased in the corners and decreased in the middle of the straight lines. Indeed, to melt the left and right sides of the square, the mower must be increased in the corners. However to remain under the maximum temperature, the power is reduced where the phase constraint is not critical. The results on the optimized zigzags (Figures 8.7 and 8.9) show that coupling the power and the path might improve the results. Indeed, the path optimization had converged to a result that did not fully satisfied the constraints. Running this power optimization afterwards improve the previous results by decreasing the maximum temperature constraints.

8.3.5 Numerical results of the relaxation penalization problem with jump control

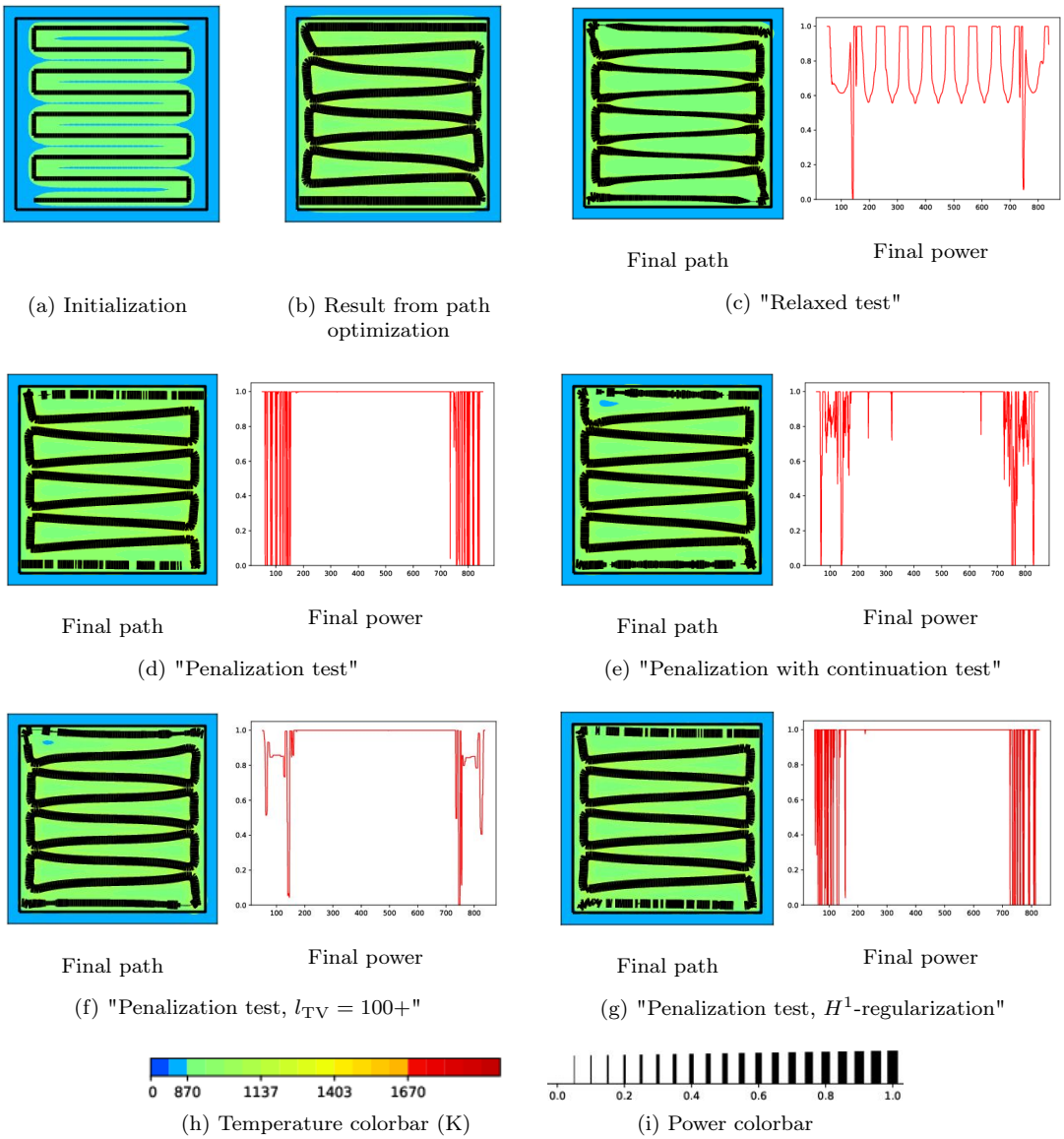
The coupled optimization is now considered, aimed at optimizing the path and the power to allow for path splittings. These first results aim at choosing an efficient penalization as well as comparing these results to the Chapter 6. From each initialization (Figure 8.5), the optimization is:

- without any penalization function. This test is called the "relaxed test".
- with the penalization function related to f_{pen}, where $q_{\mathcal{P}} = 1$ fixed. This test is called the "penalization test".
- with the penalization function related to f_{pen}, where $q_{\mathcal{P}}$ initialized to 1 and multiplied by 1.1 every 40 iterations. This test is called the "penalization with continuation test".

In these three first tests, the coefficient l_{TV} remains 0 during the whole optimization. To add the control of the number of jumps, two additional tests are run:

- "penalization test" in which the coefficient l_{TV} is initialized to 100 and multiplied by 1.25 every 40 iteration. This test is called "penalization test, $l_{\text{TV}} = 100+$".
- "penalization test" in which the coefficient l_{TV} is set to 0 but a H^1 regularization of the power derivative is introduced and $\nu_\zeta = 100$ (in the other tests, a simple L^2 gradient is chosen). This test is called "penalization test, H^1-regularization".

For each of the initializations proposed in Figure 8.5, each of these tests is run. Note that for all of them, the path is computed with a regularization coefficient $\nu_\Gamma = 15$ (see (8.3.8)). These results are to be compared to the ones found in Chapter 6 in which the path only was optimized. For each of the initialization cases, a Figure presenting all the results is given as well as a Table gathering the numerical values. Also note that the rediscretization of the path is different in this section than it was in Chapter 6. Indeed, the coherence of the power must be kept. Note also that, in all the length computations, the path at which the source is off is counted in the length. Controlling it another way is part of the



(a) Initialization (b) Result from path optimization (c) "Relaxed test"

(d) "Penalization test" (e) "Penalization with continuation test"

(f) "Penalization test, $l_{\mathrm{TV}} = 100+$" (g) "Penalization test, H^1-regularization"

(h) Temperature colorbar (K) (i) Power colorbar

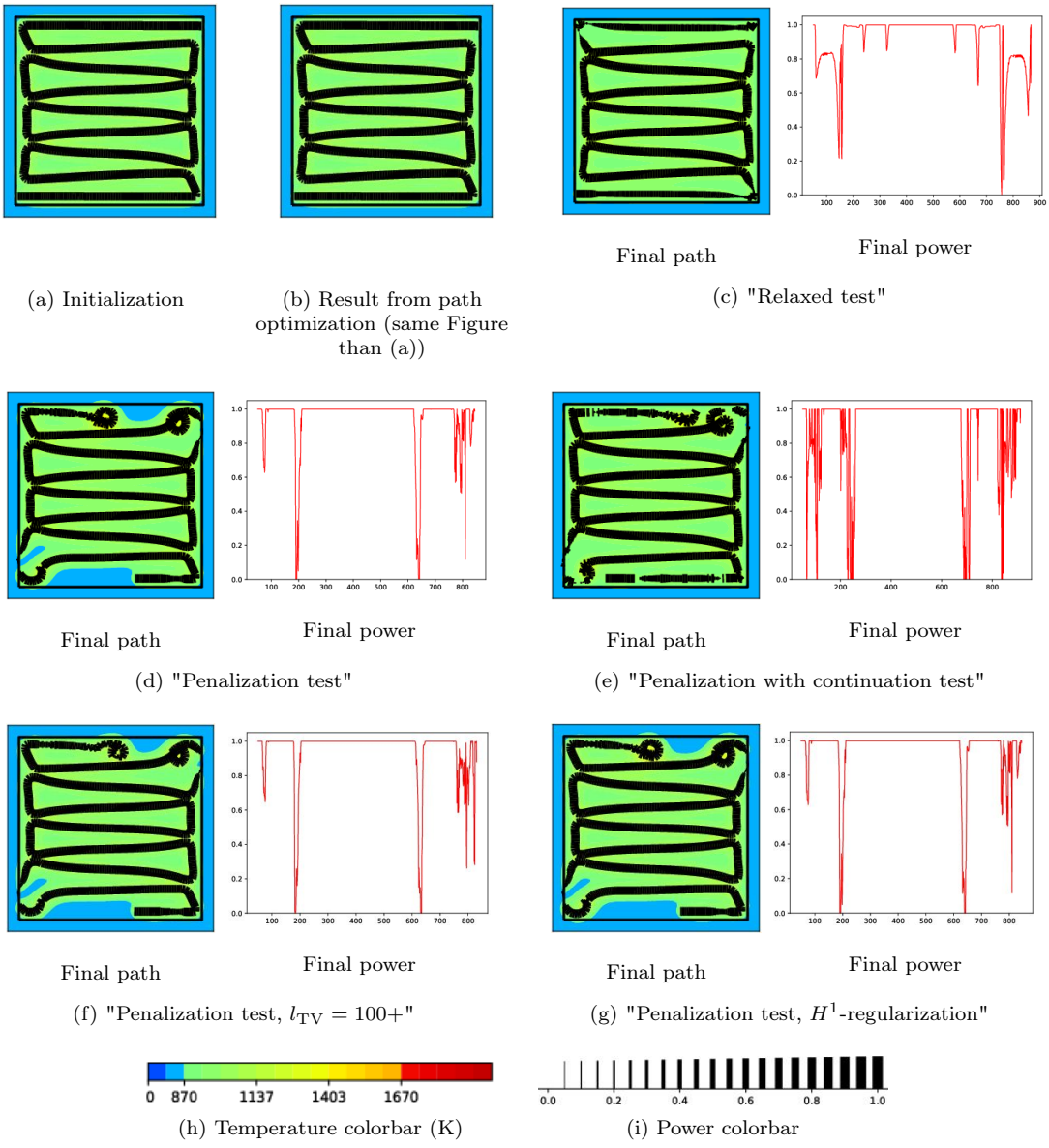
Figure 8.10: Coupled power and path optimization starting from zigzag9-ALU (aluminium)

perspectives (in the transient case especially, the velocity of the source is higher when the laser is off, which must be taken into account in the final time computation).

Case	L_{F} (m)	$\overline{C}_{\phi}$	$\overline{C}_{\mathrm{M,D_S}}$	$\overline{C}_{\mathrm{M,D\setminus D_S}}$	TV
initialization ($\zeta = 0.5$)	1.019e−2	7.84e−4	0.00	0.00	0
path optimization	1.159e−2	4.20e−7	0.00	2.84e−5	0
relaxed test	1.163e−2	7.65e−7	0.00	7.65e−7	12.9
penalization test	1.172e−2	1.58e−7	0.00	1.86e−7	57.3
penalization with continuation test	1.171e−2	2.55e−7	0.00	9.24e−7	28.0
penalization test, $l_{\mathrm{TV}} = 100+$	1.173e−2	7.80e−8	0.00	5.23e−7	10.1
penalization test, H^1-reg	1.169e−2	1.78e−7	0.00	1.61e−7	57.0

Table 8.3: Comparison of the cost and constraints of the coupled optimization for the final path starting from zigzag9-ALU (aluminium)

First of all, a comparison between the path optimization result and the relaxed test shows that adding the power as an optimization variable does not necessarily decrease the final path length (Figure 8.10 (b) and (c). However, the constraints are better satisfied: allowing the power to be smaller than its maximum value leads to an easier decrease of the maximum temperature constraints but, on the other hand, the length must be increased to fulfill the phase constraint. In the real process, the important parameter is actually the volumic energy. It would be thus interesting to run these tests for this quantity in the transient test case. Then, comparing the penalization tests with the relaxed test, it seems that the penalization chosen is efficient, yet better in the non continuation formulation. If latter confirmed, this could lead to definitely choosing the classical penalization. Note that in both penalized results (Figures 8.10 (d) and (e)), the power is only modified on the top lines of the path. It indeed seems that adapting the path is easier than modifying the power, which is kept whenever the modification of the path gets too complicated: at the top and bottom of the square. When it comes to controlling the jumps, it appears that the total variation control acts against the penalization of intermediate power values. Yet, it still allows a correct bang-bang control while drastically reducing the number of jumps (TV from 57 to 10). On the other hand, in accordance to Remark 8.3, the H^1 regularization has a very limited impact on the power oscillations.



(a) Initialization

(b) Result from path optimization (same Figure than (a))

(c) "Relaxed test"

(d) "Penalization test"

(e) "Penalization with continuation test"

(f) "Penalization test, $l_{\mathrm{TV}} = 100+$"

(g) "Penalization test, H^1-regularization"

(h) Temperature colorbar (K)

(i) Power colorbar

Figure 8.11: Coupled power and path optimization starting from zigzag9BIS-ALU (aluminium)

Case	L_{F} (m)	$\overline{C}_{\phi}$	$\overline{C}_{\mathrm{M,D_S}}$	$\overline{C}_{\mathrm{M,D\setminus D_S}}$	TV
initialization ($\zeta = 0.5$)	1.159e−2	4.20e−7	0.00	2.84e−5	0
path optimization	1.159e−2	4.20e−7	0.00	2.84e−5	0
relaxed test	1.179e−2	2.63e−8	0.00	4.13e−8	11.4
penalization test	1.154e−2	2.60e−4	7.06e−10	1.61e−5	12.8
penalization with continuation test	1.249e−2	7.91e−8	0.00	2.08e−7	40.1
penalization test, $l_{\mathrm{TV}} = 100+$	1.138e−2	2.40e−4	0.00	1.25e−5	12.6
penalization test, H^1-reg	1.154e−2	2.60e−4	7.37e−10	1.61e−5	12.8

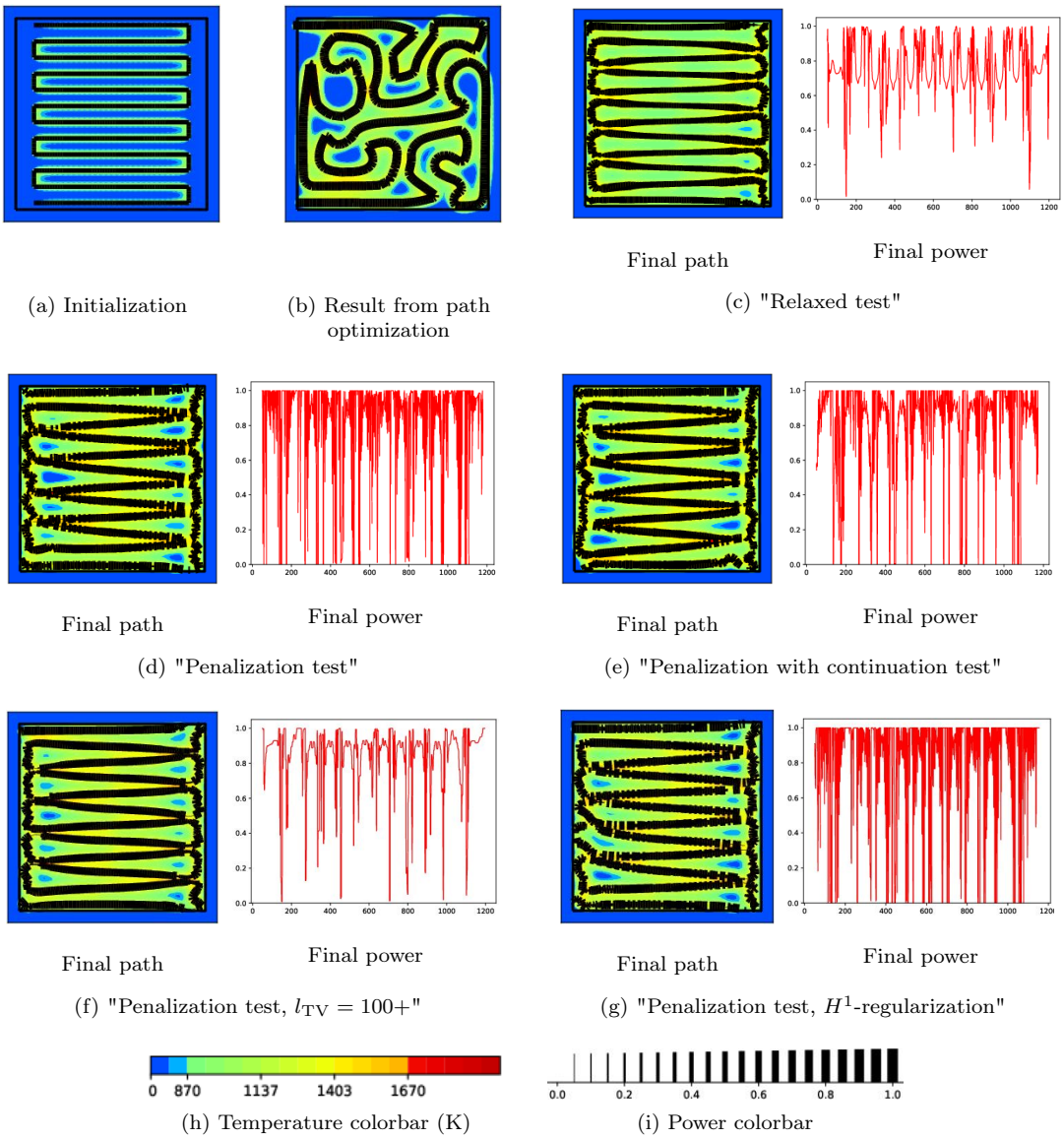
Table 8.4: Comparison of the cost and constraints of the coupled optimization for the final path starting from zigzag9BIS-ALU (aluminium)

This test starting from zigzag9bis-ALU (Figure 8.11 and Table 8.4) is not easy for the optimization process: the initialization is already very good. An improvement would then correspond to a slight modification of the power variable. This is indeed what can be observed in Figure 8.11(c). Adding the penalization of the intermediate values, deteriorates the results: on the first iterations, modifications are made to improve the path. In the meantime, the penalization constraint is activated, leading the minimizing sequence towards a different local minimum than the initialization was. Despite the non fulfillment of the constraints, it seems that the classical penalization test gives better results in terms of bang bang properties than the penalization with continuation test (and of course than the relaxed test). As for the total variation control, this test does not allow any clear conclusion.

Case	L_{F} (m)	$\overline{C}_{\phi}$	$\overline{C}_{\mathrm{M,D_S}}$	$\overline{C}_{\mathrm{M,D\setminus D_S}}$	TV
initialization ($\zeta = 0.5$)	1.325e−2	3.25e−2	0.00	0.00	0
path optimization	1.247e−2	3.26e−3	3.37e−4	2.95e−3	0
relaxed test	1.579e−2	6.53e−6	3.04e−7	3.12e−6	56.7
penalization test	1.592e−2	1.39e−4	8.80e−6	3.10e−5	196
penalization with continuation test	1.615e−2	2.12e−4	1.10e−5	2.82e−5	131
penalization test, $l_{\mathrm{TV}} = 100+$	1.596e−2	3.38e−5	3.78e−6	2.81e−5	46.0
penalization test, H^1-reg	1.583e−2	8.70e−5	4.36e−6	3.39e−5	212

Table 8.5: Comparison of the cost and constraints of the coupled optimization for the final path starting from zigzag12-TI (titanium)

As in the path only optimization, the titanium test case is more difficult to optimize (see Figure 8.12 and Table 8.5). Indeed, the lower conductivity induces rapid power oscillations. This leads to a less satisfied penalization of the intermediate powers. Yet, its effect is very clear when comparing the final power of the penalization tests to the one resulting from the relaxed test. As for the length minimization and satisfaction of the constraints, it seems that, in this physical context too, the addition of the power variable helps the constraints. Indeed, if the relaxed test result has a higher length than the optimized path result, the constraints are smaller. Moreover, the final path is a lot closer to the initialization. As for the addition of the total variation control, the conclusions are very close to the aluminium ones.



(a) Initialization (b) Result from path optimization (c) "Relaxed test"

(d) "Penalization test" (e) "Penalization with continuation test"

(f) "Penalization test, $l_{\mathrm{TV}} = 100+$" (g) "Penalization test, H^1-regularization"

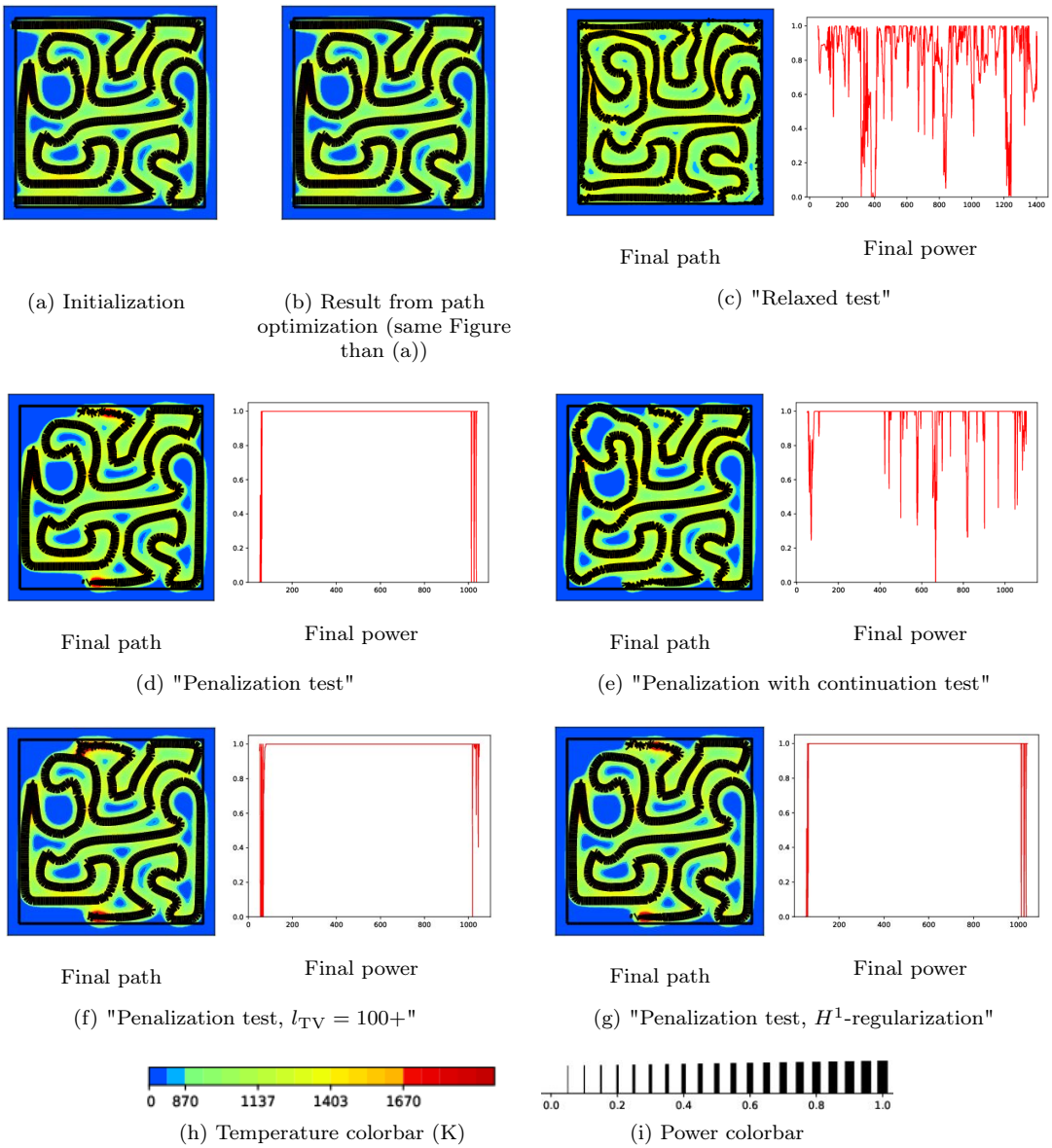
(h) Temperature colorbar (K) (i) Power colorbar

Figure 8.12: Coupled power and path optimization starting from zigzag12-TI (titanium)

Case	L_{F} (m)	$\overline{C}_{\phi}$	$\overline{C}_{\mathrm{M,D_S}}$	$\overline{C}_{\mathrm{M,D\backslash D_S}}$	TV
initialization ($\zeta = 0.5$)	1.247e−2	3.26e−3	3.37e−4	2.95e−3	0
path optimization	1.247e−2	3.26e−3	3.37e−4	2.95e−3	0
relaxed test	1.586e−2	2.57e−6	6.11e−8	1.54e−6	63.5
penalization test	1.201e−2	1.31e−2	7.78e−4	3.06e−3	10.0
penalization with continuation test	1.276e−2	3.50e−3	1.12e−4	4.05e−3	26.9
penalization test, $l_{\mathrm{TV}} = 100+$	1.210e−2	1.29e−2	1.12e−3	3.47e−3	10.6
penalization test, H^1-reg	1.201e−2	1.31e−2	7.78e−4	3.06e−3	10.0

Table 8.6: Comparison of the cost and constraints of the coupled optimization for the final path starting from zigzag12BIS-TI (titanium)

The final test, starting from zigzag12bis-TI (see Figure 8.13 and Table 8.6) leads to the same conclusion than the test in the aluminium test case, starting from zigzag9bis-ALU. Indeed, the optimization starts



(a) Initialization

(b) Result from path optimization (same Figure than (a))

(c) "Relaxed test"

(d) "Penalization test"

(e) "Penalization with continuation test"

(f) "Penalization test, $l_{\mathrm{TV}} = 100+$"

(g) "Penalization test, H^1-regularization"

(h) Temperature colorbar (K)

(i) Power colorbar

Figure 8.13: Coupled power and path optimization starting from zigzag12BIS-TI (titanium)

from an already quite good path. If adding the power optimization improves the results (Figure 8.13(c)), the penalization of the intermediate power values brings the minimizing sequence to a new local minimum not as good as the initialization.

8.3.6 Conclusion

The path optimization in the steady state context proposed in Chapter 6 led to the assumption that the number of connected components available to optimize the path impacts the results. The algorithm developed in this section, based on physical considerations, seems to allow for the addition of this feature as an optimization variable. Indeed, especially in the aluminium case where the conductivity is higher, the relaxation penalization technique chosen allows for the clear splitting of the path into different portions. If this optimization is complicated by the decrease of the conductivity, the results in the titanium case still allow for telling the difference between different parts. As expected, the resulting paths can be quite degenerated with the introduction of several variations to perfectly match the constraints. The total variation algorithm proposed is efficient in controlling this. Yet, it does not allow for mastering a precise number of jumps and its improvements are part of the perspectives.

8.4 TOPOLOGY OPTIMIZATION OF THE PATH (STEADY STATE CONTEXT)

Besides the very physical interpretation developed above, the source's switching on and off can be dealt with geometrical considerations. Indeed, it corresponds to changing the path topology, modifying its number of connected components. This section aims, in the steady state context only, at exploring this second point of view.

Topological differentiation has been widely developed to complement geometric shape optimization techniques [47, 153, 175] (for further information, see [164] and references therein). Mainly used to create holes or inclusions in an object modifying its mechanical properties, it has also been adapted to the modification of source terms in partial differential equation [54, 55]. Most of the time the hole or inclusion is a volumic set, namely a small bounded open set, but it can also be a lower dimensional set like a segment as for cracks detection for example [22, 120, 176, 188].
The aim of this section is to adapt the notion of topological derivative to the one dimensional steady path in a two dimensional working space. Note that the object which topology is modified is involved in the source term of the physical problem and the theory developed in this section is mainly based on [54].

The Augmented Lagrangian steady optimization problem considered is recalled

$$\min_{\Gamma\subset D} L_{\mathrm{F}}(\Gamma) = \int_\Gamma ds + l_C C + \frac{c}{2}C^2$$
$$\text{such that} \quad \begin{cases} C(\Gamma) = l_\phi \displaystyle\int_D \left[(y_\phi - y)^+\right]^2 dx + l_{\mathrm{M}} \int_D \left[(y - y_{\mathrm{M}})^+\right]^2 dx \\ \\ \forall \phi \in H^1(D), \quad \displaystyle\int_D (\lambda \nabla y \nabla \phi + \beta\,(y - y_{\mathrm{ini}})\,\phi)\,dx - P\int_\Gamma \phi ds = 0. \end{cases} \tag{8.4.1}$$

Remark 8.6. We recall that the maximum temperature constraint C_{M} gathers both the maximum temperature out of the domain constraint $C_{\mathrm{M,D\setminus D_S}}$ and the maximum temperature in the domain constraint $C_{\mathrm{M,D_S}}$.

8.4.1 A notion of topological derivative for lines in the plane

We now introduce a notion of topological derivative for lines in the plane. We consider a simplified optimization problem, in which the objective function is

$$J(\Gamma) = \int_\Gamma ds + \int_D j(y)dx, \tag{8.4.2}$$

with $j : \mathbb{R} \to \mathbb{R}$, such that it exists a constant $C \geq 0$ satisfying $\forall u \in \mathbb{R}$, $|j''(u)| \leq C$, $|j'(u)| \leq C(1 + |u|)$, $|j(u)| \leq C(1 + |u|^2)$.

Consider two different perturbations of the path: one consisting in removing part of the existing path, defined by Definition 8.1 and illustrated by Figure 8.14. The second one consists in adding a new connected component to the existing path, defined by Definition 8.2 and illustrated by Figure 8.15.

Definition 8.1. Let $\epsilon > 0$. A path perturbation for a path Γ is of type $\mathcal{R}$ when it is a curvilinear section $\mathcal{C}^\epsilon_{x_0}$ of the path Γ such that

$$\mathcal{C}^\epsilon_{x_0} = \{\gamma_{x_0}(t), t \in [0, \epsilon]\}, \tag{8.4.3}$$

with γ_{x_0} a restriction of the path parametrization starting at $x_0 \in \Gamma$ ($\gamma_{x_0}(0) = x_0$). The perturbed path $\Gamma^\epsilon_{x_0}$ is then defined as $\Gamma^\epsilon_{x_0} = \Gamma \setminus \mathcal{C}^\epsilon_{x_0}$ and its number of connected components if increased by one with respect to the number for the path Γ.

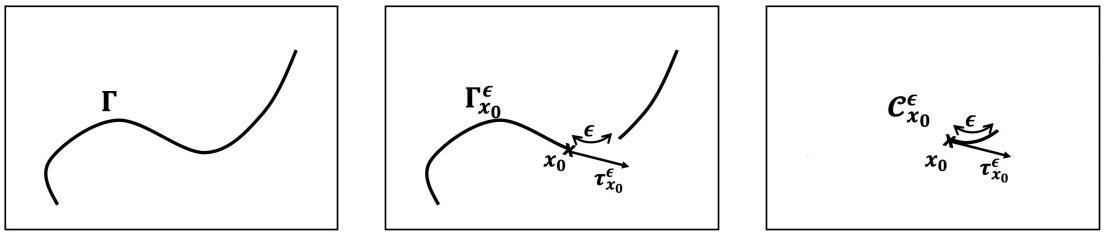
Remark 8.7. We define the set of generic points of Γ as the points at which the curvature is different from 0 and the set of exceptional points those such that the curvature is 0. Assuming that the measure of the set of exceptional points is 0, we can consider that, for any path perturbation of type $\mathcal{R}$, the starting point x_0 is a generic point. Thus, for ϵ small enough, the curve $\mathcal{C}^\epsilon_{x_0}$ does not include any inflexion point.

Definition 8.2. Let $\epsilon > 0$. A path perturbation for a path Γ is of type $\mathcal{A}$ when it is a segment $\mathcal{S}^\epsilon_{x_1}$ such that

$$\mathcal{S}^\epsilon_{x_1} = \{\gamma_{x_1}(t) = x_1 + t\tau^\epsilon_{x_1}, t \in [0, \epsilon]\}, \tag{8.4.4}$$

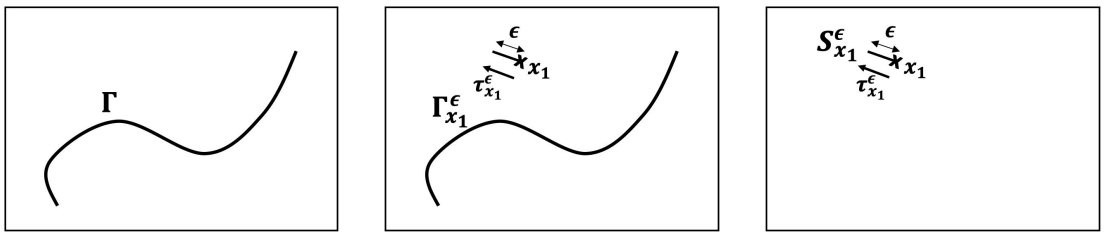
with $x_1 \in D$ and $\tau^\epsilon_{x_1}$ any normalized direction. The perturbed path is then defined as $\Gamma^\epsilon_{x_1} = \Gamma \cup \mathcal{S}^\epsilon_{x_1}$ and its number of connected components if increased by one with respect to the number for the path Γ.

Remark 8.8. A path perturbation of type $\mathcal{A}$ can be added anywhere in the working domain D provided that $\mathcal{S}^\epsilon_{x_1} \subset D$. In particular, it can cross or even follow the path Γ.



(a) Path Γ (b) Perturbed path $\Gamma^\epsilon_{x_0}$ (2 connected components) (c) Path $\mathcal{C}^\epsilon_{x_0}$

Figure 8.14: Removing a small part of the path



(a) Path Γ (b) Perturbed path $\Gamma^\epsilon_{x_1}$ (2 connected components) (c) Path $\mathcal{S}^\epsilon_{x_1}$

Figure 8.15: Adding a small segment to the path

Considering that ϵ is very small, the objective is to determine the sensibility of the optimization problem to infinitesimal perturbations.

Proposition 8.2. *Let Γ be a curve satisfying Definition 3.1. Consider $J(\Gamma)$ satisfying (8.4.2). Then, the following asymptotic development holds for a perturbation of type $\mathcal{R}$*

$$J(\Gamma^\epsilon_{x_0}) = J(\Gamma) - (1 - Pp(x_0))\,\epsilon + o(\epsilon), \tag{8.4.5}$$

and for a perturbation of type $\mathcal{A}$

$$J(\Gamma^\epsilon_{x_1}) = J(\Gamma) + (1 - Pp(x_1))\,\epsilon + o(\epsilon), \tag{8.4.6}$$

with $p \in H^1(D)$ solution the adjoint function

$$\begin{cases} -\lambda \Delta p + \beta p = -j'(y) & in\, D \\ \lambda \partial_n p = 0 & on\, \partial D. \end{cases} \tag{8.4.7}$$

The term in ϵ in the derivatives given in (8.4.5) and (8.4.6) is called the topological gradient.

Preliminary to the proof, we define the Hilbert space $\mathcal{D}^{1,2}(\mathbb{R}^2)$, called the Beppo-Levi or Deny-Lions space:

Definition 8.3. The Beppo-Levi space is defined as the closure of the space $\mathcal{D}\left(\mathbb{R}^2\right)$ of infinitely smooth functions with compact support, closure with respect to the L^2-norm of the gradient.

$$\mathcal{D}^{1,2}(\mathbb{R}^2) = \overline{\mathcal{D}(\mathbb{R}^2)}^{\|\nabla \cdot\|_{L^2(\mathbb{R}^2)}}.$$

The following lemma supplement this definition [54, 55].

Lemma 8.1.

$$\begin{aligned}\mathcal{D}^{1,2}(\mathbb{R}^2) &= \left\{\phi \in H^1_{loc}\left(\mathbb{R}^2\right); \nabla\phi \in L^2(\mathbb{R}^2)^2\right\} \\ &= \left\{\phi; \frac{\phi}{(r+1)\log(r+2)} \in L^2(\mathbb{R}^2); \nabla\phi \in L^2(\mathbb{R}^2)^2\right\}.\end{aligned}$$

The scalar product associated to this space is $\forall \phi, \psi \in \mathcal{D}^{1,2}(\mathbb{R}^2)$, $(\psi,\phi)_{\mathcal{D}^{1,2}(\mathbb{R}^2)} = \displaystyle\int_{\mathbb{R}^2} \nabla\phi \cdot \nabla\psi dx$. A final Lemma is defined.

Lemma 8.2. *Consider the problem*

$$\left\{\begin{array}{ll} -\lambda\Delta w = P\chi_S & in \mathbb{R}^2 \\ |\nabla w| \to 0 & x \to \infty. \end{array}\right. \tag{8.4.8}$$

It admits a unique solution $w \in \mathcal{D}^{1,2}(\mathbb{R}^2)$ and there exists a constant $C \geq 0$ such that, $\forall x \in \mathbb{R}^2$,

$$|\nabla w(x)| \leq \frac{C}{1+|x|}.$$

PROOF. (LEMMA 8.2).
let $\Phi \in \mathcal{D}^{1,2}(\mathbb{R}^2)$ and let B a closed ball such that $S \subset B$. Then, $\Phi \in H^1(B)$ and the trace theorem can be applied on the segment S leading to

$$\left|\int_{\mathbb{R}^2} P\chi_S \Phi dx\right| \leq CP\|\Phi\|_{H^1(B)} \leq C\left(\|\nabla\Phi\|_{L^2(\mathbb{R}^2)} + \|\Phi \mathbb{1}_B\|_{L^2(\mathbb{R}^2)}\right).$$

Finally, following the characterization given in Lemma 8.1,

$$\begin{aligned}\|\Phi\mathbb{1}_B\|_{L^2(\mathbb{R}^2)} &= \int_{\mathbb{R}^2} \mathbb{1}_B (|x|+1)\log(|x|+2)\frac{\Phi}{(|x|+1)\log(|x|+2)}dx \\ &\leq C\|(|x|+1)\log(|x|+2)\|_{L^2(B)}\|\nabla\Phi\|_{L^2(\mathbb{R}^2)}.\end{aligned}$$

Since the function $x \mapsto (|x|+1)\log(|x|+2) \in L^2(B)$, the linear form $\Phi \mapsto \displaystyle\int_{\mathbb{R}^2} P\chi_S\Phi dx$ is continuous on the Beppo Levi space and an application of Lax-Milgram finally shows that $w \in \mathcal{D}^{1,2}(\mathbb{R}^2)$.

The proof related to the bounds on the gradient is given in [54, 55]. □

PROOF. (PROPOSITION 8.2).
We first focus on perturbations of type $\mathcal{R}$. The perturbed temperature y_ϵ is defined as the solution of the perturbed problem:

$$\left\{\begin{array}{ll} -\lambda\Delta y_\epsilon + \beta(y_\epsilon - y_{ini}) = P\chi_{\Gamma^\epsilon_{x_0}} & \text{in } D \\ \lambda\partial_n y_\epsilon = 0 & \text{on } \partial D, \end{array}\right. \tag{8.4.9}$$

and we set $v_\epsilon = y - y_\epsilon$, satisfying

$$\left\{\begin{array}{ll} -\lambda\Delta v_\epsilon + \beta v_\epsilon = P\left(\chi_\Gamma - \chi_{\Gamma^\epsilon_{x_0}}\right) = P\chi_{\mathcal{C}^\epsilon_{x_0}} & \text{in } D \\ \lambda\partial_n v_\epsilon = 0 & \text{on } \partial D. \end{array}\right. \tag{8.4.10}$$

Then, it exists $\tilde{y}$ such that $\forall x \in D$, $\tilde{y}(x) \in [y(x), y_\epsilon(x)]$ (non ordered) and

$$\begin{aligned}J(\Gamma^\epsilon_{x_0}) &= \int_\Gamma ds - \int_{\mathcal{C}^\epsilon_{x_0}} ds + \int_D j(y - v_\epsilon)dx = \int_\Gamma ds - \int_{\mathcal{C}^\epsilon_{x_0}} ds + \int_D \left(j(y) - v_\epsilon j'(y) + \frac{1}{2}j''(\tilde{y})v_\epsilon \cdot v_\epsilon\right)dx \\ &= J(\Gamma) \underbrace{- \int_{\mathcal{C}^\epsilon_{x_0}} ds - \int_D j'(y)v_\epsilon dx}_{A} + \underbrace{\frac{1}{2}\int_D j''(\tilde{y})v_\epsilon \cdot v_\epsilon dx}_{B}.\end{aligned}$$

Computation of A :
First of all,

$$\int_{\mathcal{C}^\epsilon_{x_0}} ds = \int_0^\epsilon \|\gamma'_{x_0}(t)\|dt = \int_0^\epsilon \|\tau^\epsilon_{x_0} + O(t)\|dt = \int_0^\epsilon (1+o(1))\,dt = \epsilon + o(\epsilon). \tag{8.4.11}$$

If the adjoint function $p \in H^1(D)$ satisfies (8.4.7) and since $v_\epsilon \in H^1(D)$ solution of (8.4.10),

$$-\int_D j'(y)v_\epsilon dx = \int_D \lambda\nabla p\nabla v_\epsilon + \beta p v_\epsilon dx = \int_D P\chi_{\mathcal{C}^\epsilon_{x_0}} p dx = \int_{\mathcal{C}^\epsilon_{x_0}} Pp(s)ds.$$

Thus,

$$\begin{aligned} -\int_D j'(y)v dx &= \int_{\mathcal{C}^\epsilon_{x_0}} Pp(s)ds = P\int_0^\epsilon p\left(x_0 + s\tau^\epsilon_{x_0} + o(t)\right)ds \\ &= P\int_0^\epsilon \left(p(x_0) + s\nabla p(x_0)\cdot\tau^\epsilon_{x_0} + o(s)\right)ds = Pp(x_0)\epsilon + o(\epsilon). \end{aligned} \tag{8.4.12}$$

Finally,

$$A = (-1 + Pp(x_0))\,\epsilon + o(\epsilon). \tag{8.4.13}$$

In case of a perturbation of type $\mathcal{A}$, the signs are simply modified.

Computation of B : The objective here is to show that $B = o(\epsilon)$. Because on the boundeness assumption on j, it exists $C \geq 0$ such that

$$B \leq C\|v_\epsilon\|^2_{L^2(D)}. \tag{8.4.14}$$

The function to control is finally v_ϵ. As usual, C denotes a positive constant, the precise value of which changes from one place to the other. For simplicity we assume that x_0 is a generic point on the curve, namely its curvature is not 0 (Remark 8.7). Then, for sufficiently small ϵ, there is no inflexion point on the curve. We define in the following $S_{\tilde\epsilon}$ the chord segment with the same endpoints as $\mathcal{C}^\epsilon_{x_0}$. The length of this segment is written $\tilde\epsilon = |S_{\tilde\epsilon}|$, which is of order ϵ, and the combination of the segment $S_{\tilde\epsilon}$ and $\mathcal{C}^\epsilon_{x_0}$ encloses a bounded open set, denoted by V_ϵ (Figure 8.16).

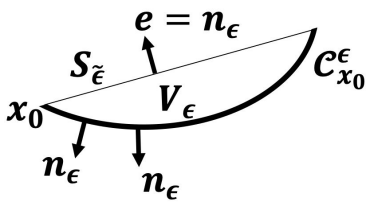


Figure 8.16: Chord to the segment removed

Finally, we define S the segment starting from 0, of length 1 with the same tangent vector than the segment $S_{\tilde\epsilon}$. Then,

$$\chi_{S_{\tilde\epsilon}}(x) = \chi_S\left(\frac{x-x_0}{\tilde\epsilon}\right).$$

We define w solution to (8.4.8). Then, $w \in \mathcal{D}^{1,2}(\mathbb{R}^2)$ and $\forall x \in \mathbb{R}^2$, $|\nabla w(x)| \leq \frac{C}{1+|x|}$.
We then set

$$w_{\tilde\epsilon}(x) = \tilde\epsilon^2 w\left(\frac{x-x_0}{\tilde\epsilon}\right).$$

Then, $\nabla w_{\tilde\epsilon}(x) = \tilde\epsilon\nabla w\left(\frac{x-x_0}{\tilde\epsilon}\right)$ and $\Delta w_{\tilde\epsilon}(x) = \Delta w\left(\frac{x-x_0}{\tilde\epsilon}\right) = -P\chi_{S_{\tilde\epsilon}}, \forall x \in \mathbb{R}^2$. Then,

$$\begin{cases} -\lambda\Delta v_\epsilon + \beta v_\epsilon = P\left(\chi_{\mathcal{C}^\epsilon_{x_0}} - \chi_{S_{\tilde\epsilon}}\right) - \lambda\Delta w_{\tilde\epsilon} & \text{in } D \\ \lambda\partial_n v_\epsilon = 0 & \text{on } \partial D, \end{cases}$$

which gives ($C \geq 0$):

$$\|v_\epsilon\|^2_{H^1(D)} \leq C\left(\left|\int_D \left(\chi_{\mathcal{C}^\epsilon_{x_0}} - \chi_{S_{\tilde\epsilon}}\right) v_\epsilon dx\right| + \left|\int_D \nabla v_\epsilon \nabla w_{\tilde\epsilon} dx\right| + \left|\int_{\partial D} \partial_n w_{\tilde\epsilon} v_\epsilon ds\right|\right). \tag{8.4.15}$$

By Cauchy-Schwarz, controlling the second term amounts to controlling $\|\nabla w_{\tilde\epsilon}\|_{L^2(D)}$ and, using Lemma 8.2,

$$\begin{aligned}\|\nabla w_{\tilde\epsilon}\|^2_{L^2(D)} &= \int_D |\nabla w_{\tilde\epsilon}|^2 dx = \tilde\epsilon^4 \int_{D/\tilde\epsilon} |\nabla w(x)|^2 dx \leq C\tilde\epsilon^4 \int_0^{|D|/\tilde\epsilon} \frac{r}{(1+r)^2} dr \\ &\leq C\tilde\epsilon^4 \left(\int_0^1 dr + \int_1^{|D|/\tilde\epsilon} \frac{dr}{r}\right) = O\left(\tilde\epsilon^3\right) = O\left(\epsilon^3\right).\end{aligned}$$

As for the term on the boundary, from Cauchy-Schwarz and the trace theorem,

$$\left|\int_{\partial D} \partial_n w_{\tilde\epsilon} v_\epsilon ds\right| \leq C\|\nabla w_{\tilde\epsilon}\|_{L^2(\partial D)} \|v_\epsilon\|_{H^1(D)}.$$

Then, since x_0 belongs to the open set D, there exists $C > 0$ such that $\forall x \in \partial D$, $|x - x_0| \geq C$, since $\nabla w_{\tilde\epsilon}(x) = \tilde\epsilon \nabla w\left(\dfrac{x - x_0}{\tilde\epsilon}\right)$,

$$\|\nabla w_{\tilde\epsilon}\|_{L^\infty(\partial D)} \leq C\tilde\epsilon^2 = O(\epsilon^2). \tag{8.4.16}$$

The final term to control is $\left|\displaystyle\int_D \left(\chi_{\mathcal{C}^\epsilon_{x_0}} - \chi_{S_{\tilde\epsilon}}\right) v_\epsilon dx\right|$. Let e be the normal to the segment $S_{\tilde\epsilon}$ oriented towards out of the set V_ϵ and n_ϵ the normal to V_ϵ pointing outwards (on $S_{\tilde\epsilon}$, $e = n_\epsilon$). Then,

$$\left|\int_D \left(\chi_{\mathcal{C}^\epsilon_{x_0}} - \chi_{S_{\tilde\epsilon}}\right) v_\epsilon dx\right| \leq \left|\int_{S_{\tilde\epsilon}} v_\epsilon e \cdot n_\epsilon ds + \int_{\mathcal{C}^\epsilon_{x_0}} v_\epsilon e \cdot n_\epsilon\right| + \left|\int_{\mathcal{C}^\epsilon_{x_0}} v_\epsilon \left(-e \cdot n_\epsilon - 1\right) ds\right|.$$

From Green formula,

$$\left|\int_{S_{\tilde\epsilon}} v_\epsilon e \cdot n_\epsilon ds + \int_{\mathcal{C}^\epsilon_{x_0}} v_\epsilon e \cdot n_\epsilon\right| = \left|\int_{\partial V_\epsilon} v_\epsilon e \cdot n_\epsilon dx\right| = \left|\int_{V_\epsilon} \nabla v_\epsilon \cdot e dx\right| \leq \|\nabla v_\epsilon\|_{L^2(D)} |V_\epsilon|.$$

Moreover, since $|V_\epsilon| \leq \tilde\epsilon^2 = O(\epsilon^2)$. Finally, because the curve $\mathcal{C}^\epsilon_{x_0}$ is smooth, it exists $\tilde s \in [0, \epsilon]$ such that $n_\epsilon(\gamma(s)) = -e$. Thus, $\forall s \in [0, \epsilon]$,

$$n_\epsilon(\gamma(s)) = n_\epsilon(\gamma(\tilde s)) + O(s - \tilde s) = -e + O(\epsilon).$$

This finally leads to, using the trace theorem on v_ϵ,

$$\begin{aligned}\left|\int_{\mathcal{C}^\epsilon_{x_0}} v_\epsilon \left(-e \cdot n_\epsilon - 1\right) ds\right| &\leq \|v_\epsilon\|_{L^2(\mathcal{C}^\epsilon_{x_0})} \sqrt{\int_0^\epsilon \left((-e) \cdot (-e + O(\epsilon)) - 1\right)^2 ds} \\ &\leq C\|\nabla v_\epsilon\|_{L^2(D)} \sqrt{O(\epsilon\epsilon^2)} \\ &\leq \|v_\epsilon\|_{H^1(D)} O(\epsilon^{3/2}).\end{aligned}$$

Following (8.4.15), this finally leads to $\|v_\epsilon\|_{L^2(D)} = O(\epsilon^2)$ which gives the result. As for a perturbation of type $\mathcal{A}$, the proof is exactly the same with an inversion of the signs for the computation of A. □

8.4.2 Numerical algorithm to introduce path splittings

To start with, we restrict the goal to splitting the path only, without any length nor objective function modification. A point $x \in \Gamma$ is chosen to be the cutting point, i.e. the new end point of one connected component and the new starting point of a second one. Using the development of the objective function (Proposition 8.2), to force its decrease, the point x must be chosen so that

$$1 - Pp(x) > 0 \quad \Longrightarrow \quad p(x) < \frac{1}{P}.$$

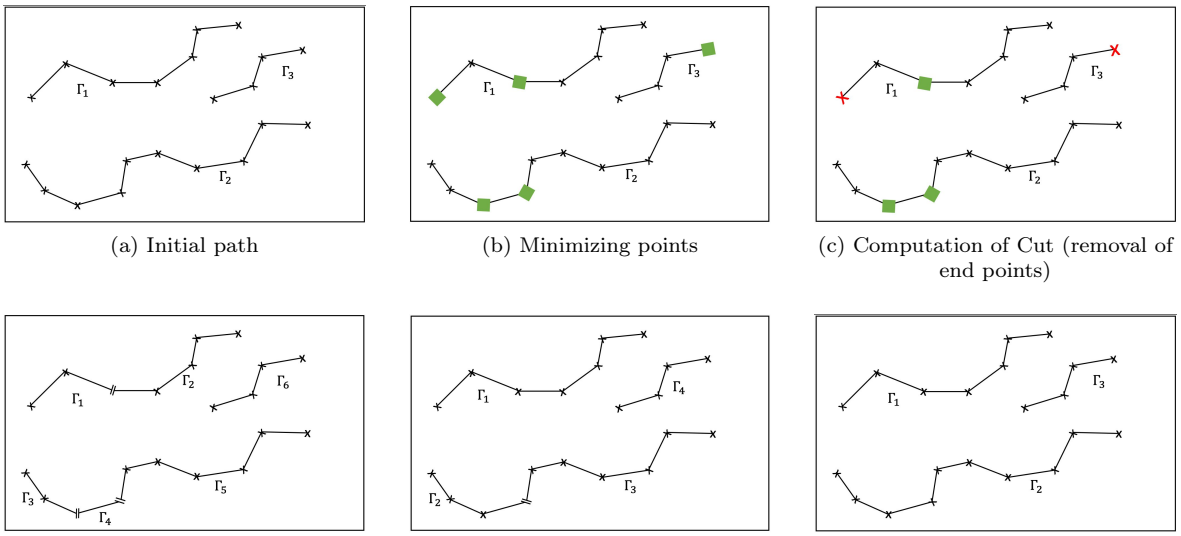
We now detail two different processes required for path splitting using the topological derivative. First, the cutting process itself must be designed. Then, a fusion process must be added. Indeed, to prevent the path from degenerating, we deliberately choose to gather two connected components in one if their endpoints are close enough.

Cutting process

The cutting process consists in splitting the path at specific points chosen with respect to (8.4.2). To do so, a first step consists in finding all the path discretization nodes u^* on the path Γ such that

$$u^* \in \text{Cut} = \arg\min_{u \in \Gamma} p(u), \tag{8.4.17}$$

and the corresponding minimum p^*. Note that, if an end points u_{ep} of a connected component of the path satisfies $p(u_{\text{ep}}) = p^*$, it is NOT added in Cut. Indeed, since the length must remain fixed during the process, cutting at an endpoint does not modify the path. The space Cut can thus be empty. If $p^* < \frac{1}{P}$, the path is cut at n_{cut} nodes, n_{cut} being fixed by the user to control the number of splits at one iteration (or related to the total number of connected components of the path). If the number n_{cut} is greater than the cardinal number of Cut the path is split at each node belonging to Cut (and the number of connected components is increased by card (Cut)). Else, n_{cut} points are randomly chosen within Cut to be the splitting points (and the number of connected components is increased by n_{cut}). Figure 8.17 illustrates this process with Figure 8.17(a) the initial path, Figures 8.17(b) and 8.17(c) the computation of Cut: in Figure 8.17, the endpoints of each connected component are removed from Cut. Finally, Figures 8.17(d) gives the final result in case of unlimited number of cuts per iteration or unlimited number of connected components whereas Figure 8.17(e) limits the number of cuts to 1 (or the number of connected components to 4) and Figure 8.17(f) limits the number of connected components to 3.



(a) Initial path

(b) Minimizing points

(c) Computation of Cut (removal of end points)

(d) Result in case of ilimited cuts

(e) Result in case of maximum 4 path connected components (cutting points chosen randomly in Cut)

(f) Result in case of maximum 3 path connected components (cutting points chosen randomly in Cut)

Figure 8.17: Cutting algorithm

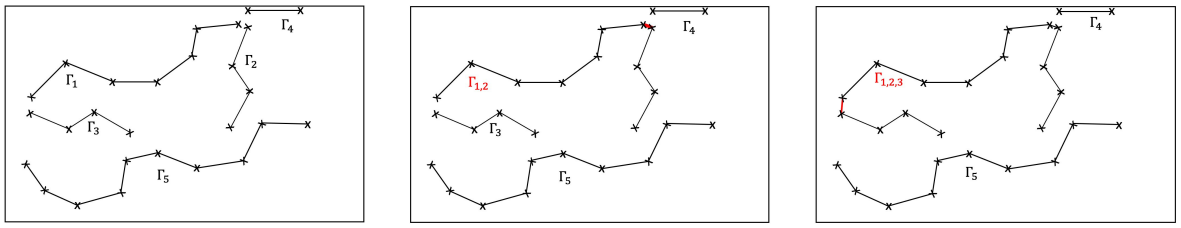
Remark 8.9. With this process, cutting at a path crossing is not a problem. Indeed, if two different nodes coincide, their adjoint value are the same and both belong to the space Cut. Yet, the cutting process can be applied all the same.

Remark 8.10. Note that there might be no cutting even if $p^* < \frac{1}{P}$. Indeed, if this minimum is reached at end points of the connected components of the path Γ only, they are not added to Cut which remains empty.

Fusion process

In order to limit the number of paths, a fusion process is designed so that, if the end points of two different path connected components are closer than the minimum distance between points d_{l}, they are connected and the resulting connected component is rediscretized. Figure 8.18 illustrates this process: (a) presents the initial path, on which one can notice that the connected components Γ_1, Γ_2, Γ_4 have close endpoints as well as Γ_1 and Γ_3. The closest being between Γ_1 and Γ_2, both connected components

are "gathered", reducing the number of connected components to 4 (Figure 8.18). If the endpoint of Γ_4 was close to the endpoints of Γ_1 and Γ_2 in Figure 8.18 (a), it is not anymore to the path $\Gamma_{1,2}$ (Figure 8.18(b)) and no path connection at this point will be made. Finally, $\Gamma_{1,2}$ and Γ_3 are gathered (Figure 8.18(c)).



(a) Initial path (b) First "fusion" (c) Second "fusion"

Figure 8.18: Melting algorithm

Numerical algorithm

The following algorithm is proposed: with F arbitrarily fixed by the user, every F iterations, the topological derivative is computed and the path points at which it is minimum are stored. If this minimum satisfies $p^* < \frac{1}{P}$, then the connected component is cut. The rest of the iterations consist in classic path advection but for each iteration straight before a cutting one at which the fusion process is done. Every cutting and fusion iterations are automatically accepted. As for the path connected components composed of two points connected components, if their length gets smaller than the distance d_l they are automatically deleted. Finally, the maximum number of cuts at each iteration is fixed to 10. Two different parameters can then be tested:

- the periodF_{top} at which happens a cutting iteration,

- the maximum number of connected components allowed (meaning that if the number of authorized cuts brings the number of connected components higher than the maximum allowed, not all the cuts are realized and a random choice will be made among them).

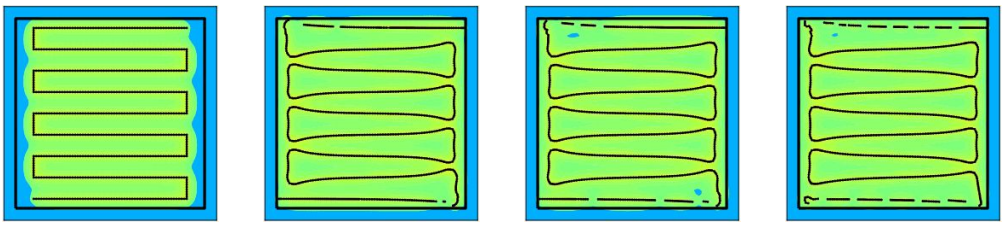
The algorithm is the following:

Illustration

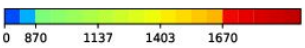
The algorithm is applied using an Augmented Lagrangian method with $l_C^0 = 1$,$c = 10$, $\nu_\Gamma = 15$. The pology of the path is modified every F_{top} iterations and the maximum number of connected components allowed in 15. The evolution of the path as well as the evolution of the different quantities (length, constraint, number of connected components composing the path) are shown in Figure 8.19 for the aluminium and Figure 8.20 for the titanium, respectively initialized by a zigzag with 9 lines and a zigzag with 12 lines.

Initialize the path Γ
Solve the heat equation and compute the objective functions and constraints
Compute the shape derivatives
for *iteration<number of iteration* **do**
  if *cutting iteration (iteration* $\mod F_{\text{top}} = 0$*)* **then**
    Apply the cutting procedure: the number of cutting can be controlled to avoid too many per iteration or to limit the total number of connected components
    Re-discretize the path in order to keep coherence in term of powers
  end
  if *fusion (iteration* $\mod F_{\text{top}} = F_{\text{top}} - 1$*)* **then**
    Gather together the connected components which endpoints are closer than d_1)
    Re-discretize the path in order to keep coherence in term of powers
  end
  if *advection iteration (rest of the iterations)* **then**
    Update the path in the classic way
    Re-discretize the path in order to keep coherence in term of powers
  end
  Compute the new objective function
  if *the new objective function is smaller than before (up to a tolerance) or if this was a fusion or cutting iteration* **then**
    Iteration accepted
    Update of the Lagrange multipliers and recompute the objective function
    Compute the shape and power derivatives
    Increase the step coefficient ($*1.2$) if the iteration was an advection one
  end
  else
    Reject the iteration
    Decrease the step coefficients ($*0.6$) if the iteration was an advection one
  end
end

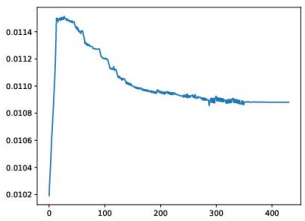
Algorithm 8.2: Shape optimization and cutting path algorithm



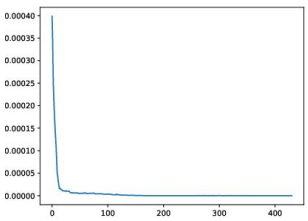
(a) Initial path (b) Iteration 50 (c) Iteration 100 (d) Result (386)



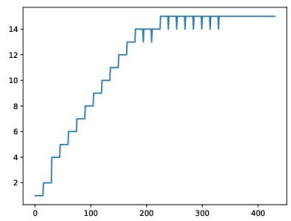
(e) Temperature colorbar (K)



(f) Length L_{F} evolution

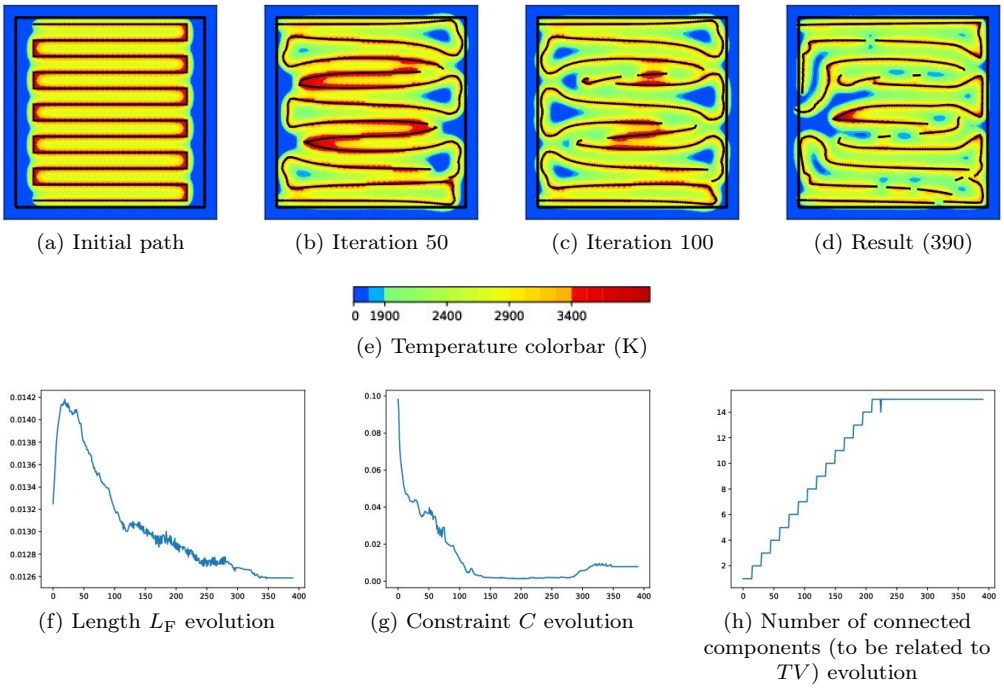


(g) Constraint C evolution



(h) Number of connected components (to be related to TV) evolution

Figure 8.19: Path and temperature as well as length, constraint and number of connected components evolution with respect to iterations, starting from zigzag9-ALU (aluminium)



(a) Initial path (b) Iteration 50 (c) Iteration 100 (d) Result (390)

(e) Temperature colorbar (K)

(f) Length L_F evolution (g) Constraint C evolution (h) Number of connected components (to be related to TV) evolution

Figure 8.20: Path and temperature as well as length, constraint and number of connected components evolution with respect to iterations, starting from zigzag12-TI (titanium)

These first results highlight once again the interest of splitting the path in the steady state context. Indeed, as for the power source control, it is clear that the algorithm takes advantage to optimizing more than one connected component. The number of connected components was actually limited to 15 and, in both cases, this number is saturated. With no restriction on the number of path connected components, the result would likely have been degenerated with many more pieces of the path. In the aluminium case, the result can be compared to the coupled power and path optimization. Indeed, as in Figure 8.10 (d) and (f), the cuttings happen on the top and bottom horizontal lines. These cuts reduce the maximum temperature constraint out of the square but also increase the phase constraint. As for the titanium case, because the number of connected components authorized is much lower in the topology optimization tests, the results are different (to compare with Figure 8.12 (d) and (f)). The number of connected components is better limited but the phase constraint is not fully satisfied. We also observe that the maximum temperature constraint out of the domain is not well satisfied either. This could be improved by modifying the Lagrangian initial coefficients of decreasing the maximum temperature allowed out of the domain. Such modifications remain part of the perspectives.

If this new technique is less physical than the previous one, it allows an easier control of the exact number of connected components. Yet, while the path length on which the source is off was taken into account in the power settings, this new technique does not give any information about the linking between the different pieces of the path.

8.4.3 Numerical algorithm to add paths connected components

If cutting is a good first step, the objective is also to add connected components to the path. In order to keep the same length while modifying the path topology, we now aim at "moving" part of the path. To do so, part of the existing path, of length L, is removed whereas a path of size L is added in the working domain D. To make sure that the objective function decreases, like in the cutting only case, we need to find:

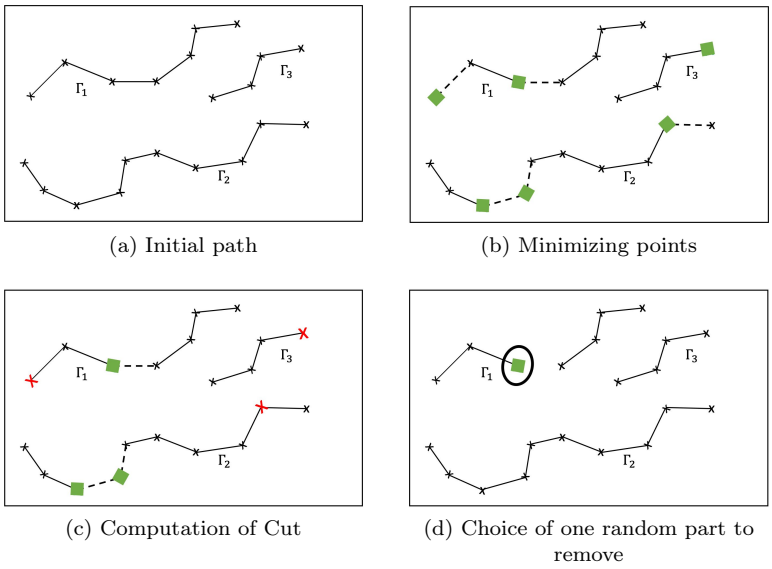
- $x_0 \in \Gamma$ such that $p(x_0) < \frac{1}{P}$ (cutting inequality).
- $x_1 \in D$ such that $p(x_1) > \frac{1}{P}$ (adding inequality).

If many cuts could previously be performed during one iteration, the transformation must now be better controlled and only one part is moved at each iteration. Note that the length is kept constant during

the topology modification iterations only. For the other iterations, the classic advection of the path is run and each connected component can lengthen or shrink.

Cutting process

The cutting process is a bit different than previously. Indeed, instead of only splitting, we now remove part of the path. Once the node point of the path discretization at which the cutting must be done is found, the path segment following this point is removed, creating two connected components. The length d of this element is stored. In order to really change the topology of the path, the first point as well as the last and one before last points of the path are removed from the points possibilities. Indeed, considering them would only mean decreasing the path length and not changing the topology (Figure 8.21). If the minimum of the adjoint was reached at these points only, then, the topology change process is aborted. Else, a random choice is made within the other points at which the minimum is reached (see Figure 8.21).



(a) Initial path

(b) Minimizing points

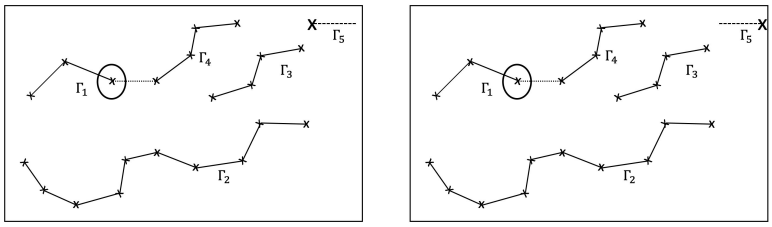
(c) Computation of Cut

(d) Choice of one random part to remove

Figure 8.21: Cutting algorithm for the add-removal process. (a)Initial path, (b)The points minimizing the adjoint and satisfying the cutting inequality on the path are found, the segment following them might be removed. (c)The points corresponding to endpoints or one before last point of the path components are removed from the list, since removing the corresponding segment would not modify the topology of the path but only decrease the length of one of the connected components. (d)Within the still available cutting points, one is randomly chosen. The corresponding segment is removed.

Addition process

The cutting process being defined, a segment must now be added. Its starting point is a point chosen randomly within the points of the working domain reaching the maximum of the adjoint. To keep constant path length, the segment added is of length d, corresponding to the length of the element removed from the existing path. As for the direction, since no information can be found in the topological derivative, it is arbitrarily fixed to e_x (or $-e_x$ in case the segment gets out of the working domain). This process is summed up in Figure 8.22.



(a) Adding a segment of size L starting from a point for which the direction e_x is available

(b) Adding a segment of size L starting from a point for which the direction e_x is not available ($-e_x$ is chosen)

Figure 8.22: Adding algorithm for the add-removal process

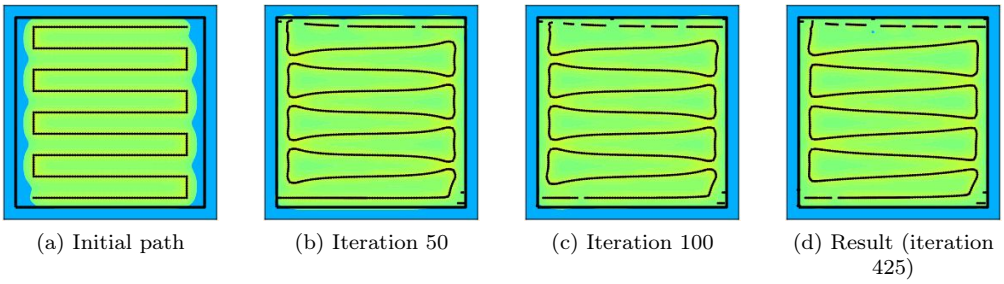
Algorithm 8.3 recaps the whole cutting-adding process. Note that in this algorithm, the added parts of path are composed of two points. Thus, if the advection make their length decrease in the iterations following their creation, they might simply disappear.

Find the set Cut $= \{\mathrm{x} \in \Gamma, \quad \text{s.t.} \quad \forall \mathrm{y} \in \Gamma, \mathrm{p(x)} \leq \mathrm{p(y)}\}$ and the minimum value of p reached by these points. The points are of course looked for in the path discretization nodes.
Find the set Add $= \{\mathrm{x} \in \mathrm{D}, \quad \text{s.t.} \quad \forall \mathrm{y} \in \mathrm{D}, \mathrm{p(x)} \geq \mathrm{p(y)}\}$ and the maximum value corresponding. The points are in practice looked for within the physical mesh nodes.
if $p(x_0) < \frac{1}{P}$ *and* $p(x_1) > \frac{1}{P}$ **then**
    Remove the starting point, last and one before last points of each connected components of the path from the set Cut
    if Cut $= \emptyset$ **then**
        Abort the topology changes and leave the path unmodified
    end
    else
        Choose one point randomly in Cut, remove the element el starting from this point (thus creating two connected components) and store its length d
        Choose one point x_1 randomly in Add
        if $\forall t \in [0, d], \quad s = x_1 + te_x \in D$ **then**
            Add the segment $\{x_1 + te_x, \quad t \in [0, d]\}$ to the path Γ
        end
        else
            Add the segment $\{x_1 - te_x, \quad t \in [0, d]\}$ to the path Γ
        end
    end
end
else
    Abort the topology changes and leave the path unmodified
end

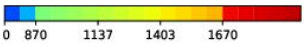
Algorithm 8.3: Add and remove algorithm

Illustration

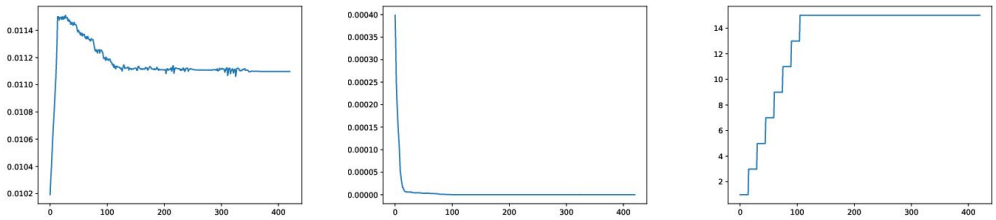
To illustrate the process, as in the previous case, the algorithm is applied using an Augmented Lagrangian method with $l_C^0 = 1$ and $c = 10$, $\nu_\Gamma = 15$. The topology of the path is modified every F_{top} iterations and the maximum number of connected components allowed is 15. The evolution of the path as well as the evolution of the different quantities (length, constraint, number of connected components composing the path) is given by Figure 8.23 for the aluminium (initialized by a zigzag with 9 lines) and Figure 8.24 for the titanium (initialized by a zigzag with 12 lines).



(a) Initial path (b) Iteration 50 (c) Iteration 100 (d) Result (iteration 425)

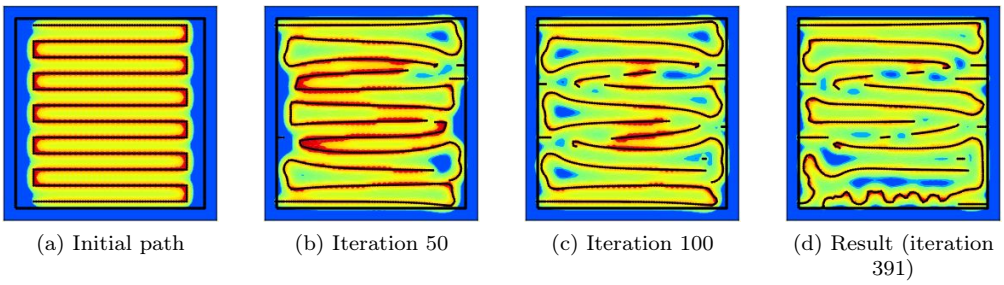


(e) Temperature colorbar (K)

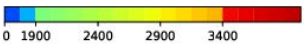


(f) Length L_{F} evolution (g) Constraint C evolution (h) Number of connected components (to be related to TV) evolution

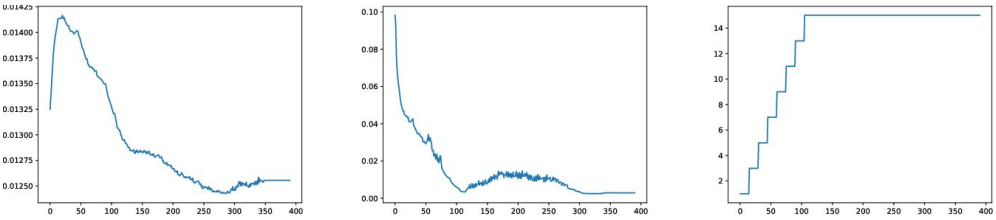
Figure 8.23: Path and temperature as well as length, constraint and number of connected components evolution with respect to iterations, starting from zigzag9-ALU (aluminium)



(a) Initial path (b) Iteration 50 (c) Iteration 100 (d) Result (iteration 391)



(e) Temperature colorbar (K)



(f) Length L_{F} evolution (g) Constraint C evolution (h) Number of connected components (to be related to TV) evolution

Figure 8.24: Path and temperature as well as length, constraint and number of connected components evolution with respect to iterations, starting from zigzag12-TI (titanium)

In this context again, the number of authorized connected components is saturated. Note that the connected components are mostly created on the sides of the square or where the phase constraint is not satisfied (Figure 8.24(c)). The aluminium results are very similar to the ones previously obtained (Figure 8.19) with path splitting on the top and bottom horizontal lines. The adding of paths also happens in

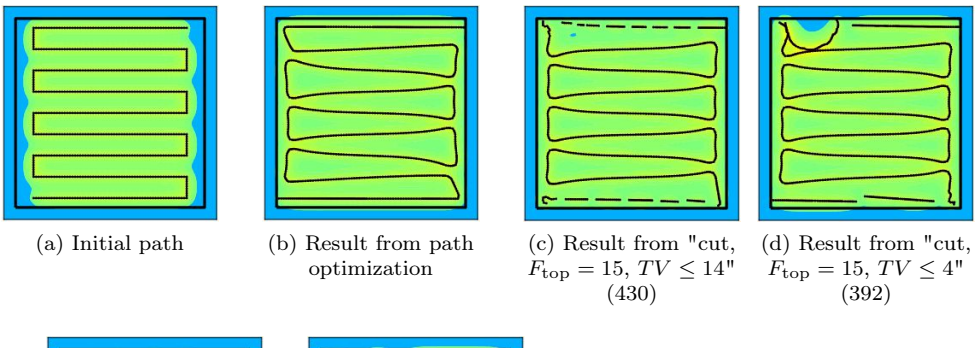
the bottom and upper parts of the working domain. The new result in the titanium context is different from the previous one. Once again, the pattern does not change much but the constraints seem to be better satisfied.

8.4.4 Numerical results

To further test the algorithm, for each of the initializations tested in the power context (zigzag with 9 lines, optimized path starting from a zigzag with 9 lines for the aluminium; zigzag with 12 lines, optimized path from a zigzag with 12 lines for the titanium), four optimizations are run, with the topology modifications performed every F_{top} iterations:

- "cut, $F_{\text{top}} = 15$, $TV \leq 14$": cutting process only, with the topology modification every 15 iterations and 15 connected components possible,
- "cut, $F_{\text{top}} = 15$, $TV \leq 4$": cutting process only, with the topology modification every 15 iterations and 5 connected components possible,
- "add-remove, $F_{\text{top}} = 15$, $TV \leq 14$" cutting and adding process, with the topology modification every 15 iterations and 15 connected components possible,
- "add-remove, $F_{\text{top}} = 15$, $TV \leq 4$": cutting and adding process, with the topology modification every 15 iterations and 5 connected components possible.

For each of the initializations (zigzag 9 lines and result from the path optimization starting from a zigzag with 9 lines in the aluminium case and zigzag with 12 lines and result from the path optimization starting from a zigzag with 12 lines in the titanium case), a Figure gives the 4 results and the numerical values are summed up in a Table. The first initialization is the zigzag with 9 lines. The resulting path shapes and temperatures are given by Figure 8.25 and Table 8.7.



(a) Initial path

(b) Result from path optimization

(c) Result from "cut, $F_{\text{top}} = 15$, $TV \leq 14$" (430)

(d) Result from "cut, $F_{\text{top}} = 15$, $TV \leq 4$" (392)

(e) Result from "add-remove, $F_{\text{top}} = 15$, $TV \leq 14$" (425)

(f) Result from "add-remove, $F_{\text{top}} = 15$, $TV \leq 4$" (411)

(g) Temperature colorbar (K)

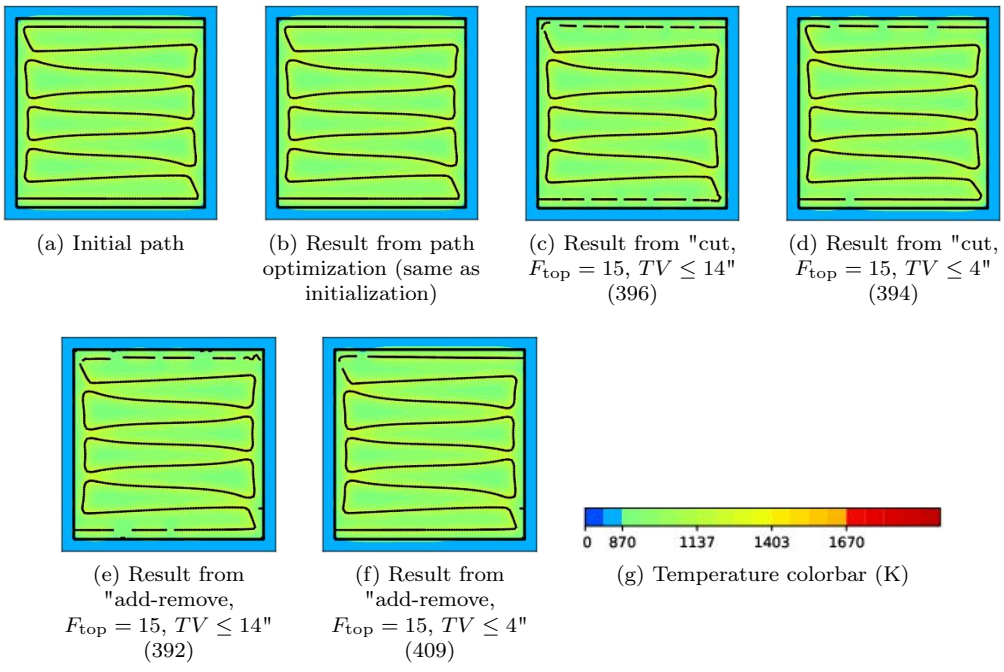
Figure 8.25: Path and temperature resulting from topology optimization, starting from zigzag9-ALU (aluminium)

Optimization case	L_F (m)	$\overline{C}_\phi$	$\overline{C}_{M,D_S}$	$\overline{C}_{M,D\setminus D_S}$	TV
initialization (classic)	1.019e−2	3.31e−4	0.00	3.09e−7	0.0
path optimization	1.159e−2	4.20e−7	0.00	2.84e−5	0.0
cut, F_{top}=15, $TV \leq 14$	1.088e−2	5.14e−8	0.00	8.73e−8	14.0
cut, F_{top}=15, $TV \leq 4$	1.163e−2	1.70e−5	0.00	2.62e−5	4.0
add-remove, $F_{top} = 15$, $TV \leq 14$	1.110e−2	8.21e−9	0.00	6.78e−10	14.0
add-remove, $F_{top} = 15$, $TV \leq 4$	1.155e−2	1.59e−7	0.00	1.18e−5	4.0

Table 8.7: Comparison of the cost and the constraints, topology optimization, starting from zigzag9-ALU (aluminium)

Even allowing only 5 connected components, the results shown in Figure 8.25 and Table 8.7 seem to be better than when optimizing the path without any variation of the number of connected components. Yet, it appears that the cutting-adding process is better than the cutting only, even if the randomness is higher. These results also allow for a first comparison with the algorithm using power (Figure 8.10 and Table 8.3). If it is irrelevant to compare the lengths (since the off path is taken into account in the power process and not in the topology optimization one), the final constraints are slightly better with the power process whereas the total variations are from far better controlled (since imposed) with topology optimization.

The second optimization process is initialized by the optimal path starting from zigzag9bis-ALU. The results are given by Figure 8.26 and Table 8.8.



(a) Initial path

(b) Result from path optimization (same as initialization)

(c) Result from "cut, $F_{top} = 15$, $TV \leq 14$" (396)

(d) Result from "cut, $F_{top} = 15$, $TV \leq 4$" (394)

(e) Result from "add-remove, $F_{top} = 15$, $TV \leq 14$" (392)

(f) Result from "add-remove, $F_{top} = 15$, $TV \leq 4$" (409)

(g) Temperature colorbar (K)

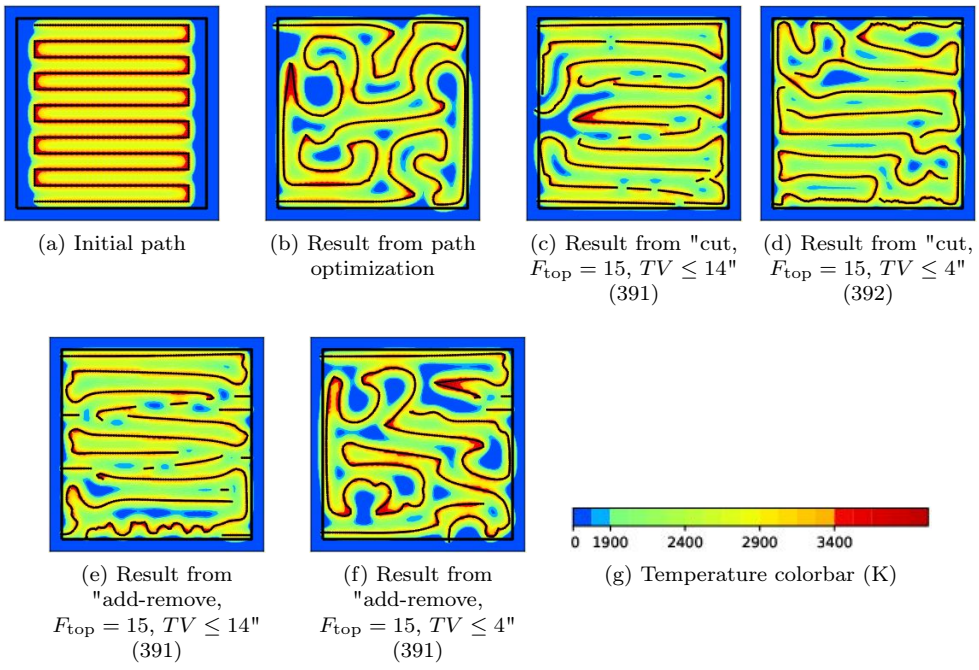
Figure 8.26: Path and temperature resulting from topology optimization, starting from zigzag9BIS-ALU (aluminium)

Optimization case	L_F (m)	$\overline{C}_\phi$	$\overline{C}_{M,D_S}$	$\overline{C}_{M,D\setminus D_S}$	TV
initialization (classic)	1.159e−2	4.20e−7	0.00	2.84e−5	0.0
path optimization	1.159e−2	4.20e−7	0.00	2.84e−5	0.0
cut, $F_{top} = 15$, $TV \leq 14$	1.105e−2	1.11e−7	0.00	2.17e−7	14.0
cut, $F_{top} = 15$, $TV \leq 5$	1.133e−2	3.88e−7	0.00	1.29e−5	4.0
add-remove, $F_{top} = 15$, $TV \leq 14$	1.111e−2	4.60e−8	0.00	5.51e−7	14.0
add-remove, $F_{top} = 15$, $TV \leq 5$	1.150e−2	1.43e−7	0.00	2.25e−5	4.0

Table 8.8: Comparison of the cost and the constraints, topology optimization, starting from zigzag9BIS-ALU (aluminium)

In this specific case (Figure 8.26 and Table 8.8), the results with topology optimization are better than in the classic path optimization (Figure 8.26 (a) and (b)): increasing the number of path connected components helps improving the path. Then, they are better than the results of the power process (Figure 8.11). The power based algorithm starts with a power variable at 1 and bringing this variable to 0 requires violating the penalization constraint. On the contrary with topology optimization, this process is realized abruptly and does not increase any function.

The titanium test case is now considered, with the initialization corresponding to a zigzag with 12 lines.



(a) Initial path

(b) Result from path optimization

(c) Result from "cut, $F_{top} = 15$, $TV \leq 14$" (391)

(d) Result from "cut, $F_{top} = 15$, $TV \leq 4$" (392)

(e) Result from "add-remove, $F_{top} = 15$, $TV \leq 14$" (391)

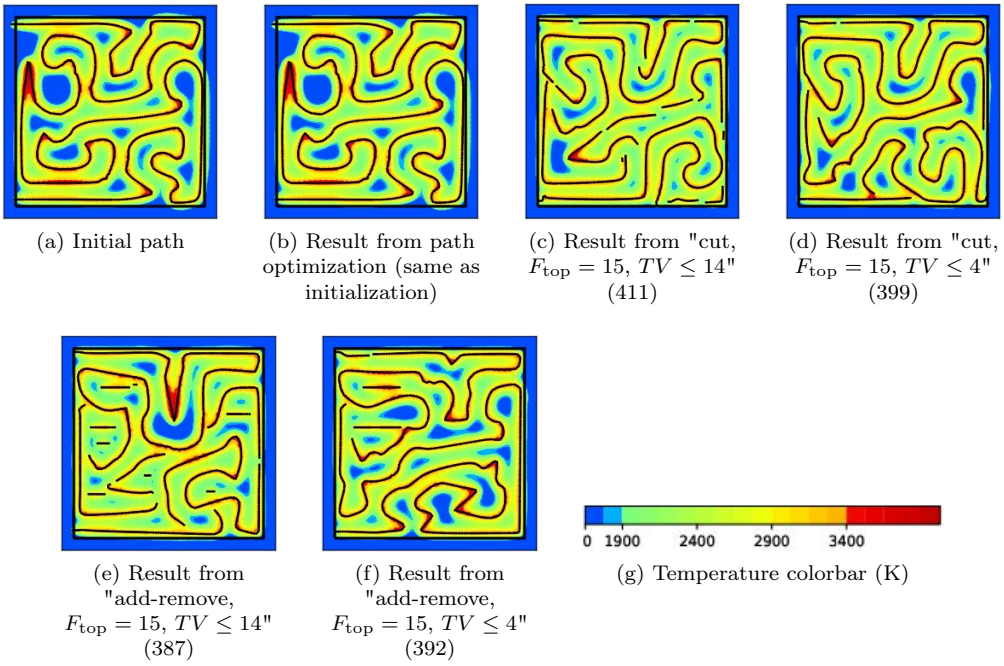
(f) Result from "add-remove, $F_{top} = 15$, $TV \leq 4$" (391)

(g) Temperature colorbar (K)

Figure 8.27: Path and temperature resulting from topology optimization, starting from zigzag12-TI (titanium)

Optimization case	L_{F} (m)	$\overline{C}_{\phi}$	$\overline{C}_{\mathrm{M,D_S}}$	$\overline{C}_{\mathrm{M,D\backslash D_S}}$	TV
initialization (classic)	1.325e−2	1.55e−2	5.11e−4	0.00	0.0
path optimization	1.247e−2	3.26e−3	3.37e−4	2.95e−3	0.0
cut, $F_{\mathrm{top}} = 15$, $TV \leq 14$	1.259e−2	7.48e−4	1.22e−4	1.10e−3	14.0
cut, $F_{\mathrm{top}} = 15$, $TV \leq 4$	1.219e−2	5.46e−3	4.91e−4	2.61e−3	4.0
add-remove, $F_{\mathrm{top}} = 15$, $TV \leq 14$	1.256e−2	3.86e−4	2.44e−5	2.38e−4	14.0
add-remove, $F_{\mathrm{top}} = 15$, $TV \leq 4$	1.234e−2	2.78e−3	3.07e−4	2.27e−3	4.0

Table 8.9: Comparison of the cost and the constraints, topology optimization, starting from zigzag12-TI (titanium)



(a) Initial path

(b) Result from path optimization (same as initialization)

(c) Result from "cut, $F_{\mathrm{top}} = 15$, $TV \leq 14$" (411)

(d) Result from "cut, $F_{\mathrm{top}} = 15$, $TV \leq 4$" (399)

(e) Result from "add-remove, $F_{\mathrm{top}} = 15$, $TV \leq 14$" (387)

(f) Result from "add-remove, $F_{\mathrm{top}} = 15$, $TV \leq 4$" (392)

(g) Temperature colorbar (K)

Figure 8.28: Path and temperature resulting from topology optimization, starting from zigzag12BIS-TI (titanium)

Optimization case	L_{F} (m)	$\overline{C}_{\phi}$	$\overline{C}_{\mathrm{M,D_S}}$	$\overline{C}_{\mathrm{M,D\backslash D_S}}$	TV
initialization (classic)	1.247e−2	3.26e−3	3.37e−4	2.95e−3	0.0
path optimization	1.247e−2	3.26e−3	3.37e−4	2.95e−3	0.0
cut, $F_{\mathrm{top}} = 15$, $TV \leq 14$	1.263e−2	3.06e−4	1.78e−5	1.57e−4	14.0
cut, $F_{\mathrm{top}} = 15$, $TV \leq 4$	1.261e−2	4.30e−4	2.20e−5	2.69e−4	4.0
add-remove, $F_{\mathrm{top}} = 15$, $TV \leq 14$	1.275e−2	6.48e−4	2.49e−4	5.13e−4	14.0
add-remove, $F_{\mathrm{top}} = 15$, $TV \leq 4$	1.265e−2	8.84e−4	6.93e−5	5.61e−4	4.0

Table 8.10: Comparison of the cost and the constraints, topology optimization, starting from zigzag12BIS-TI (titanium)

The conclusions are similar to the results obtained for the aluminium. In both cases, the final path is better when allowing modifications of the number of path connected components. We can even conclude

that the more components are allowed, the better the path is and we could imagine that without any limitations, the algorithm could ass a large number of small components. This idea is even stronger in the titanium case since the conductivity is lower and thus each path point induces a high temperature. For example in Figure 8.28(e), the algorithm should add a component in the "blue central zone". However, since the maximum number of connected components is reached, this cannot be done. Thus, since the total variation is only penalized in the power based algorithm and not imposed as in the topology optimization based one, the constraints are better satisfied in Figures 8.12 and 8.13. Yet, the number of maximum connected components is not imposed and can thus get high. Finally, note that the algorithm does not easily modify the "shape" of the added connected components which remain segments. A perspective would then be to add connected components with a non zero curvature but with a randomly chosen one (indeed the topological gradient does not provide any information on this).

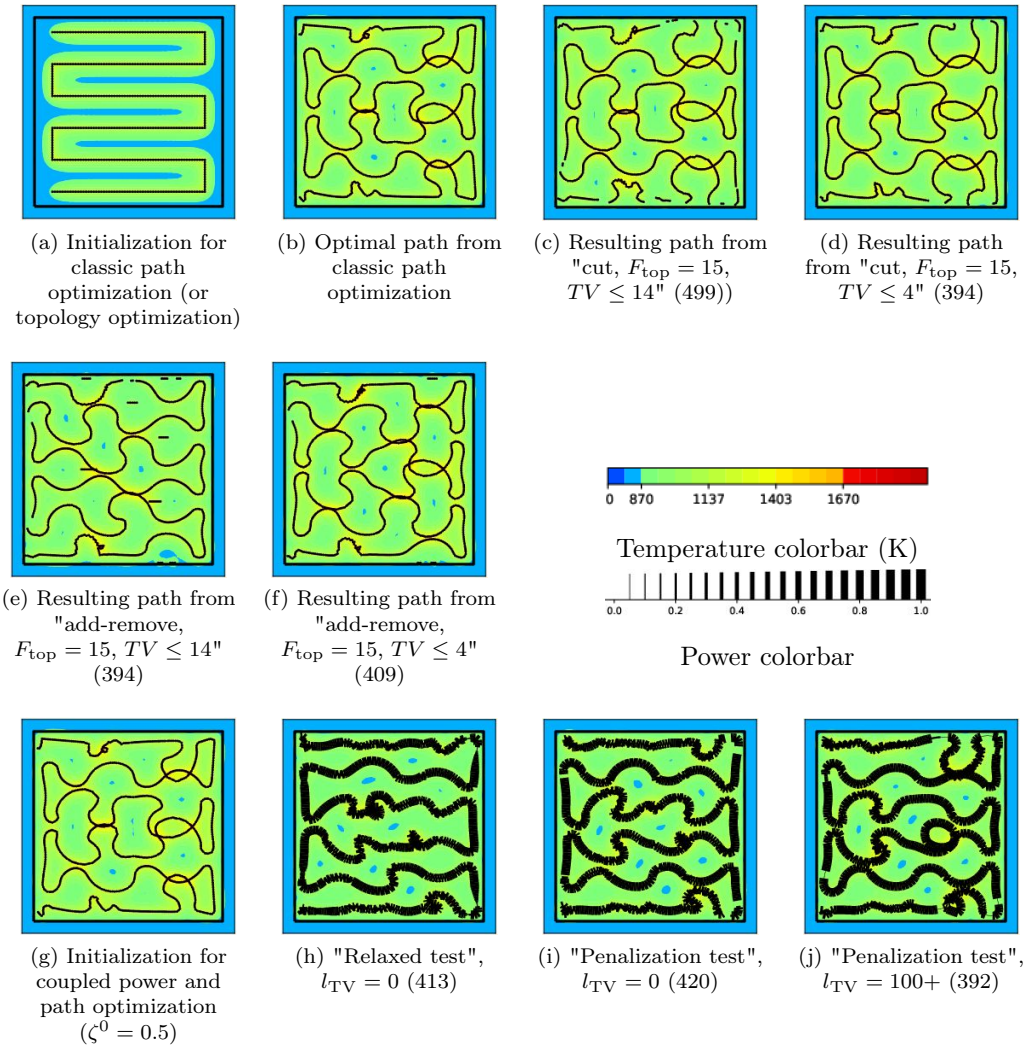
8.5 FURTHER COMPARISONS BETWEEN THE COUPLED POWER AND PATH OPTIMIZATION AND THE TOPOLOGY OPTIMIZATION METHODS

To further compare the algorithms, the results from different initializations are now compared. After running three different initializations for aluminium and titanium for a square geometry, the shape of the part to build is modified. In each case, 8 tests are run:

- simple path optimization (see Chapter 6),
- "cut, $F_{\text{top}} = 15$, $TV \leq 14$": cutting process only, with the topology modification every 15 iterations and 15 connected components possible,
- "cut, $F_{\text{top}} = 15$, $TV \leq 4$": cutting process only, with the topology modification every 15 iterations and 5 connected components possible,
- "add-remove, $F_{\text{top}} = 15$, $TV \leq 14$": cutting and adding process, with the topology modification every 15 iterations and 15 connected components possible,
- "add-remove, $F_{\text{top}} = 15$, $TV \leq 4$": cutting and adding process, with the topology modification every 15 iterations and 5 connected components possible,
- "relaxed test" (coupled power and shape optimization with $l_{\text{TV}} = 0$),
- "penalization test, $l_{\text{TV}} = 0$" (coupled power and path optimization with intermediate power values penalization with no restriction on the total variations),
- "penalization test, $l_{\text{TV}} = 100+$" (coupled power and path optimization with intermediate power values penalization and $l_{\text{TV}} = 100$ increased along the iterations).

8.5.1 Square geometry, aluminium context

In the aluminium context first, three initializations are considered: the zigzag with 6 lines (results in Figure 8.29 and Table 8.11), the spiral (Figure 8.30 and Table 8.12) and the contour with 4 lines (Figure 8.31 and Table 8.13). For the coupled power and shape optimization, the power value ζ is initialized to 0.5 in each case.



(a) Initialization for classic path optimization (or topology optimization)

(b) Optimal path from classic path optimization

(c) Resulting path from "cut, $F_{\text{top}} = 15$, $TV \leq 14$" (499))

(d) Resulting path from "cut, $F_{\text{top}} = 15$, $TV \leq 4$" (394)

(e) Resulting path from "add-remove, $F_{\text{top}} = 15$, $TV \leq 14$" (394)

(f) Resulting path from "add-remove, $F_{\text{top}} = 15$, $TV \leq 4$" (409)

(g) Initialization for coupled power and path optimization ($\zeta^0 = 0.5$)

(h) "Relaxed test", $l_{\text{TV}} = 0$ (413)

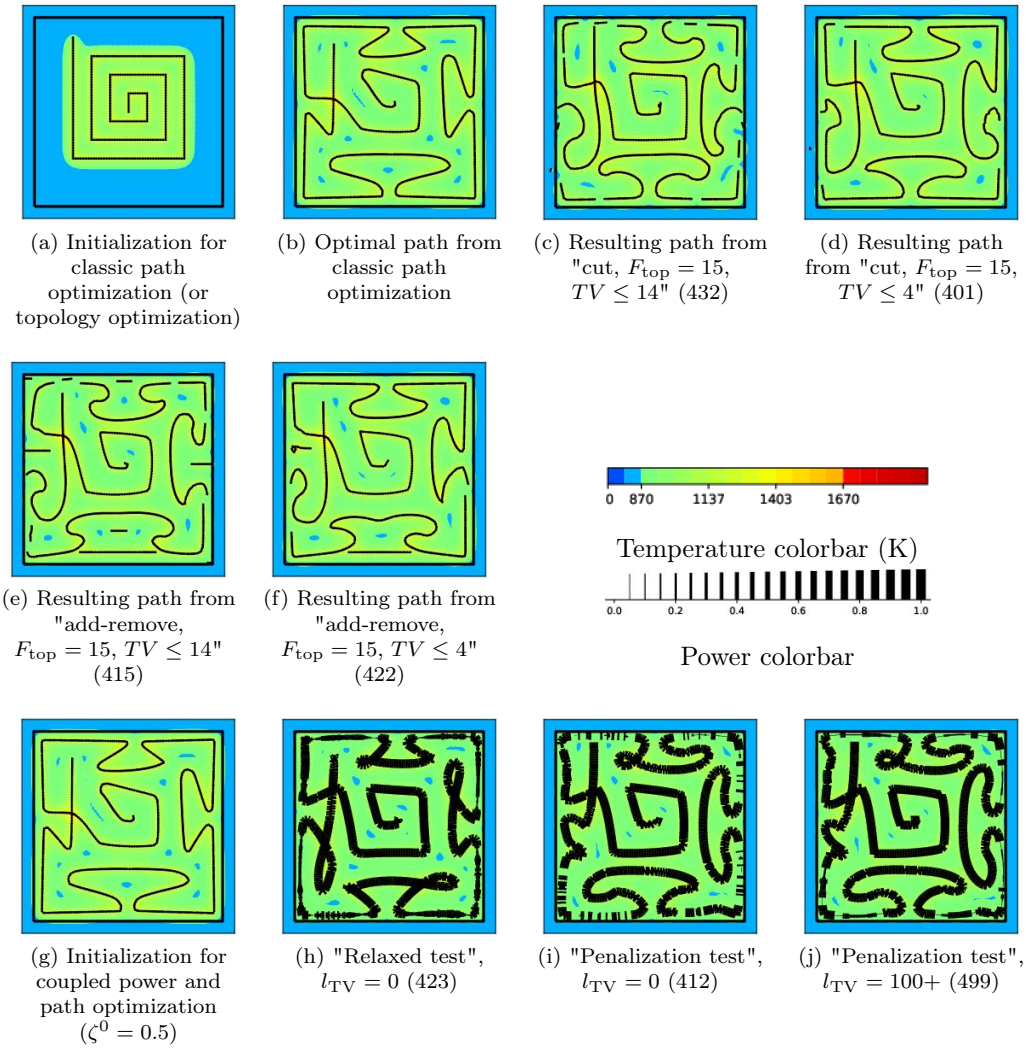
(i) "Penalization test", $l_{\text{TV}} = 0$ (420)

(j) "Penalization test", $l_{\text{TV}} = 100+$ (392)

Figure 8.29: Path and temperature results for the splitting tests starting from a zigzag with 6 lines (aluminium)

Optimization case	L_{F} (m)	$\overline{C}_{\phi}$	$\overline{C}_{\text{M,D}_\text{S}}$	$\overline{C}_{\text{M,D}\setminus\text{D}_\text{S}}$	TV
initialization (classic)	7.098e−3	5.85e−4	0.00	0.00	0.0
path optimization	1.265e−2	4.92e−7	0.00	2.17e−6	0.0
cut, $F_{\text{top}} = 15$, $TV \leq 14$	1.246e−2	6.01e−8	0.00	2.28e−7	14.0
cut, $F_{\text{top}} = 15$, $TV \leq 4$	1.254e−2	3.00e−7	0.00	1.62e−6	4.0
add-remove, $F_{\text{top}} = 15$, $TV \leq 14$	1.145e−2	8.80e−7	0.00	2.89e−6	13.0
add-remove, $F_{\text{top}} = 15$, $TV \leq 4$	1.272e−2	4.56e−7	0.00	1.27e−6	4.0
initialization ($\zeta = 0.5$)	7.098e−3	2.03e−3	0.00	0.00	0.0
relaxed test	1.160e−2	5.08e−7	0.00	1.26e−7	65
penalization test	1.264e−2	5.54e−7	2.84e−10	3.91e−7	54
penalization test, $l_{\text{TV}} = 100+$	1.316e−2	4.45e−7	0.00	1.49e−7	29

Table 8.11: Comparison of the cost and constraints for the splitting tests from a zigzag with 6 lines (aluminium)



(a) Initialization for classic path optimization (or topology optimization)

(b) Optimal path from classic path optimization

(c) Resulting path from "cut, $F_{\text{top}} = 15$, $TV \leq 14$" (432)

(d) Resulting path from "cut, $F_{\text{top}} = 15$, $TV \leq 4$" (401)

(e) Resulting path from "add-remove, $F_{\text{top}} = 15$, $TV \leq 14$" (415)

(f) Resulting path from "add-remove, $F_{\text{top}} = 15$, $TV \leq 4$" (422)

(g) Initialization for coupled power and path optimization ($\zeta^0 = 0.5$)

(h) "Relaxed test", $l_{\text{TV}} = 0$ (423)

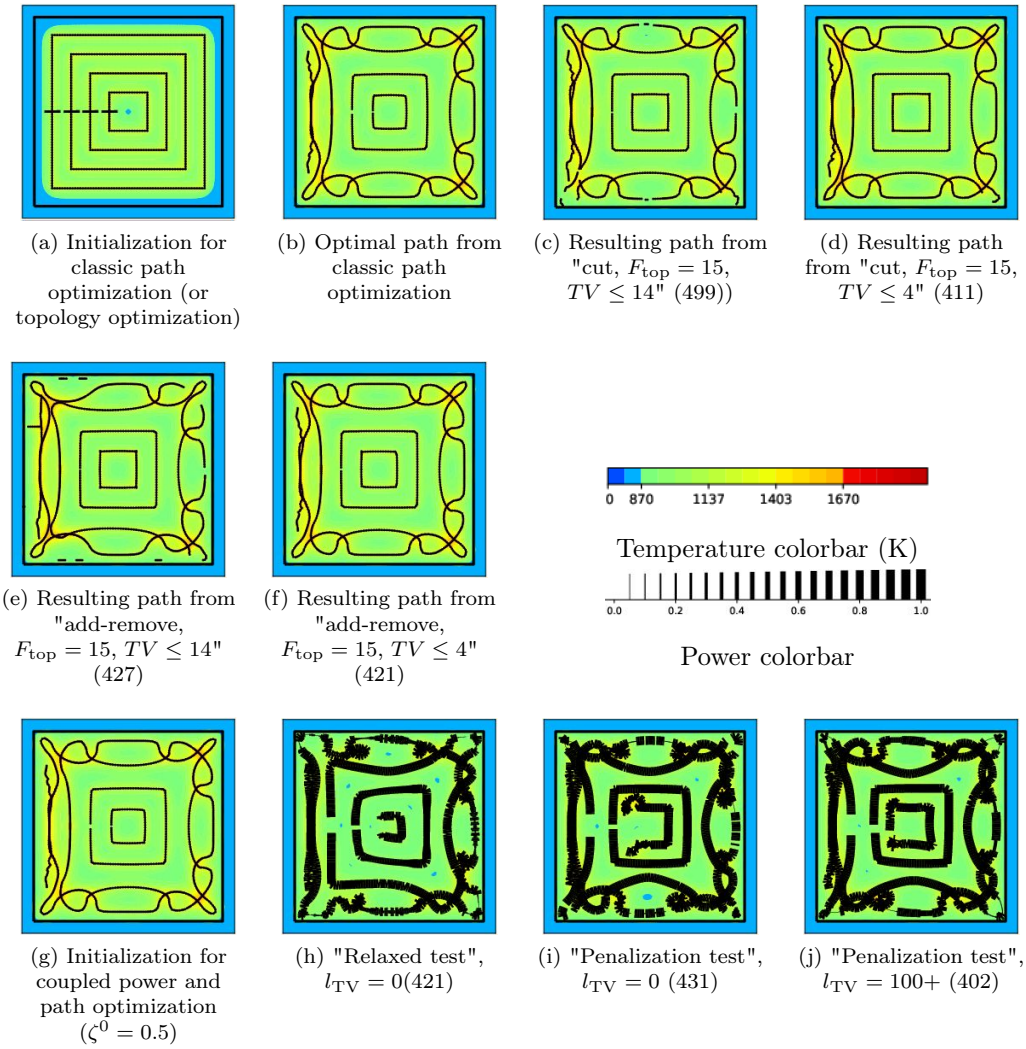
(i) "Penalization test", $l_{\text{TV}} = 0$ (412)

(j) "Penalization test", $l_{\text{TV}} = 100+$ (499)

Figure 8.30: Path and temperature results for the splitting tests starting from a spiral (aluminium)

Optimization case	L_{F} (m)	$\overline{C}_{\phi}$	$\overline{C}_{\text{M,D}_\text{S}}$	$\overline{C}_{\text{M,D}\setminus\text{D}_\text{S}}$	TV
initialization (classic)	5.635e−3	4.60e−3	0.00	0.00	0.0
path optimization	1.060e−2	1.37e−6	0.00	1.61e−5	0.0
cut, $F_{\text{top}} = 15$, $TV \leq 14$	1.106e−2	1.61e−6	0.00	6.72e−6	14.0
cut, $F_{\text{top}} = 15$, $TV \leq 4$	1.084e−2	1.25e−6	0.00	1.56e−5	4.0
add-remove, $F_{\text{top}} = 15$, $TV \leq 14$	1.033e−2	6.03e−7	0.00	1.21e−6	14.0
add-remove, $F_{\text{top}} = 15$, $TV \leq 4$	1.081e−2	1.07e−6	0.00	1.24e−5	4.0
initialization ($\zeta = 0.5$)	5.635e−3	5.45e−3	0.00	0.00	0.0
relaxed test	1.209e−2	4.19e−7	0.00	1.35e−7	53
penalization test	1.244e−2	2.78e−7	0.00	2.27e−7	110
penalization test, $l_{\text{TV}} = 100+$	1.256e−2	2.76e−7	0.00	2.62e−7	43

Table 8.12: Comparison of the cost and constraints for the splitting tests from a spiral (aluminium)



(a) Initialization for classic path optimization (or topology optimization)

(b) Optimal path from classic path optimization

(c) Resulting path from "cut, $F_{\text{top}} = 15$, $TV \leq 14$" (499))

(d) Resulting path from "cut, $F_{\text{top}} = 15$, $TV \leq 4$" (411)

(e) Resulting path from "add-remove, $F_{\text{top}} = 15$, $TV \leq 14$" (427)

(f) Resulting path from "add-remove, $F_{\text{top}} = 15$, $TV \leq 4$" (421)

(g) Initialization for coupled power and path optimization ($\zeta^0 = 0.5$)

(h) "Relaxed test", $l_{\text{TV}} = 0$(421)

(i) "Penalization test", $l_{\text{TV}} = 0$ (431)

(j) "Penalization test", $l_{\text{TV}} = 100+$ (402)

Figure 8.31: Path and temperature results for the splitting tests starting from a contour with 4 lines (aluminium)

Optimization case	L_{F} (m)	$\overline{C}_{\phi}$	$\overline{C}_{\text{M,D}_\text{S}}$	$\overline{C}_{\text{M,D}\setminus\text{D}_\text{S}}$	TV
initialization (classic)	1.008e−2	4.25e−4	0.00	0.00	3.0
path optimization	1.334e−2	2.20e−7	0.00	2.80e−7	0.0
cut, $F_{\text{top}} = 15$, $TV = 14$	1.333e−2	1.03e−7	0.00	1.74e−7	13.0
cut, $F_{\text{top}} = 15$, $TV = 4$	1.381e−2	1.86e−7	0.00	2.20e−7	4.0
add-remove, $F_{\text{top}} = 15$, $TV = 14$	1.338e−2	2.02e−7	0.00	2.84e−7	13.0
add-remove, $F_{\text{top}} = 15$, $TV = 4$	1.359e−2	2.54e−7	0.00	3.64e−7	3.0
initialization ($\zeta = 0.5$)	1.008e−2	1.10e−3	0.00	0.00	0.0
relaxed test	1.395e−2	3.67e−8	0.00	1.57e−8	108
penalization test	1.550e−2	7.31e−8	0.00	5.68e−8	96
penalization test, $l_{\text{TV}} = 100+$	1.543e−2	8.51e−9	0.00	6.74e−8	47

Table 8.13: Comparison of the cost and constraints for the splitting tests from a contour with 4 lines (aluminium)

In all these cases, the first observation is that, compared to the initializations, the length is increased but the constraints are globally decreased. The constraints results are better for the power based algorithm. On the other hand, the intermediate power variables and the number of jumps are only penalized while they are imposed by the topology optimization algorithm. It thus seems normal to get better fulfilled constraints in the power based settings since the freedom in the optimization is larger.

Despite the different final values, the global shape of the path remains the same. In Figure 8.29, the cuts appear on the bottom and top lines (same behavior than for the zigzag with 9 lines). In Figure 8.30, the cuts mostly appear on the square sides. Note that in both the topology optimization and power based algorithms, the path intersections are not removed. However, it intuitively seems that they should since in industrial applications, these intersections are prohibited. Further investigating these issues is part of the perspectives. Finally, the results are not that clear in Figure 8.31. The topology optimization codes seem to add connected components on the sides of the square. However, the initialization already starting with 4 connected components (and thus $TV = 3$), both tests limiting the number of connected components to 5 do not provide much information. As for the power based algorithm, it confirms the interest of a high number of connected components to adjust the constraint on the object boundary mostly.

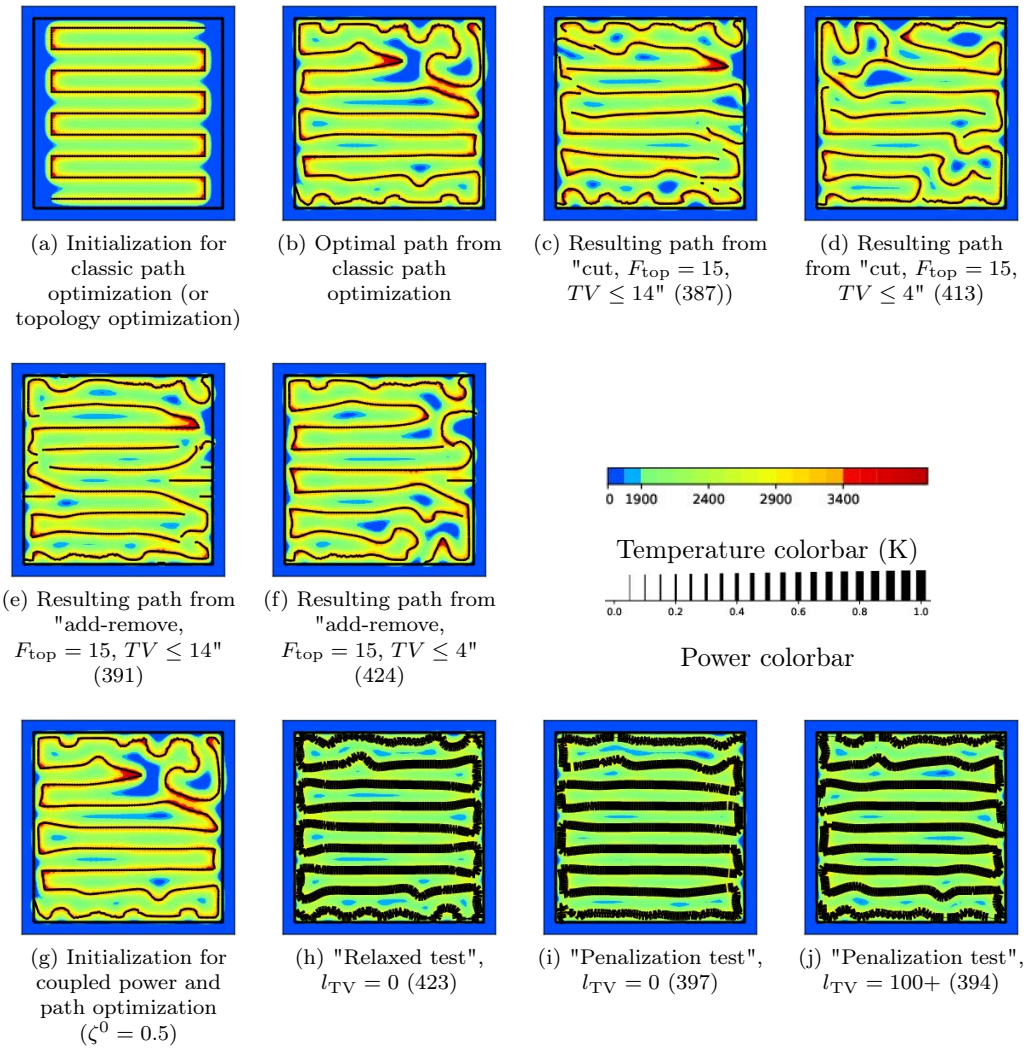
To further compare the results, the number of connected components should not be restricted. However, the computational time of this algorithm dealing with more than 20 connected components is high and cannot be reasonably considered.

8.5.2 Square geometry, titanium context

In the titanium context, three initializations are considered: the zigzag with 9 lines (results in Figure 8.32 and Table 8.14), the spiral (Figure 8.33 and Table 8.15) and the contour with 4 lines (Figure 8.34 and Table 8.16). For the coupled power and shape optimization, the power value ζ is initialized to 0.5 in each case.

Optimization case	L_{F} (m)	$\overline{C}_{\phi}$	$\overline{C}_{\mathrm{M,D_S}}$	$\overline{C}_{\mathrm{M,D\backslash D_S}}$	TV
initialization (classic)	1.019e−2	1.96e−2	1.96e−5	0.00	0.0
path optimization	1.232e−2	8.99e−4	1.19e−4	4.35e−4	0.0
cut, $F_{\mathrm{top}} = 15$, $TV \leq 14$	1.241e−2	8.50e−4	9.49e−5	4.84e−4	14.0
cut, $F_{\mathrm{top}} = 15$, $TV \leq 4$	1.230e−2	8.44e−4	1.92e−5	2.52e−4	4.0
add-remove, $F_{\mathrm{top}} = 15$, $TV \leq 14$	1.224e−2	4.69e−4	3.27e−5	2.22e−4	13.0
add-remove, $F_{\mathrm{top}} = 15$, $TV \leq 4$	1.226e−2	9.16e−4	2.93e−5	3.93e−4	3.0
initialization ($\zeta = 0.5$)	1.019e−2	5.88e−2	0.00	0.00	0.0
relaxed test	1.284e−2	5.43e−5	1.37e−6	1.10e−5	39
penalization test	1.245e−2	1.27e−4	5.65e−6	1.66e−5	37
penalization test, $l_{\mathrm{TV}} = 100+$	1.274e−2	4.94e−5	2.79e−6	7.67e−6	24

Table 8.14: Comparison of the cost and constraints for the splitting tests from a zigzag with 9 lines (titanium)



(a) Initialization for classic path optimization (or topology optimization)

(b) Optimal path from classic path optimization

(c) Resulting path from "cut, $F_{\text{top}} = 15$, $TV \leq 14$" (387))

(d) Resulting path from "cut, $F_{\text{top}} = 15$, $TV \leq 4$" (413)

(e) Resulting path from "add-remove, $F_{\text{top}} = 15$, $TV \leq 14$" (391)

(f) Resulting path from "add-remove, $F_{\text{top}} = 15$, $TV \leq 4$" (424)

(g) Initialization for coupled power and path optimization ($\zeta^0 = 0.5$)

(h) "Relaxed test", $l_{\text{TV}} = 0$ (423)

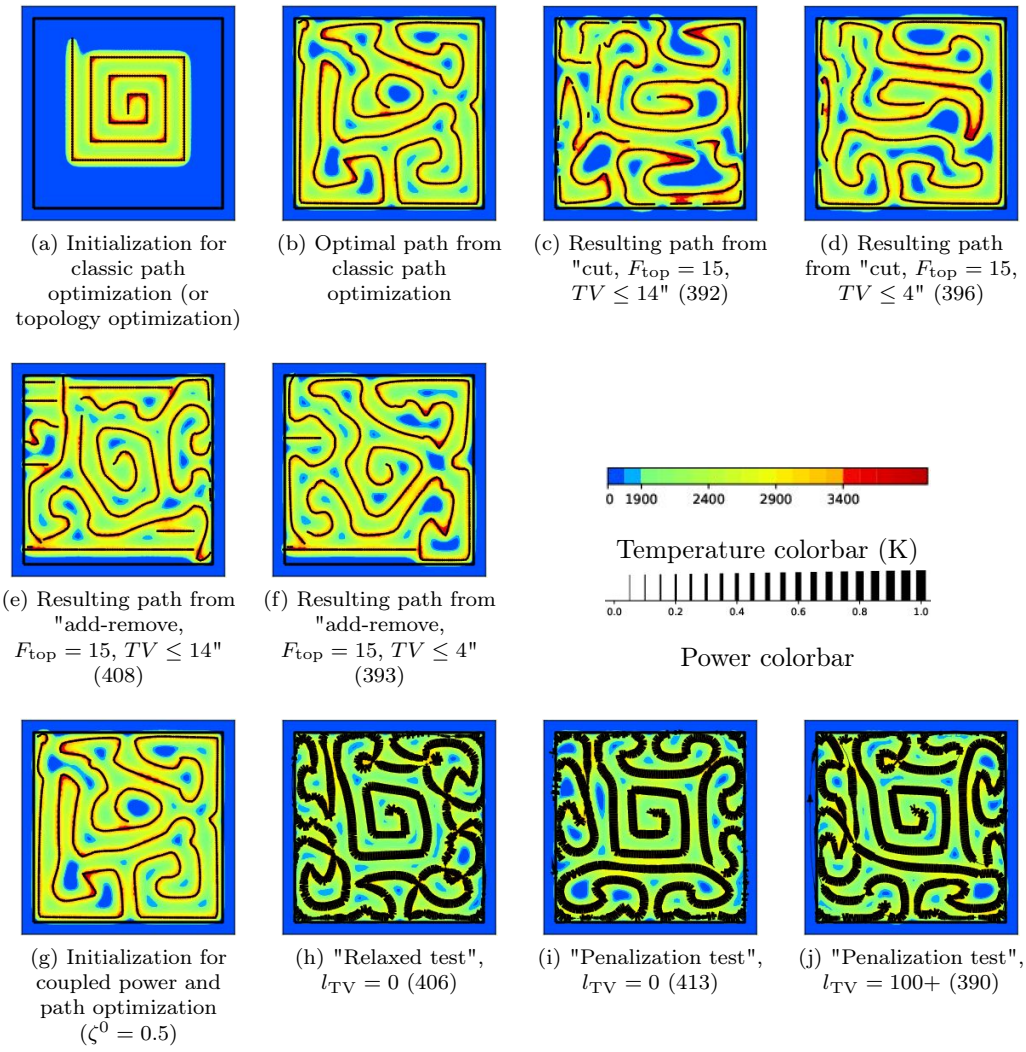
(i) "Penalization test", $l_{\text{TV}} = 0$ (397)

(j) "Penalization test", $l_{\text{TV}} = 100+$ (394)

Figure 8.32: Path and temperature results for the splitting tests starting from a zigzag with 9 lines (titanium)

Optimization case	L_{F} (m)	$\overline{C}_{\phi}$	$\overline{C}_{\text{M,D}_\text{S}}$	$\overline{C}_{\text{M,D}\setminus\text{D}_\text{S}}$	TV
initialization (classic)	5.635e−3	1.50e−1	4.71e−5	0.00	0.0
path optimization	1.286e−2	5.40e−4	4.21e−5	4.97e−4	0.0
cut, $F_{\text{top}} = 15$, $TV \leq 14$	1.248e−2	2.41e−3	2.77e−4	9.53e−4	14.0
cut, $F_{\text{top}} = 15$, $TV \leq 4$	1.264e−2	1.34e−3	7.95e−5	1.77e−3	4.0
add-remove, $F_{\text{top}} = 15$, $TV \leq 14$	1.210e−2	2.62e−3	3.18e−5	1.14e−4	13.0
add-remove, $F_{\text{top}} = 15$, $TV \leq 4$	1.230e−2	2.01e−3	3.51e−5	4.70e−4	4.0
initialization ($\zeta = 0.5$)	5.635e−3	1.79e−1	0.00	0.00	0.0
relaxed test	1.644e−2	8.40e−5	2.87e−6	2.44e−5	106
penalization test	1.622e−2	3.51e−4	7.90e−6	2.17e−4	93
penalization test, $l_{\text{TV}} = 100+$	1.729e−2	1.90e−4	5.24e−6	9.63e−5	72

Table 8.15: Comparison of the cost and constraints for the splitting tests from a spiral (titanium)



(a) Initialization for classic path optimization (or topology optimization)

(b) Optimal path from classic path optimization

(c) Resulting path from "cut, $F_{\text{top}} = 15$, $TV \leq 14$" (392)

(d) Resulting path from "cut, $F_{\text{top}} = 15$, $TV \leq 4$" (396)

(e) Resulting path from "add-remove, $F_{\text{top}} = 15$, $TV \leq 14$" (408)

(f) Resulting path from "add-remove, $F_{\text{top}} = 15$, $TV \leq 4$" (393)

(g) Initialization for coupled power and path optimization ($\zeta^0 = 0.5$)

(h) "Relaxed test", $l_{\text{TV}} = 0$ (406)

(i) "Penalization test", $l_{\text{TV}} = 0$ (413)

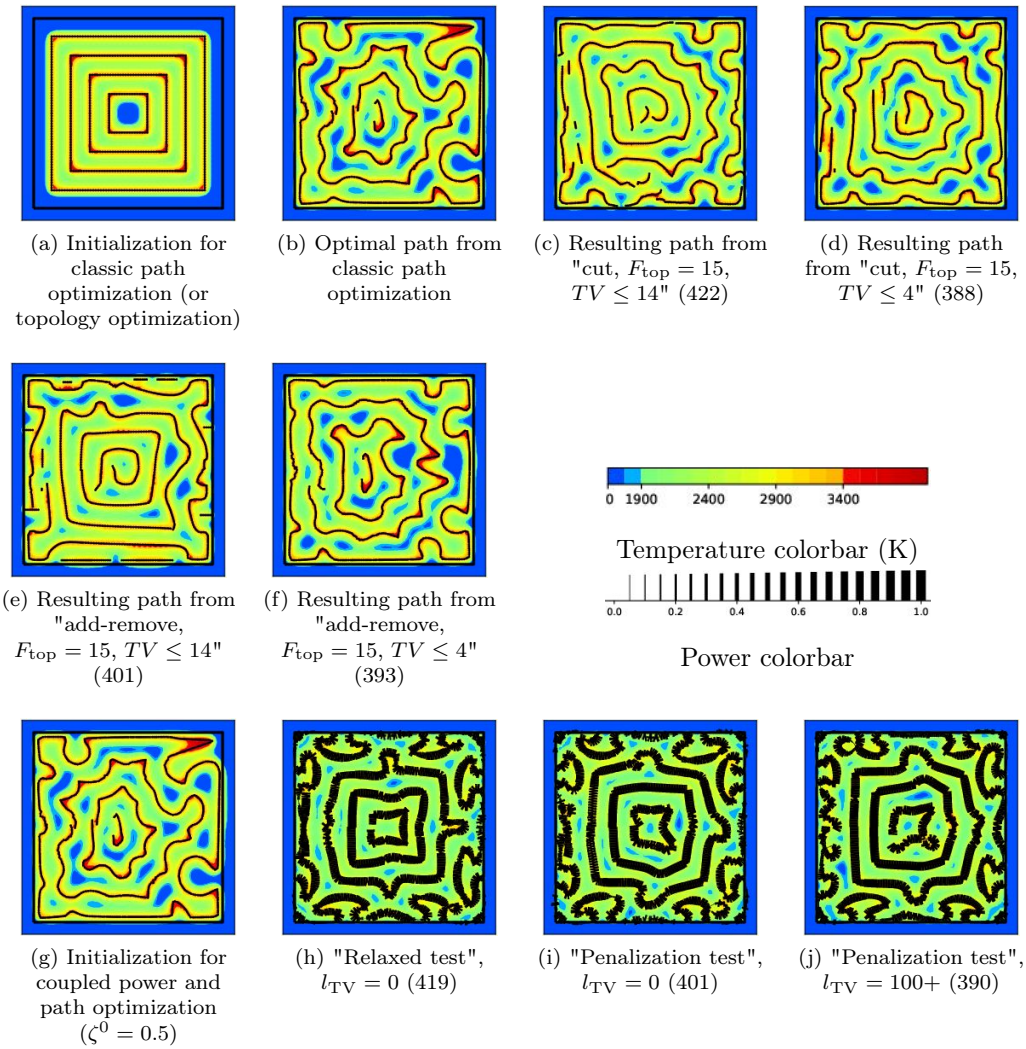
(j) "Penalization test", $l_{\text{TV}} = 100+$ (390)

Figure 8.33: Path and temperature results for the splitting tests starting from a spiral (titanium)

Optimization case	L_{F} (m)	$\overline{C}_{\phi}$	$\overline{C}_{\text{M,D}_\text{S}}$	$\overline{C}_{\text{M,D}\setminus\text{D}_\text{S}}$	TV
initialization (classic)	1.008e−2	2.74e−2	2.43e−5	0.00	3.0
path optimization	1.234e−2	1.01e−3	2.34e−4	1.39e−3	0.0
cut, $F_{\text{top}} = 15$, $TV \leq 14$	1.242e−2	2.34e−4	1.59e−5	1.26e−4	14.0
cut, $F_{\text{top}} = 15$, $TV \leq 4$	1.238e−2	4.12e−4	2.19e−5	2.72e−4	4.0
add-remove, $F_{\text{top}} = 15$, $TV \leq 14$	1.274e−2	3.34e−4	1.35e−5	6.67e−4	14.0
add-remove, $F_{\text{top}} = 15$, $TV \leq 4$	1.230e−2	1.02e−3	5.82e−5	5.02e−4	3.0
initialization ($\zeta = 0.5$)	1.008e−2	6.40e−2	0.00	0.00	0.0
relaxed test	1.640e−2	2.18e−5	5.71e−7	1.05e−5	142
penalization test	1.641e−2	3.27e−5	1.63e−6	2.63e−5	170
penalization test, $l_{\text{TV}} = 100+$	1.573e−2	5.33e−5	2.87e−5	5.40e−5	83

Table 8.16: Comparison of the cost and constraints for the splitting tests from a contour with 4 lines (titanium)

The results in the titanium context are not as clear. Indeed, once again, the low conductivity leads to



(a) Initialization for classic path optimization (or topology optimization)

(b) Optimal path from classic path optimization

(c) Resulting path from "cut, $F_{\text{top}} = 15$, $TV \leq 14$" (422)

(d) Resulting path from "cut, $F_{\text{top}} = 15$, $TV \leq 4$" (388)

(e) Resulting path from "add-remove, $F_{\text{top}} = 15$, $TV \leq 14$" (401)

(f) Resulting path from "add-remove, $F_{\text{top}} = 15$, $TV \leq 4$" (393)

(g) Initialization for coupled power and path optimization ($\zeta^0 = 0.5$)

(h) "Relaxed test", $l_{\text{TV}} = 0$ (419)

(i) "Penalization test", $l_{\text{TV}} = 0$ (401)

(j) "Penalization test", $l_{\text{TV}} = 100+$ (390)

Figure 8.34: Path and temperature results for the splitting tests starting from a contour with 4 lines (titanium)

more intricate results. Yet, we observe once again how interesting it is to allow more than one connected component: in all the cases considered, the constraints have been decreased by allowing splits in the path. It also appears that most of these additional parts of the path are small (see especially Figures 8.32 (c) and (e), 8.33 (c) and (e), 8.34(c) and (e)). They adapt to the zones in which the phase constraint is not satisfied. Finally, if this is not as clear than in the aluminium case (since the path optimization only does not lead to as good results), it also seems that the path pieces appear mainly on the sides of the square. Shared by both physical test case, this feature leads to interesting perspectives: in industrial applications, the contour of the geometry is usually scanned apart. Including this process in the simulation would it decrease the number of connected components? Which process is the most accurate? Which allows the largest residual stresses decrease?

8.5.3 Comparison between the coupled power and path optimization and the topology optimization methods for complex geometries

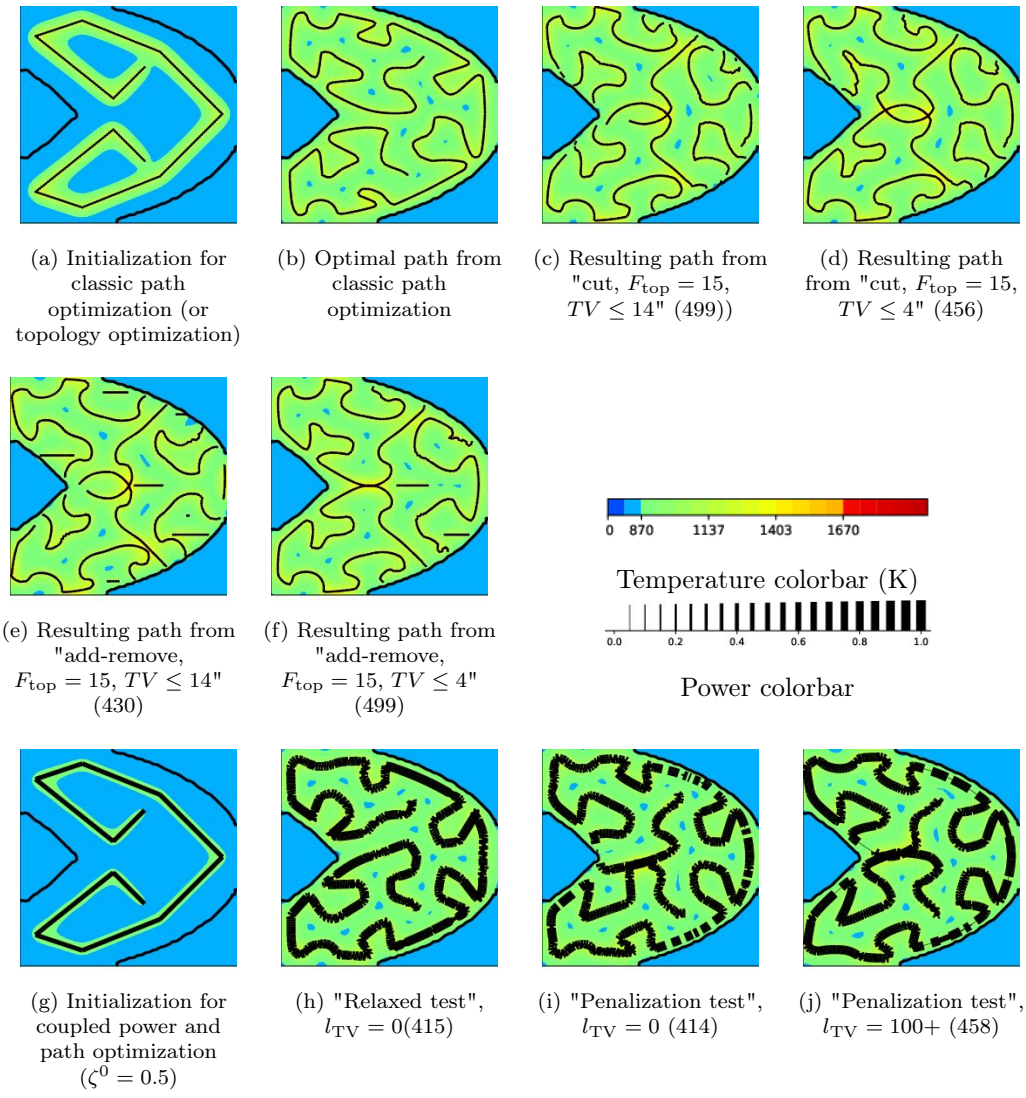
To end this comparison, the zero hole, one hole and three holes objects are now chosen as the parts to build, with only one path initialization considered. The results are summed in six Figures and Tables:

- Figure 8.35 and Table 8.35 for the zero hole object made in aluminium,
- Figure 8.36 and Table 8.36 for the one hole object made in aluminium,
- Figure 8.37 and Table 8.37 for the three holes object made in aluminium,

- Figure 8.38 and Table 8.38 for the zero hole object made in titanium,

- Figure 8.39 and Table 8.39 for the one hole object made in titanium,

- Figure 8.40 and Table 8.40 for the three holes object made in titanium.



(a) Initialization for classic path optimization (or topology optimization)

(b) Optimal path from classic path optimization

(c) Resulting path from "cut, $F_{\text{top}} = 15$, $TV \leq 14$" (499))

(d) Resulting path from "cut, $F_{\text{top}} = 15$, $TV \leq 4$" (456)

(e) Resulting path from "add-remove, $F_{\text{top}} = 15$, $TV \leq 14$" (430)

(f) Resulting path from "add-remove, $F_{\text{top}} = 15$, $TV \leq 4$" (499)

(g) Initialization for coupled power and path optimization ($\zeta^0 = 0.5$)

(h) "Relaxed test", $l_{\text{TV}} = 0$(415)

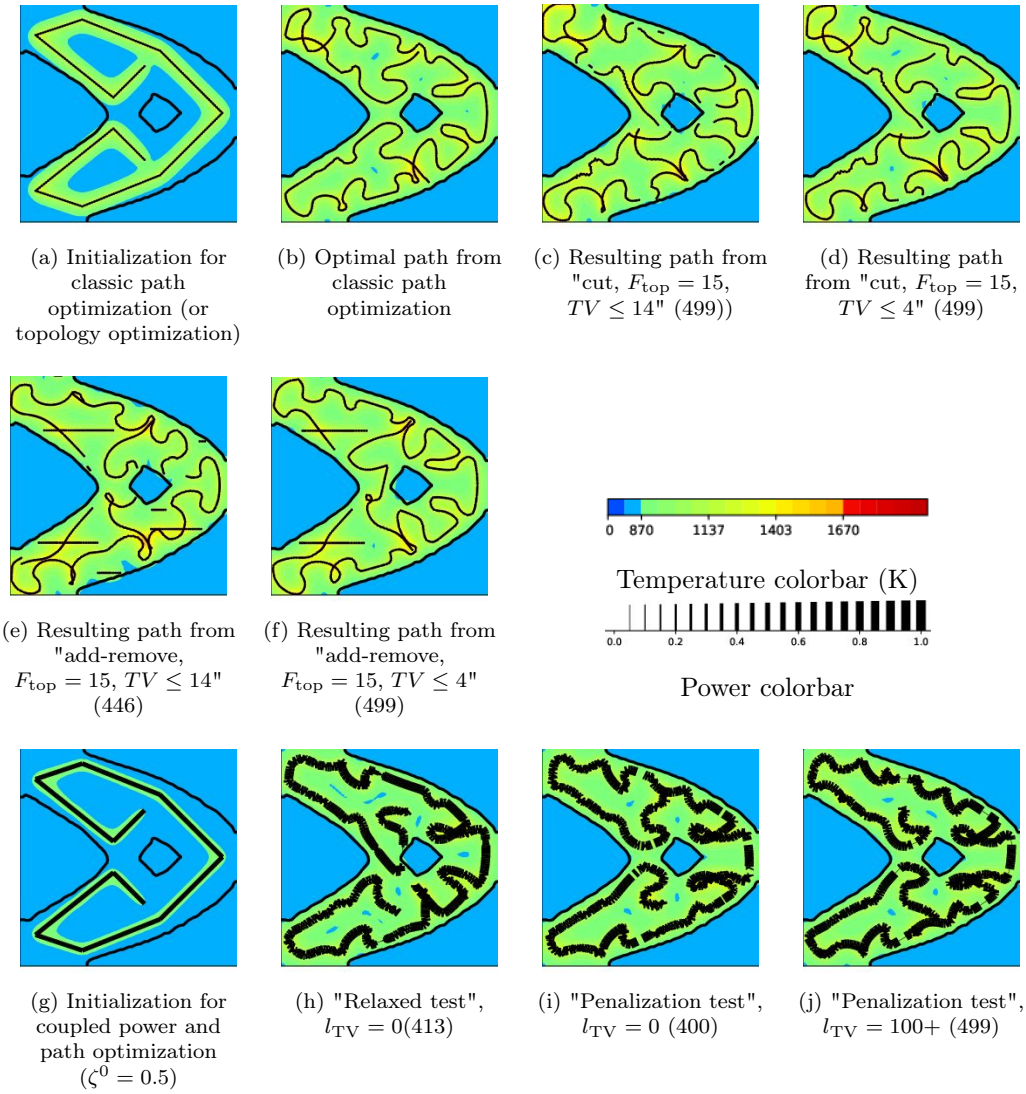
(i) "Penalization test", $l_{\text{TV}} = 0$ (414)

(j) "Penalization test", $l_{\text{TV}} = 100+$ (458)

Figure 8.35: Path and temperature results for the splitting tests for the zero hole object (aluminium)

Optimization case	L_F (m)	$\overline{C}_\phi$	$\overline{C}_{M,D_S}$	$\overline{C}_{M,D\setminus D_S}$	TV
initialization (classic)	4.687e−3	2.95e−3	0.00	0.00	0.0
path optimization	9.961e−3	3.69e−7	0.00	1.81e−7	0.0
cut, $F_{top} = 15$, $TV \leq 14$	1.029e−2	3.64e−7	0.00	1.69e−7	14.0
cut, $F_{top} = 15$, $TV \leq 4$	1.085e−2	6.00e−7	0.00	6.41e−7	4.0
add-remove, $F_{top} = 15$, $TV \leq 14$	1.056e−2	1.24e−6	0.00	1.14e−6	14.0
add-remove, $F_{top} = 15$, $TV \leq 4$	1.033e−2	3.00e−7	0.00	1.93e−7	4.0
initialization ($\zeta = 0.5$)	4.687e−3	4.85e−3	0.00	0.00	0.0
relaxed test	1.028e−2	8.37e−7	0.00	1.56e−7	25
penalization test	1.069e−2	7.16e−7	0.00	1.76e−7	64
penalization test, $l_{TV} = 100+$	1.094e−2	7.19e−7	0.00	2.22e−7	29

Table 8.17: Comparison of the cost and constraints for the splitting tests for the zero hole object (aluminium)



(a) Initialization for classic path optimization (or topology optimization)

(b) Optimal path from classic path optimization

(c) Resulting path from "cut, $F_{top} = 15$, $TV \leq 14$" (499))

(d) Resulting path from "cut, $F_{top} = 15$, $TV \leq 4$" (499)

(e) Resulting path from "add-remove, $F_{top} = 15$, $TV \leq 14$" (446)

(f) Resulting path from "add-remove, $F_{top} = 15$, $TV \leq 4$" (499)

(g) Initialization for coupled power and path optimization ($\zeta^0 = 0.5$)

(h) "Relaxed test", $l_{TV} = 0$(413)

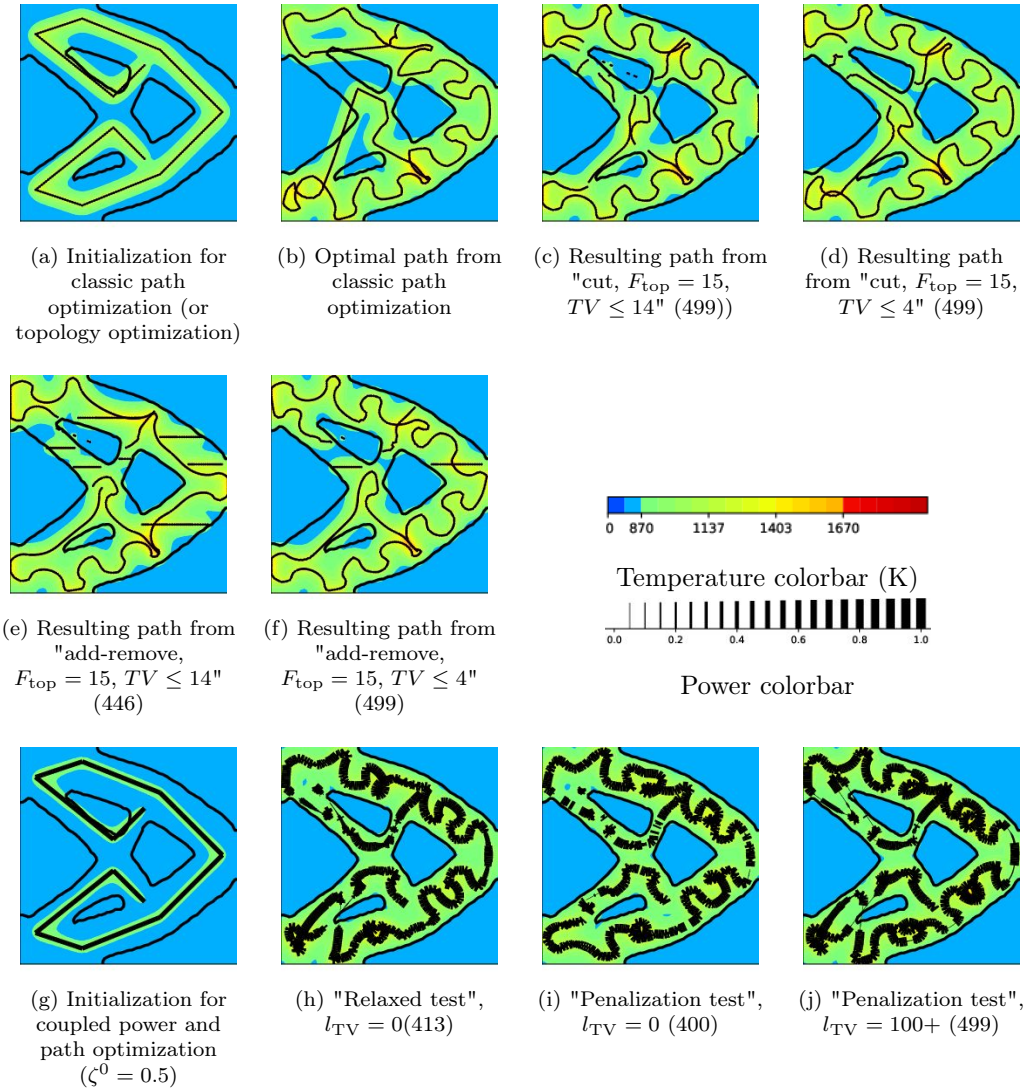
(i) "Penalization test", $l_{TV} = 0$ (400)

(j) "Penalization test", $l_{TV} = 100+$ (499)

Figure 8.36: Path and temperature results for the splitting tests for the one hole object (aluminium)

Optimization case	L_F (m)	$\overline{C}_\phi$	$\overline{C}_{M,D_S}$	$\overline{C}_{M,D\setminus D_S}$	TV
initialization (classic)	4.687e−3	1.74e−3	0.00	6.33e−7	0.0
path optimization	9.335e−3	3.64e−7	0.00	3.95e−7	0.0
cut, $F_{top} = 15$, $TV \leq 14$	9.617e−3	9.83e−7	0.00	6.69e−7	14.0
cut, $F_{top} = 15$, $TV \leq 4$	9.614e−3	3.55e−7	0.00	3.11e−7	4.0
add-remove, $F_{top} = 15$, $TV \leq 14$	9.888e−3	4.22e−6	0.00	5.07e−6	13.0
add-remove, $F_{top} = 15$, $TV \leq 4$	9.516e−3	6.49e−7	0.00	9.87e−7	3.0
initialization ($\zeta = 0.5$)	4.687e−3	3.53e−3	0.00	0.00	0.0
relaxed test	8.894e−3	8.54e−7	0.00	1.25e−7	47
penalization test	9.553e−3	5.50e−7	0.00	1.22e−7	42
penalization test, $l_{TV} = 100+$	9.852e−3	6.23e−7	0.00	1.83e−7	43

Table 8.18: Comparison of the cost and constraints for the splitting tests for the one hole object (aluminium)



(a) Initialization for classic path optimization (or topology optimization)

(b) Optimal path from classic path optimization

(c) Resulting path from "cut, $F_{top} = 15$, $TV \leq 14$" (499))

(d) Resulting path from "cut, $F_{top} = 15$, $TV \leq 4$" (499)

(e) Resulting path from "add-remove, $F_{top} = 15$, $TV \leq 14$" (446)

(f) Resulting path from "add-remove, $F_{top} = 15$, $TV \leq 4$" (499)

(g) Initialization for coupled power and path optimization ($\zeta^0 = 0.5$)

(h) "Relaxed test", $l_{TV} = 0$(413)

(i) "Penalization test", $l_{TV} = 0$ (400)

(j) "Penalization test", $l_{TV} = 100+$ (499)

Figure 8.37: Path and temperature results for the splitting tests for the three holes object (aluminium)

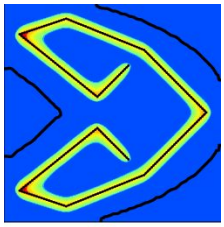
Optimization case	L_{F} (m)	$\overline{C}_{\phi}$	$\overline{C}_{\mathrm{M,D_S}}$	$\overline{C}_{\mathrm{M,D\backslash D_S}}$	TV
initialization (classic)	4.687e−3	1.45e−3	0.00	7.03e−4	0.0
path optimization	8.880e−3	1.80e−4	0.00	1.24e−3	0.0
cut, $F_{\mathrm{top}} = 15$, $TV \leq 14$	8.902e−3	1.19e−4	4.69e−9	1.31e−4	14.0
cut, $F_{\mathrm{top}} = 15$, $TV \leq 4$	9.268e−3	1.05e−4	4.15e−12	2.19e−4	4.0
add-remove, $F_{\mathrm{top}} = 15$, $TV \leq 14$	8.745e−3	1.32e−4	8.98e−9	5.80e−5	13.0
add-remove, $F_{\mathrm{top}} = 15$, $TV \leq 4$	9.019e−3	1.10e−4	1.96e−9	1.06e−4	4.0
initialization ($\zeta = 0.5$)	4.687e−3	3.31e−3	0.00	3.74e−5	0.0
relaxed test	1.039e−2	8.90e−7	0.00	3.62e−7	78
penalization test	1.030e−2	7.58e−7	0.00	2.80e−7	74
penalization test, $l_{\mathrm{TV}} = 100+$	1.158e−2	7.10e−7	0.00	4.57e−7	58

Table 8.19: Comparison of the cost and constraints for the splitting tests for the three holes object (aluminium)

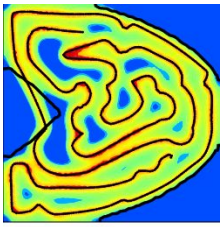
The first geometry (Figure 8.35 and Table 8.35) does not include any hole and is thus the easiest to scan. The simple path optimization shown in Figure 8.35 (b) is already well fulfilling the temperature constraints. Increasing the number of connected components does not drastically improve the results. Actually, but for Figure 8.35(e), the final constraints are very close to the classic optimization results (Table 8.17). This confirms once again the concept of optimal scanning energy. To really compare both techniques, for the aluminium, the geometry must be more complex. The observations related to the zero hole object still apply for the one hole part (Figure 8.36 and Table 8.36). Indeed, once again, the classic path optimization leads to very good results, that are not improved by allowing more than one connected component in the path. The third object is far more interesting (Figure 8.37 and Table 8.19). Indeed, because of the complex geometry, the classic path optimization could not fulfill the phase and maximal temperature out of the domain constraints (Figure 8.37(b)). All the tests result in better final paths. The power optimization method decreases most of the constraints. Yet, even with a jump control, the final total variation is high. Note that the tests resulting from topology optimization create path pieces in the holes and further investigations should be conducted to determine whether this is an optimization artifact or not. If yes, this raises the perspective of better preventing these issues: decreasing the maximum temperature out of the geometry or working on a projection on the object are part of them.

Optimization case	L_{F} (m)	$\overline{C}_{\phi}$	$\overline{C}_{\mathrm{M,D_S}}$	$\overline{C}_{\mathrm{M,D\backslash D_S}}$	TV
initialization (classic)	4.687e−3	1.28e−1	7.79e−6	0.00	0.0
path optimization	1.217e−2	2.21e−3	8.89e−5	1.55e−2	0.0
cut, $F_{\mathrm{top}} = 15$, $TV \leq 14$	1.195e−2	5.24e−4	2.80e−5	3.03e−4	14.0
cut, $F_{\mathrm{top}} = 15$, $TV \leq 4$	1.204e−2	7.35e−4	4.92e−5	2.02e−4	4.0
add-remove, $F_{\mathrm{top}} = 15$, $TV \leq 14$	1.172e−2	8.65e−4	3.47e−5	2.19e−4	14.0
add-remove, $F_{\mathrm{top}} = 15$, $TV \leq 4$	1.195e−2	4.78e−4	2.66e−5	6.44e−4	4.0
initialization ($\zeta = 0.5$)	4.687e−3	1.89e−1	0.00	0.00	0.0
relaxed test	1.464e−2	1.43e−4	4.01e−6	4.61e−5	102
penalization test	1.465e−2	1.94e−4	7.47e−6	8.01e−5	172
penalization test, $l_{\mathrm{TV}} = 100+$	1.510e−2	2.39e−4	1.02e−5	1.97e−4	84

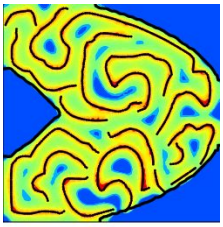
Table 8.20: Comparison of the cost and constraints for the splitting tests for the zero hole object (titanium)



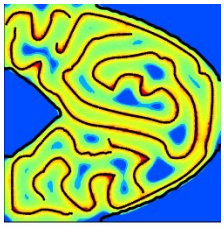
(a) Initialization for classic path optimization (or topology optimization)



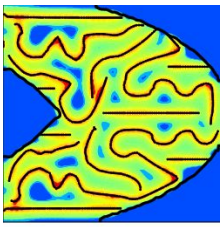
(b) Optimal path from classic path optimization



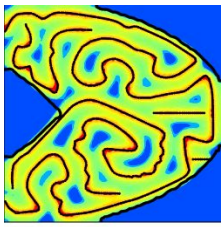
(c) Resulting path from "cut, $F_{\text{top}} = 15$, $TV \leq 14$" (411))



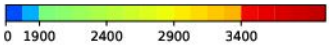
(d) Resulting path from "cut, $F_{\text{top}} = 15$, $TV \leq 4$" (394)



(e) Resulting path from "add-remove, $F_{\text{top}} = 15$, $TV \leq 14$" (394)

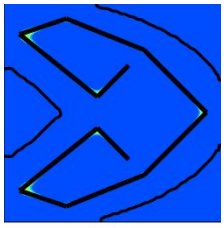


(f) Resulting path from "add-remove, $F_{\text{top}} = 15$, $TV \leq 4$" (394)

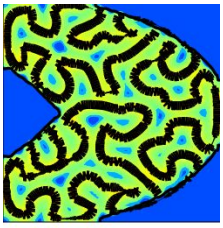


Temperature colorbar (K)

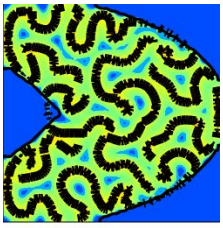
Power colorbar



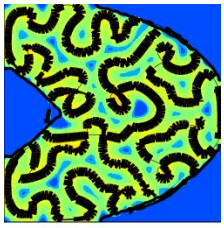
(g) Initialization for coupled power and path optimization ($\zeta^0 = 0.5$)



(h) "Relaxed test", $l_{\text{TV}} = 0$(411)



(i) "Penalization test", $l_{\text{TV}} = 0$ (422)

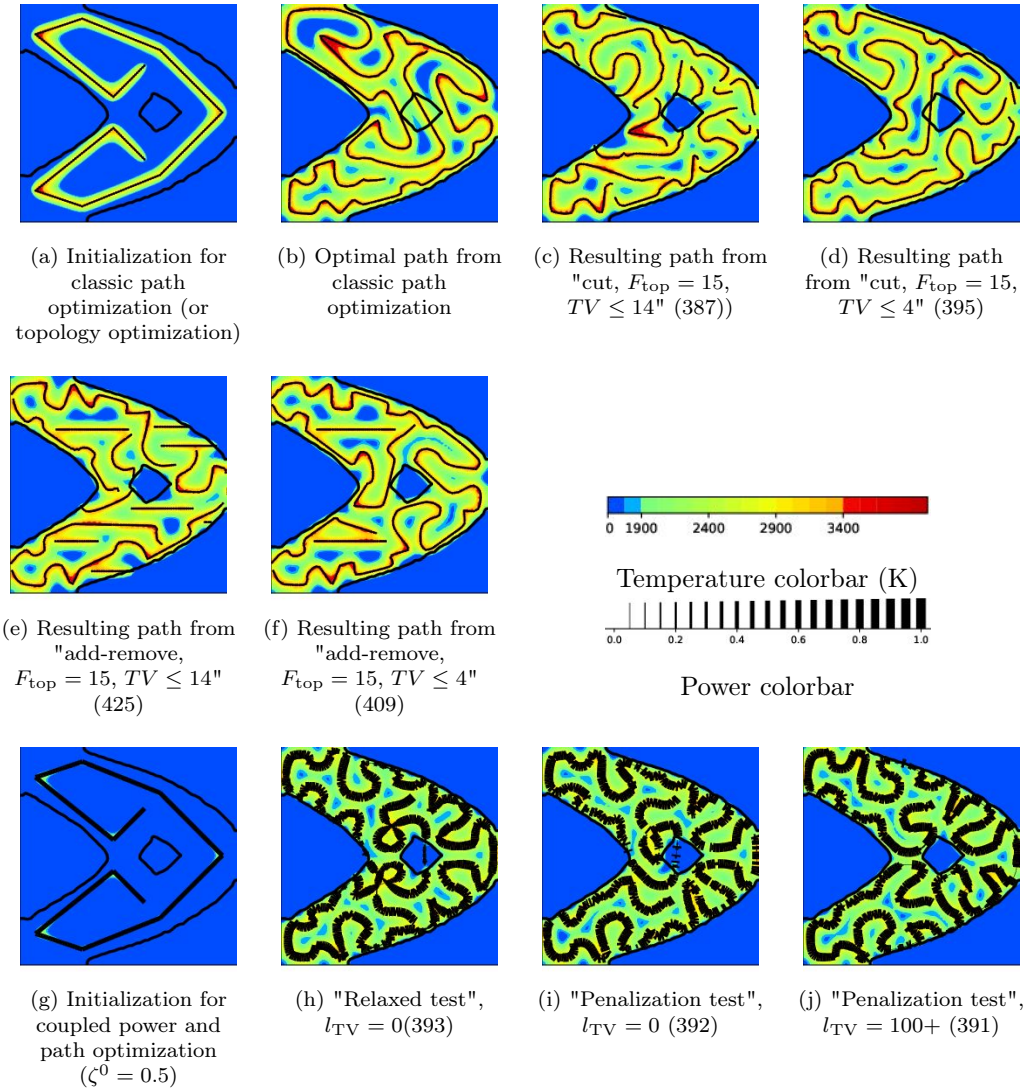


(j) "Penalization test", $l_{\text{TV}} = 100+$ (390)

Figure 8.38: Path and temperature results for the splitting tests for the zero hole object (titanium)

Optimization case	L_{F} (m)	$\overline{C}_{\phi}$	$\overline{C}_{\text{M,D}_\text{S}}$	$\overline{C}_{\text{M,D}\setminus\text{D}_\text{S}}$	TV
initialization (classic)	4.687e−3	9.04e−2	1.00e−5	0.00	0.0
path optimization	9.848e−3	1.51e−3	1.64e−4	7.86e−3	0.0
cut, $F_{\text{top}} = 15$, $TV \leq 14$	9.638e−3	9.98e−4	8.95e−5	2.10e−4	14.0
cut, $F_{\text{top}} = 15$, $TV \leq 4$	9.700e−3	3.01e−4	1.35e−5	2.76e−3	4.0
add-remove, $F_{\text{top}} = 15$, $TV \leq 14$	9.647e−3	1.02e−3	3.80e−5	4.04e−4	14.0
add-remove, $F_{\text{top}} = 15$, $TV \leq 4$	9.668e−3	8.29e−4	3.74e−5	4.79e−4	4.0
initialization ($\zeta = 0.5$)	4.687e−3	1.56e−1	0.00	0.00	0.0
relaxed test	1.168e−2	1.54e−4	4.37e−6	2.85e−5	82
penalization test	1.130e−2	1.78e−4	6.80e−6	3.69e−5	144
penalization test, $l_{\text{TV}} = 100+$	1.110e−2	1.22e−4	1.20e−5	4.93e−5	67

Table 8.21: Comparison of the cost and constraints for the splitting tests for the one hole object (titanium)



(a) Initialization for classic path optimization (or topology optimization)

(b) Optimal path from classic path optimization

(c) Resulting path from "cut, $F_{\text{top}} = 15$, $TV \leq 14$" (387))

(d) Resulting path from "cut, $F_{\text{top}} = 15$, $TV \leq 4$" (395)

(e) Resulting path from "add-remove, $F_{\text{top}} = 15$, $TV \leq 14$" (425)

(f) Resulting path from "add-remove, $F_{\text{top}} = 15$, $TV \leq 4$" (409)

(g) Initialization for coupled power and path optimization ($\zeta^0 = 0.5$)

(h) "Relaxed test", $l_{\text{TV}} = 0$(393)

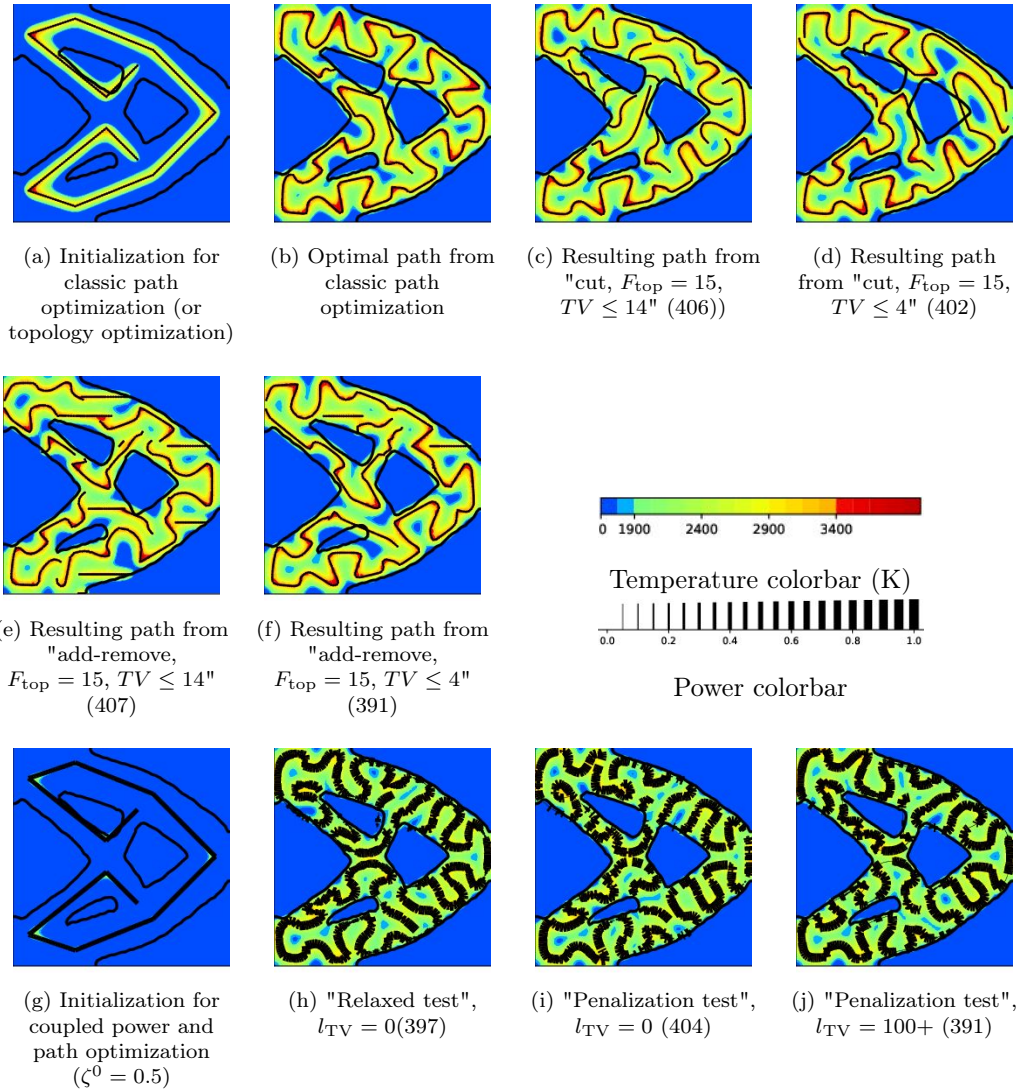
(i) "Penalization test", $l_{\text{TV}} = 0$ (392)

(j) "Penalization test", $l_{\text{TV}} = 100+$ (391)

Figure 8.39: Path and temperature results for the splitting tests for the one hole object (titanium)

Optimization case	L_{F} (m)	$\overline{C}_{\phi}$	$\overline{C}_{\text{M,D}_\text{S}}$	$\overline{C}_{\text{M,D}\setminus\text{D}_\text{S}}$	TV
initialization (classic)	4.687e−3	8.38e−2	1.05e−5	2.79e−3	0.0
path optimization	9.580e−3	1.62e−3	1.15e−4	1.88e−3	0.0
cut, $F_{\text{top}} = 15$, $TV \leq 14$	9.467e−3	6.55e−4	4.65e−5	2.11e−4	14.0
cut, $F_{\text{top}} = 15$, $TV \leq 4$	9.568e−3	1.40e−3	5.66e−5	4.61e−3	4.0
add-remove, $F_{\text{top}} = 15$, $TV \leq 14$	9.405e−3	1.53e−3	5.66e−5	6.31e−4	13.0
add-remove, $F_{\text{top}} = 15$, $TV \leq 4$	9.445e−3	1.67e−3	6.45e−5	2.48e−3	4.0
initialization ($\zeta = 0.5$)	4.687e−3	1.51e−1	0.00	9.09e−6	0.0
relaxed test	1.218e−2	1.66e−4	3.29e−6	6.43e−5	103
penalization test	1.198e−2	2.12e−4	6.01e−6	6.27e−5	167
penalization test, $l_{\text{TV}} = 100+$	1.163e−2	1.48e−4	1.59e−5	5.64e−5	81

Table 8.22: Comparison of the cost and constraints for the splitting tests for the three holes object (titanium)



(a) Initialization for classic path optimization (or topology optimization)

(b) Optimal path from classic path optimization

(c) Resulting path from "cut, $F_{\text{top}} = 15$, $TV \leq 14$" (406))

(d) Resulting path from "cut, $F_{\text{top}} = 15$, $TV \leq 4$" (402)

(e) Resulting path from "add-remove, $F_{\text{top}} = 15$, $TV \leq 14$" (407)

(f) Resulting path from "add-remove, $F_{\text{top}} = 15$, $TV \leq 4$" (391)

(g) Initialization for coupled power and path optimization ($\zeta^0 = 0.5$)

(h) "Relaxed test", $l_{\text{TV}} = 0$(397)

(i) "Penalization test", $l_{\text{TV}} = 0$ (404)

(j) "Penalization test", $l_{\text{TV}} = 100+$ (391)

Figure 8.40: Path and temperature results for the splitting tests for the three holes object (titanium)

All along this work, it has been made clear that path optimization is harder for low conductivity materials. Thus, the classic path optimization for the first "simpler" object leads to unsatisfied constraints (Figures 8.38(b), 8.39(b), 8.40(b)). In each of the considered cases, the new tests lead to improved paths. Once again the temperature constraints are lower using the power method. Yet, the same order of magnitude is shared by the results from topology optimization while the total variation is drastically decreased. Note that, once again, the path does not necessarily remains in the geometry (Figures 8.38(b,i,j), 8.39(b,d,h,i)). Yet, this is not always an issue: in Figure 8.39(h,i) for example, the path crossing the hole does not induce enough heat to melt the powder. If a filter was applied after the power based optimization, this source on this part of the path would probably be off. Finally, in Figure 8.40, the interest of allowing small path pieces is clear. Indeed, the geometry is intricate and cutting the source gives more flexibility in the scanning design. This is made especially clear comparing the results from power optimization and from topology optimization. Indeed, a higher total variation allows for lower temperature constraints.

8.6 CONCLUSION

In path optimization in the steady state context (Chapter 6), some tests allowing cuts in the path have led to interesting results, calling for further studying this possibility. This chapter aims at answering the

question.

To do so, two different methods have been proposed: a coupled optimization of the path and the source power and a topology optimization of the path. From the results tested in both cases, it seems quite clear that adding this "cutting" feature to the path design improves the fulfillment of the temperature constraints. Especially for low conductivity material, it reduces the maximum temperature and better fit the geometry to scan. It even appears in most of the results that the higher the number of connected components of the path, the better is the path.

However, to be efficiently used in industry, the number of power jumps must be limited. Indeed, the source cannot be instantaneously switched on and off: each jump increases the scanning time. Each of the developed methods proposes a different control. In power optimization, this control appears as an optimization constraint. It is thus difficult to accurately limit the number of jumps. On the contrary, since the number of jumps is fixed by the user and integrated in the algorithm, this control is a lot easier using topology optimization. However, the computational time is higher: the number of total variation obtained in the power case cannot be reasonably dealt with topology optimization. Indeed, since the topology modification is geometric and impacts the path discretization itself, increasing the number of path connected components drastically increases the computations. Finally, since a path generated with topology optimization is split into different connected components, a post-treatment algorithm must be run to efficiently link each part of the path.

Adapting this feature in the transient context is part of the perspectives. The settings proposed by the coupled path and power optimization can be transferred. Indeed, using the point based discretization for the path, a description of the power variable very similar to the steady case can be introduced. On the contrary, the settings related to topology optimization perfectly suit the steady state case and adapting them to the transient context is a long-term perspective.

CHAPTER 9

SHAPE AND PATH COUPLED OPTIMIZATION IN THE STEADY STATE CONTEXT

Contents

9.1 Introduction . . . 231
9.2 Settings of the optimization problem and differentiation . . . 231
9.2.1 Problem settings . . . 231
9.2.2 Differentiation . . . 232
9.3 Numerical algorithm . . . 233
9.3.1 Path update: inner loop . . . 233
9.3.2 Shape update: outer loop . . . 234
9.4 Numerical results . . . 236
9.4.1 Cantilever test case . . . 237
9.4.2 Large cantilever test case . . . 243
9.5 Conclusion . . . 249

9.1 Introduction

9.2 Settings of the optimization problem and differentiation

9.2.1 Problem settings

In this chapter, we add a shape optimization problem in linearized elasticity to the path optimization problem described in Chapter 6. The new optimization problem thus depends on two variables: the bounded domain $\Omega \subset D$ to scan (previously corresponding to D_S), and the scanning path Γ.

Alike the problem proposed in Section 3.3, we consider the domain $\Omega \subset D$ an open bounded set with Lipschitz boundary $\partial\Omega$. This boundary, with exterior unit normal vector n_Ω is composed of three disjoints parts (Figure 3.2): $\partial\Omega_{\mathrm{D}}$ on which the displacement is imposed (Dirichlet boundary condition), $\partial\Omega_{\mathrm{N}}$ on which a load $g \in L^2(\partial\Omega_{\mathrm{N}})$ is applied, and $\partial\Omega_{\mathrm{F}}$ which is traction-free. Only the boundary $\partial\Omega_{\mathrm{F}}$ is optimizable and the boundary $\partial\Omega_{\mathrm{D}}$ is not reduced to the empty set ($\partial\Omega_{\mathrm{D}} \neq \emptyset$). The solid (and corresponding shape Ω) is comprised of an elastic material with a Hooke's tensor A relating the elastic stress and strain defined following (3.3.5). The elastic displacement $u \in H^1(\Omega, \mathbb{R}^2)$ is solution to the elasticity partial differential equation recalled here

$$\begin{cases} -\mathrm{div}\left(A\epsilon(u)\right) = 0 & \text{in } \Omega \\ A\epsilon(u).n = g & \text{on } \partial\Omega_{\mathrm{N}} \\ A\epsilon(u).n = 0 & \text{on } \partial\Omega_{\mathrm{F}} \\ u = 0 & \text{on } \partial\Omega_{\mathrm{D}}, \end{cases} \tag{9.2.1}$$

where

$$\epsilon(u) = \frac{1}{2}\left(\nabla u + \nabla u^T\right). \tag{9.2.2}$$

Setting $H^1_D(\Omega, \mathbb{R}^2) = \{v \in H^1(\Omega, \mathbb{R}^2),$ such that $v = 0$ on $\partial\Omega_{\mathrm{D}}\}$ the set of functions in $H^1(\Omega, \mathbb{R}^2)$ canceling on $\partial\Omega_{\mathrm{D}}$, the elastic variational problem states that,

$$\int_\Omega A\epsilon(u) : e(\varphi)dx - \int_{\partial\Omega_{\mathrm{N}}} g\varphi ds = 0, \qquad \forall \varphi \in H^1_D(\Omega, \mathbb{R}^2). \tag{9.2.3}$$

Alike in Chapter 6, we consider in the domain D a path $\Gamma \in \mathcal{G}$ with tangent τ_Γ, normal n_Γ and curvature κ_Γ. It carries an energy source inducing a temperature y in the whole domain D. This temperature $y \in H^1(D,\mathbb{R})$ is solution of the classic heat equation

$$\begin{cases} -\nabla\left(\lambda \nabla y\right) + \beta\left(y - y_{\text{ini}}\right) = P\chi_\Gamma & \text{in } D, \\ \lambda \partial_n y = 0 & \text{on } \partial D, \end{cases} \tag{9.2.4}$$

and the variational formulation

$$\int_D \lambda \nabla y \nabla \phi + \beta\left(y - y_{\text{ini}}\right)\phi dx - \int_\Gamma P\phi ds = 0, \qquad \forall \varphi \in H^1(D,\mathbb{R}). \tag{9.2.5}$$

The objective is to minimize the normalized shape compliance and the path length. The objective function $J(\Omega,\Gamma)$ is thus defined by

$$J(\Omega,\Gamma) = \frac{1}{C^0_{\text{ply}}}\int_\Omega A\epsilon(u) : \epsilon(u) dx + \frac{1}{\Gamma^0}\int_\Gamma ds = \frac{C_{\text{ply}}(\Omega)}{C^0_{\text{ply}}} + \frac{L_{\text{F}}}{L^0_{\text{F}}},$$

with $C^0_{\text{ply}} = C_{\text{ply}}(\Omega^0)$ the initial compliance and $L^0_{\text{F}} = L_{\text{F}}(\Gamma^0)$ the initial length. No balancing is chosen between both functions. It is of course part of perspectives to compromise between the compliance and the path length depending on the industrial requirements.
In this first approach, we aim at characterizing the impact of the scanning path consideration on shape optimization. To do so, we choose a comparison with fixed volume: the differences in the designs are only a matter of material redistribution, facilitating the interpretation. The volume constraint applied is thus

$$\frac{1}{V^0}\int_\Omega dx = \frac{V}{V^0} = \frac{V_{tar}}{V^0},$$

with V_{tar} fixed by the user and $V^0 = \int_{\Omega^0} dx$ the initial volume. The temperature constraints introduced in Section 4.4.1 (and used in Chapters 6 and 8) are also considered, gathered as one:

$$C(\Omega,\Gamma) = \underbrace{\int_\Omega \left[(y_\phi - y)^+\right]^2 dx}_{C_\phi(\Omega,\Gamma)} + \underbrace{\int_\Omega \left[(y - y_{\text{M,D}_\text{S}})^+\right]^2 dx}_{C_{\text{M,D}_\text{S}}(\Omega,\Gamma)} + \underbrace{\int_{D\setminus\Omega} \left[\left(y - y_{\text{M,D}\setminus\text{D}_\text{S}}\right)^+\right]^2 dx}_{C_{\text{M,D}\setminus\text{D}_\text{S}}(\Omega,\Gamma)}. \tag{9.2.6}$$

The effective constraint is $\dfrac{C}{C^0} = 0$ with C^0 the initial constraint. This optimization problem is finally summed up as

$$\min_{\Omega,\Gamma} J(\Omega,\Gamma) \quad \text{such that} \quad \begin{cases} \dfrac{V}{V^0} = \dfrac{V_{tar}}{V^0}, \\ \dfrac{C}{C^0} = 0. \end{cases} \tag{9.2.7}$$

9.2.2 Differentiation

The derivative with respect to each variable must now be found.

Proposition 9.1. *Let $\Gamma \in \mathcal{G}$ and $\Omega \subset D$ an open bounded set such that $\partial\Omega \in \mathcal{G}$. Then, each of the functions involved in the optimization problem (compliance, volume, path length and temperature constraint) are differentiable at Ω and $\forall \theta_\Omega \in \mathcal{C}^2\left(\overline{D},\mathbb{R}^2\right)$ such that $\forall x \in \partial\Omega_\text{D} \cup \partial\Omega_\text{N}$, $\theta_\Omega(x)\cdot n(x) = 0$,*

$$\begin{cases} D_\Omega C_{\text{ply}}(\Omega,\Gamma)(\theta_\Omega) &= \displaystyle\int_{\partial\Omega_\text{F}} \left(-Ae(u) : e(u)\right)\theta_\Omega \cdot n_\Omega ds, \\ D_\Omega L_{\text{F}}(\Omega,\Gamma)(\theta_\Omega) &= 0, \\ D_\Omega V(\Omega,\Gamma)(\theta_\Omega) &= \displaystyle\int_{\partial\Omega_\text{F}} \theta_\Omega \cdot n_\Omega ds, \\ D_\Omega C(\Omega,\Gamma)(\theta_\Omega) &= \displaystyle\int_{\partial\Omega_\text{F}} \left((y_\phi - y)^+ + (y - y_{\text{M,D}_\text{S}})^+ - \left(y - y_{\text{M,D}\setminus\text{D}_\text{S}}\right)^+\right)\theta_\Omega \cdot n_\Omega ds, \end{cases} \tag{9.2.8}$$

and differentiable at Γ such that

$$\begin{cases} D_\Gamma C_{\text{ply}}(\Omega,\Gamma)(\theta_\Gamma) &= 0, \\ D_\Gamma V(\Omega,\Gamma)(\theta_\Gamma) &= 0, \\ D_\Gamma L_{\text{F}}(\Omega,\Gamma)(\theta_\Gamma) &= \displaystyle\int_\Gamma \kappa_\Gamma \theta_\Gamma \cdot n_\Gamma ds + \theta_\Gamma(B)\cdot\tau_\Gamma(B) - \theta_\Gamma(A)\cdot\tau_\Gamma(A), \\ D_\Gamma C(\Omega,\Gamma)(\theta_\Gamma) &= \displaystyle\int_\Gamma \left(-Pp\kappa_\Gamma - P\partial_n p\right)\theta_\Gamma \cdot n_\Gamma ds \end{cases} \tag{9.2.9}$$

where the adjoint function $p \in H^1(D,\mathbb{R})$ is solution of

$$\begin{cases} -\nabla\left(\lambda\nabla p\right) + \beta p = 2\left(l_C + c_C C\right)\left(\left(y_\phi - y\right)^+ \mathbb{1}_\Omega - \left(y - y_{\text{M,D}_\text{S}}\right)^+ \mathbb{1}_\Omega - \left(y - y_{\text{M,D}\setminus\text{D}_\text{S}}\right)^+ \mathbb{1}_{D\setminus\Omega}\right) & \text{in } D, \\ \lambda\partial_n p = 0 & \text{on } \partial D. \end{cases} \tag{9.2.10}$$

PROOF.
The method of Céa is applied for each function, involving the variational formulation of linear elasticity and the variational formulation of the heat equation. As in section 3.3, the compliance problem is self-adjoint and Proposition 3.4 is applied. The only point of attention is on the differentiation of the constraint $C_{\text{M,D}\setminus\text{D}_\text{S}}$:

$$\begin{aligned} C_{\text{M,D}\setminus\text{D}_\text{S}} &= \int_D \left(\left(y - y_{\text{M,D}\setminus\text{D}_\text{S}}\right)^+\right)^2 \mathbb{1}_{D\setminus\Omega} ds \\ &= \int_D \left(\left(y - y_{\text{M,D}\setminus\text{D}_\text{S}}\right)^+\right)^2 dx - \int_\Omega \left(\left(y - y_{\text{M,D}\setminus\text{D}_\text{S}}\right)^+\right)^2 dx. \end{aligned}$$

Thus,

$$D_\Omega C_{\text{M,D}\setminus\text{D}_\text{S}}(\theta_\Omega) = 0 - \int_{\partial\Omega} \left(\left(y - y_{\text{M,D}\setminus\text{D}_\text{S}}\right)^+\right)^2 \theta_\Omega \cdot n_\Omega ds.$$

Since the temperature and the path are not involved in the elastic problem nor in the volume and compliance, the differentiation with respect to the path corresponds to the computations provided in section 6.3.1 (Chapter 6). □

To represent and update the shape, a level set method, presented in Section 3.3.1 is chosen. To compute the update direction, the gradients of the functions involved (C'_{ply}, $L'_{\text{F}} = 0$, V', C') are required and are computed using the regularization-extension equation (3.2.21) (Section 3.2.4). The advection step is given by (3.3.13) (Section 3.3.2). As for the path update, it follows the process detailed in Chapter 6 with the determination of gradients with respect to the path ($\nabla_\Gamma C_{\text{ply}} = 0$, $\nabla_\Gamma L_{\text{F}}$, $\nabla_\Gamma V = 0$, $\nabla_\Gamma C$).

9.3 Numerical algorithm

The optimization method chosen is a double loop algorithm. The outer loop is related to the shape Ω whereas the inner loop focuses on the path Γ. The number of iterations in each loop determines the balance between the optimization of both variables. The following subsections describe each of them.

9.3.1 Path update: inner loop

The inner loop consists in optimizing the path. Assume that the shape Ω^n has been updated into Ω^{n+1}. The path Γ^n must now be updated. The inner loop consists in performing N_Γ iterations ($n_\Gamma \in [\![1, N_\Gamma]\!]$) on the path starting from Γ^n assuming that the shape to build is Ω^{n+1}. This inner loop optimization problem is closed from what has been done in Chapter 6. Indeed, both the compliance and volume do not depend on the path. The inner loop thus consists in minimizing the length L_{F} under the constraint $C = 0$. An augmented Lagrangian method is used with the effective inner loop objective function

$$\mathcal{L}_{ALM,in}(\Gamma) = L_{\text{F}} + l_{C,in} C + \frac{c_{C,in}}{2} C^2. \tag{9.3.1}$$

The Lagrangian multiplier is initialized by 1 ($l^0_{C,in} = 1$) and the penalizer is fixed to 1 ($c_{C,in} = 1$). A step coefficient C_{s_Γ} is introduced to determine the path advection step. Initialized to 1, it is updated by $\min(1.2C_{s_\Gamma}, 1)$ if the effective objective function decreases (iteration accepted) and by $0.6C_{s_\Gamma}$ else. Note

that in the inner loop, the tolerance classically defined to allow for the function to increase is set to 1. In case the iteration n_Γ is accepted, the multiplier is updated as

$$l_{C,in}^{n_\Gamma+1} = l_{C,in}^{n_\Gamma} + c_{C,in} C^{n_\Gamma}. \tag{9.3.2}$$

In the following, the inner loop is written innerLoop$(\Omega, \Gamma^{n_\Gamma=0}, N_\Gamma)$.

9.3.2 Shape update: outer loop

The outer loop consists in optimizing the shape. Let n be an outer loop iteration with Ω^n and Γ^n the variables at this iteration. This outer loop consists in determining the new shape Ω^{n+1}, running the inner loop to determine the corresponding path Γ^{n+1} and accepting or rejecting the iteration.

Augmented Lagrangian method

To deal with both the volume and temperature constraints, an augmented Lagrangian method is used. Introducing the multipliers l_V and l_C, the penalizers c_V and c_C, the effective objective function is then

$$\mathcal{L}_{\text{ALM}}(\Omega, \Gamma, l_V, l_C; c_V, c_C) = \frac{C_{\text{ply}}}{C_{\text{ply}}^0} + \frac{L_{\text{F}}}{L_{\text{F}}^0} + l_V \frac{V - V_{tar}}{V^0} + \frac{c_V}{2}\left(\frac{V - V_{tar}}{V^0}\right)^2 + l_C \frac{C}{C^0} + \frac{c_C}{2}\left(\frac{C}{C^0}\right)^2. \tag{9.3.3}$$

The gradient to this effective objective function is then computed. The advection step is given by (3.3.13) (Section 3.3.2) with $C_{s,\Omega}^0 = 5$, $\eta_{\text{acc}}^s = 1.2$, $\eta_{\text{ref}}^s = 0.6$ and the tolerance *tol* initialized to 1.6 and multiplied by 0.9 every 50 iterations. This advection process is numerical run following the algorithm described in [41, 181] and the redistanciation process is provided by Freefem++[96]. The path is then updated by application of the inner loop.

initialization: Ω^0, Γ^{-1}, $C_{s,\Omega}^0 = 5$, $l_V = 1$, $c_V = 1$, $l_C^0 = 1$, $c_C = 1$
computation of the initial compliance C_{ply}^0, volume V^0, length L_{F}^{-1}, constraint C^{-1}
determination of Γ^0 (and L_{F}^0, C^0) by application of innerLoop$(\Omega^0, \Gamma^{-1}, N_\Gamma^0)$
computation of the shape derivative
for $it_\Omega \in [\![0, N_\Omega]\!]$ **do**
  update the shape: $\Omega = \Omega^{it_\Omega} - s_\Omega^{it_\Omega} \mathcal{L}'^{it_\Omega}_{\text{ALM}}$
  compute the new compliance C_{ply} and volume V
  determination of Γ (and L_{F}, C) by application of innerLoop$(\Omega, \Gamma^{it_\Omega}, N_\Gamma^{it_\Omega})$
  if $\mathcal{L}_{\text{ALM}}(\Omega, \Gamma, l_V^{it_\Omega}, l_C^{it_\Omega}) \leq tol \mathcal{L}_{\text{ALM}}(\Omega^{it_\Omega}, \Gamma^{it_\Omega}, l_V^{it_\Omega}, l_C^{it_\Omega})$ **then**
    iteration accepted: $\Omega^{it_\Omega+1} = \Omega$, $\Gamma^{it_\Omega+1} = \Gamma$
    update the lagrange multiplier $l_V^{it_\Omega+1} = l_V^{it_\Omega} + c_V V^{it_\Omega+1}$
    update the lagrange multiplier $l_C^{it_\Omega} + c_C C^{it_\Omega+1}$
    compute the new objective function and the shape derivative
    increase the step coefficient related to the shape: $C_{s,\Omega} = \min(1.2 C_{s,\Omega}, 5)$
  end
  else
    iteration rejected (shape and path rejected)
    decrease the step coefficient related to the shape: $C_{s,\Omega} = 0.6 C_{s,\Omega}$
  end
end

Algorithm 9.1: Iterative double loop algorithm to optimize the shape and the path.

Dichotomy of the volume Lagrange multiplier

If the augmented Lagrangian method is the simplest way to include the constraints to the optimization problem, it guaranties their fulfillment at convergence only. However, in most numerical applications, we must stop the algorithm after a fixed number of iterations N_Ω. This is an issue for the volume constraint. Indeed, the objective of this chapter is to determine the impact of the path determination on the design. To facilitate the interpretation, we decide to fix the volume allowed: the impact of the path corresponds to a material redistribution without any addition or removal of volume. The augmented Lagrangian method is not strong enough to force the volume constraint to be satisfied. This could be easier with the null space algorithm but another technique is a lot more efficient for this: perfectly adapting the volume

at each iteration by choosing adequately the volume Lagrange multiplier l_V. The effective augmented Lagrangian function (9.3.3) is now split into two different functions with $\mathcal{L}_{\text{ALM}} = \overline{\mathcal{L}_{\text{ALM}}} + l_V \mathcal{L}_V$ and

$$\begin{cases} \overline{\mathcal{L}_{\text{ALM}}} = \dfrac{C_{\text{ply}}}{C^0_{\text{ply}}} + \dfrac{L_{\text{F}}}{L^0_{\text{F}}} + \dfrac{c_V}{2}\left(\dfrac{V - V_{tar}}{V^0}\right)^2 + l_C \dfrac{C}{C^0} + \dfrac{c_C}{2}\left(\dfrac{C}{C^0}\right)^2 \\ \mathcal{L}_V = \dfrac{V - V_{tar}}{V^0}. \end{cases} \tag{9.3.4}$$

This leads to a new function gradient (and thus a new descent direction) $\mathcal{L}'_{\text{ALM}} = \overline{\mathcal{L}_{\text{ALM}}}' + l_V \mathcal{L}'_V$. This algorithm modifies the step computation and the shape advection. Algorithm 9.2 details this process. If the Lagrange multiplier for the temperature constraint l_C is still initialized to 1, the volume multiplier is initialized to

$$l_V^0 = -\frac{D_\Omega \overline{\mathcal{L}_{\text{ALM}}}(n_\Omega)}{D_\Omega \mathcal{L}_V(n_\Omega)}. \tag{9.3.5}$$

initialization: Ω^0, Γ^{-1}, $C^0_{s,\Omega} = 5$, $c_V = 1$, $l_C^0 = 1$, $c_C = 1$
computation of the initial compliance C^0_{ply}, volume V^0, length L_{F}^{-1}, constraint C^{-1}
determination of Γ^0 (and L^0_{F}, C^0) by application of innerLoop$(\Omega^0, \Gamma^{-1}, N_\Gamma^0)$
computation of the gradients $\left(\overline{\mathcal{L}_{\text{ALM}}}'\right)^0$ and $\left(\mathcal{L}'_V\right)^0$
initialization of l_V: $l_V^0 = -\dfrac{D_\Omega \overline{\mathcal{L}_{\text{ALM}}}(n_\Omega)}{D_\Omega \mathcal{L}_V(n_\Omega)}$ **for** $it_\Omega \in [\![0, N_\Omega]\!]$ **do**
  advection step $s_\Omega^{it_\Omega}$ defined following (3.3.13) (Section 3.3.2) for $\left(\overline{\mathcal{L}_{\text{ALM}}}'\right)^{it_\Omega} + l_V^{it_\Omega}\left(\mathcal{L}'_V\right)^{it_\Omega}$
  update the shape: $\Omega^{\text{test}} = \Omega^{it_\Omega} - s_\Omega^{it_\Omega}\left(\left(\overline{\mathcal{L}_{\text{ALM}}}'\right)^{it_\Omega} + l_V^{it_\Omega}\left(\mathcal{L}'_V\right)^{it_\Omega}\right)$
  $l_V^{\text{test}} = l_V^{it_\Omega}$, $l_V^{\text{MIN}} = l_V^{it_\Omega} - 10$, $l_V^{\text{MAX}} = l_V^{it_\Omega} + 10$, $V^{\text{test}} = V\left(\Omega^{\text{test}}\right)$, $k = 0$
  while $V^{test} \notin V_{tar} + [-\epsilon_{dicho}, \epsilon_{dicho}]$ *and* $k \leq 30$ **do**
    if $V^{test} \leq V_{tar} - \epsilon_{dicho}$ **then**
      $l_V^{\text{MAX}} = l_V^{\text{test}}$
      $l_V^{\text{test}} = \dfrac{l_V^{\text{MIN}} + l_V^{\text{MAX}}}{2}$
    end
    if $V^{test} \geq V_{tar} + \epsilon_{dicho}$ **then**
      $l_V^{\text{MIN}} = l_V^{\text{test}}$
      $l_V^{\text{test}} = \dfrac{l_V^{\text{MIN}} + l_V^{\text{MAX}}}{2}$
    end
    $\Omega^{\text{test}} = \Omega^{it_\Omega} - s_\Omega^{it_\Omega}\left(\left(\overline{\mathcal{L}_{\text{ALM}}}'\right)^{it_\Omega} + l_V^{\text{test}}\left(\mathcal{L}'_V\right)^{it_\Omega}\right)$, $V^{\text{test}} = V(\Omega^{\text{test}})$ and $k = k + 1$.
  end
  $\Omega = \Omega^{\text{test}}$: compute the new compliance C_{ply} and volume V
  determination of Γ (and L_{F}, C) by application of innerLoop$(\Omega, \Gamma^{it_\Omega}, N_\Gamma^{it_\Omega})$
  if $\mathcal{L}_{\text{ALM}}(\Omega, \Gamma, l_V^{it_\Omega}, l_C^{it_\Omega}) \leq \mathcal{L}_{\text{ALM}}(\Omega^{it_\Omega}, \Gamma^{it_\Omega}, l_V^{it_\Omega}, l_C^{it_\Omega})tol$ **then**
    iteration accepted: $\Omega^{it_\Omega+1} = \Omega$, $\Gamma^{it_\Omega+1} = \Gamma$
    update the lagrange multiplier $l_V^{it_\Omega+1} = l_V^{\text{test}} + c_V V^{it_\Omega+1}$
    update the lagrange multiplier $l_C^{it_\Omega} + c_C C^{it_\Omega+1}$
    compute the new objective function and the gradients $\left(\overline{\mathcal{L}_{\text{ALM}}}'\right)^{it_\Omega+1}$ and $\left(\mathcal{L}'_V\right)^{it_\Omega+1}$
    increase the step coefficient related to the shape: $C_{s,\Omega} = \min\left(1.2C_{s,\Omega}, 5\right)$
  end
  else
    iteration rejected (shape and path rejected)
    decrease the step coefficient related to the shape: $C_{s,\Omega} = 0.6C_{s,\Omega}$
  end
end

Algorithm 9.2: Iterative double loop algorithm to optimize the shape and the path.

The coefficient ϵ_{dicho} is initialized to 0.05 and multiplied by 0.95 at each iteration during the 120 first ones. The volume tolerance is then fixed to $\epsilon_{dicho} = 1.1\,10^{-4}$.

9.4 Numerical results

An initialization state INI is defined with a shape Ω^{INI} and a path Γ^{INI} (subfigure (a) in each Figure). From Algorithm 9.2, different tests are run, summed up in Figure 9.1:

- SONLY: **shape optimized - path fixed**, without any consideration of the temperature constraints ($l_C = 0$ and $c_C = 0$). This test is initialized with $\Omega^0 = \Omega^{\text{INI}}$.
- STEMP: **shape optimized - path fixed**, taking into consideration the temperature constraints ($l_C^0 = 1$, $c_C = 1$, $\forall it_\Omega \in [\![0, N_\Omega]\!]$, $N_\Gamma^{it_\Omega} = 0$). This test is initialized with $\Omega^0 = \Omega^{\text{INI}}$ and the fixed path used for the temperature computations is Γ^{INI}.
- SP-fromINI: **shape optimized - path optimized**. This test optimizes both variables taking into account temperature constraints ($l_C^0 = 1$, $c_C = 1$). The number of inner loop iterations follows Table 9.1. Recall that the inner loop is also broken when the inner loop step coefficient is smaller than 10^{-8}. This test is initialized with $\Omega^0 = \Omega^{\text{INI}}$ and $\Gamma^0 = \Gamma^{\text{INI}}$.

it_Ω	$[\![0,24]\!]$	25	$[\![26,44]\!]$	45	$[\![46,49]\!]$	50	$[\![51,54]\!]$	55	56	57	58	59	$[\![60,300]\!]$
$N_\Gamma^{it_\Omega}$	0	50	0	45	0	40	0	35	30	25	20	15	10

Table 9.1: Number of inner loop maximum number of iterations N_Γ depending on the outer loop iteration it_Ω.

- PONLY: **shape fixed - path optimized**. This test is initialized by $\Gamma^0 = \Gamma^{\text{INI}}$ and the fixed shape to build is Ω^{SONLY} resulting from the optimization test SONLY.
- SP-fromPONLY: **shape optimized - path optimized**. This test optimizes both variables taking into account temperature constraints ($l_C^0 = 1$, $c_C = 1$). The number of inner loop iterations follows Table 9.1. Recall that the inner loop is also broken when the inner loop step coefficient is smaller than 10^{-8}. This test is initialized with the shape $\Omega^0 = \Omega^{\text{SONLY}}$ resulting from the optimization test SONLY and with the path $\Gamma^0 = \Gamma^{\text{PONLY}}$ resulting from the optimization test PONLY.

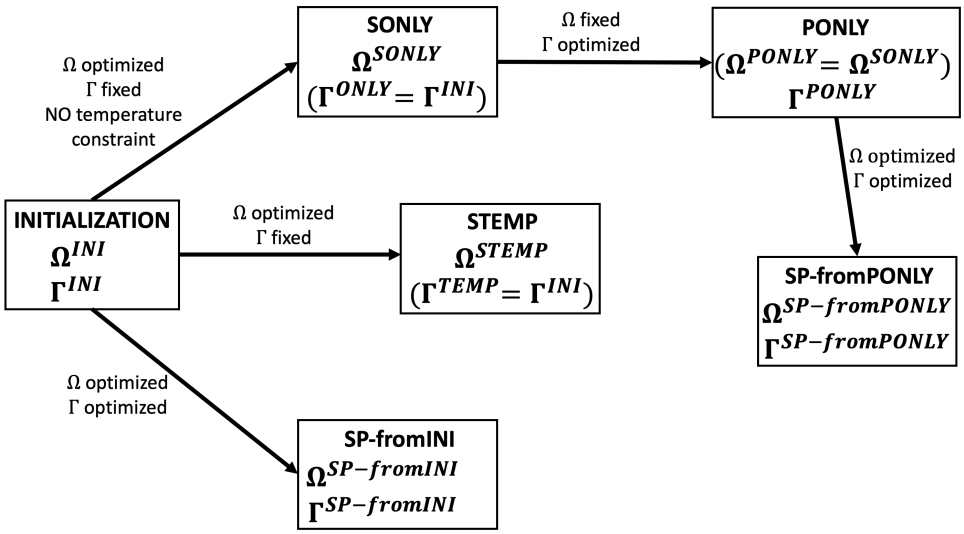


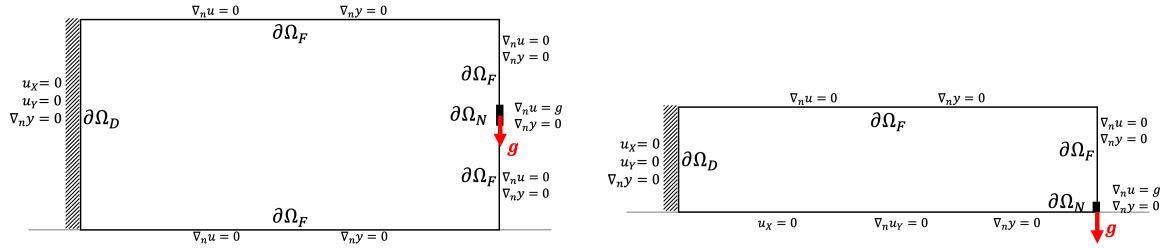
Figure 9.1: Recap scheme of the different tests

For each of these tests, the final shape, path and temperature are given. A Table sums up the final compliance, volume, path length and adimensionalized temperature constraints (7.2.27). A final radar chart sums up the quantitative results. Note that in these graphs, for each of the functions represented

(compliance and temperature constraints), the axes are reversed so that the best solutions correspond to the curve with the biggest area.

9.4.1 Cantilever test case

The first test case is a cantilever. A symmetry condition is applied and the working domain is $D = [-1.4, 1.4] \times [0, 0.7]$ given in mm. This working domain is meshed with 12800 triangular elements. The corresponding mechanical settings for the non symmetric and symmetric cases are shown in Figure 9.2. For both the titanium and aluminium, the Poisson ratio is fixed to $\nu = 0.3$ and the Young coefficient is $E = 1$. This assumption comes from the linear relation between the displacement and Young coefficient. The loading is $g = (0, -2)$. The temperature settings remain the same than in Chapters 6 and 8 (see Section 6.6 and 6.7). To simplify the problem, the path is composed of only one connected component. Thus, as in Chapter 6 for symmetric tests, the point of the path initially belonging to the (Ox) must remain on this axis.



(a) Non symmetric test case

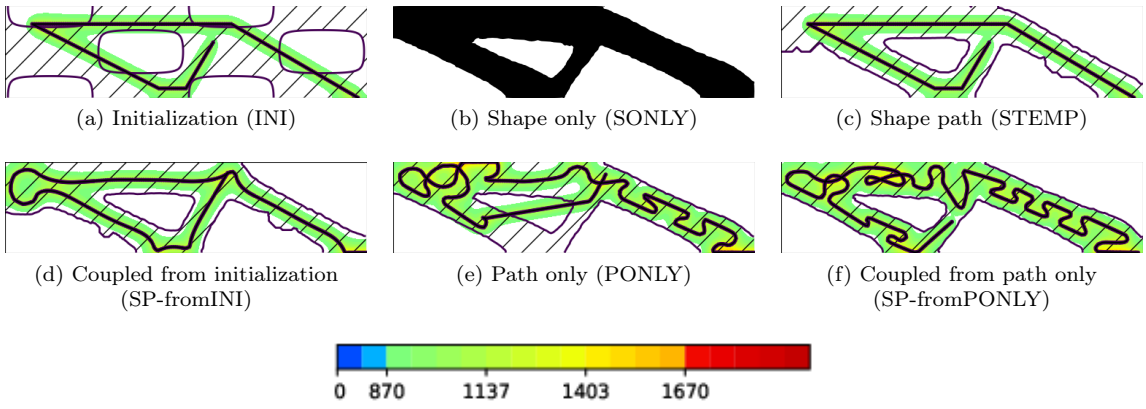
(b) Symmetric test case

Figure 9.2: Cantilever test case with the symmetry conditions for the mechanical and heat problems

For each of the aluminium and titanium test cases, two different volume ratios are tested: $V_{tar} = 0.9V^0$ and $V_{tar} = 1.1V^0$.

Aluminium test case

For the aluminium, the results of the five tests for each volume ratio are respectively presented in Figures 9.3, 9.5, by Tables 9.2, 9.3 and by the quantitative results visualization Figures 9.4 and 9.6. The initial volume is $V_0 = 1.15722e - 06m^2$.



(a) Initialization (INI)

(b) Shape only (SONLY)

(c) Shape path (STEMP)

(d) Coupled from initialization (SP-fromINI)

(e) Path only (PONLY)

(f) Coupled from path only (SP-fromPONLY)

(g) Colorbar (K)

Figure 9.3: Initialization and shape optimization for different tests, aluminium, cantilever, $V_{tar} = 0.9V_0$

These first results show that coupling a classic shape optimization algorithm to the path optimization technique developed in this work leads to an adaptation of both the shape and the path. It is clear that the temperature constraints impact the shape since, the results in Figures 9.3(c), 9.5(c) are different from 9.3(b), 9.5(b) whereas the only difference is the fact that the temperature constraints are taken

case	$\frac{V}{V^0}$	C_{ply}	L_{F}	$\overline{C}_{\phi}$	$\overline{C}_{\text{M,D}_\text{S}}$	$\overline{C}_{\text{M,D}\setminus\text{D}_\text{S}}$
INI	1.000	5.666e−6	4.383e−3	4.879e−3	0.000	7.296e−3
SONLY	0.900	2.245e−6	4.383e−3	1.696e−3	0.000	1.473e−3
STEMP	0.900	2.349e−6	4.383e−3	1.272e−3	0.000	1.219e−7
SP-fromINI	0.900	2.377e−6	5.396e−3	4.932e−4	0.000	3.698e−7
PONLY	0.900	2.245e−6	7.353e−3	1.291e−3	9.914e−10	2.823e−3
SP-fromPONLY	0.900	2.262e−6	8.316e−3	1.899e−5	0.000	2.717e−5

Table 9.2: Quantitative results for the different tests, aluminium, cantilever, $V_{tar} = 0.9V_0$

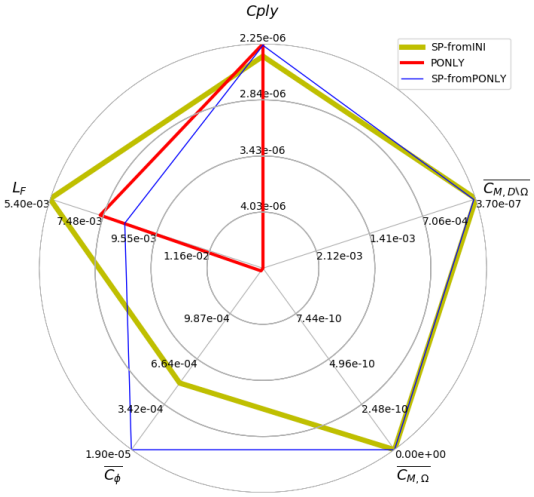
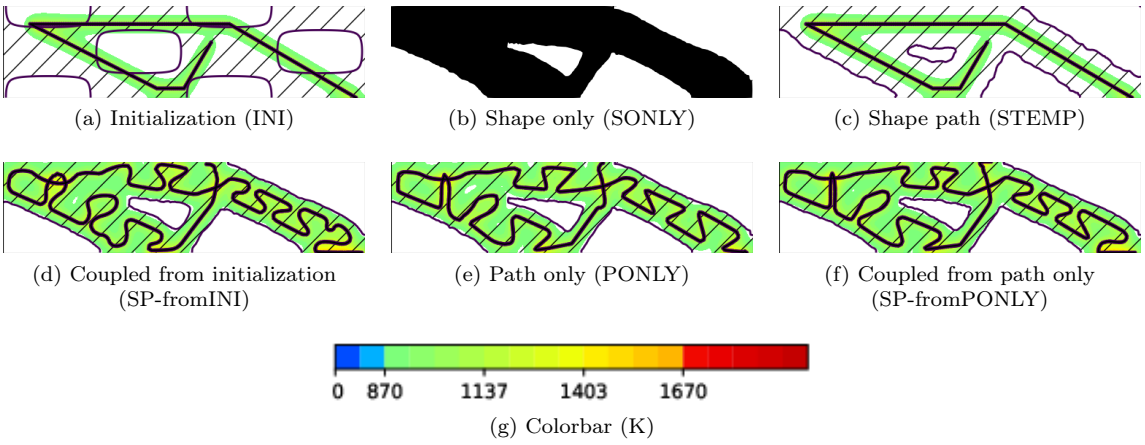


Figure 9.4: Quantitative values visualization, aluminium, cantilever, $V_{tar} = 0.9V_0$



(a) Initialization (INI) (b) Shape only (SONLY) (c) Shape path (STEMP)

(d) Coupled from initialization (SP-fromINI) (e) Path only (PONLY) (f) Coupled from path only (SP-fromPONLY)

(g) Colorbar (K)

Figure 9.5: Initialization and shape optimization for different tests, aluminium, cantilever, $V_{tar} = 1.1V_0$

into account in (c). Indeed, in both cases, the lower right bars is thickened to reduce the maximum temperature constraint out of the shape to build whereas other bars (upper right in Figure 9.3 and lower left in 9.5) are made thinner. Figures 9.3(c), 9.5(c) also make really clear how fixing the volume impacts the optimization: the shape cannot exactly adapt to the path and reduce the phase constraint without decreasing the volume. The dichotomy prevents this.
In Figures 9.3(e), 9.5(e), the shape is fixed to the result from SONLY and the path is optimized. In Figure 9.3(e), this optimization fails: the phase constraint is not satisfied and the path crosses a zone

case	$\frac{V}{V^0}$	C_{ply}	L_{F}	$\overline{C}_{\phi}$	$\overline{C}_{\text{M,D}_\text{S}}$	$\overline{C}_{\text{M,D}\setminus\text{D}_\text{S}}$
INI	1.000	5.666e−6	4.383e−3	4.879e−3	0.000	7.296e−3
SONLY	1.100	1.917e−6	4.383e−3	2.465e−3	0.000	1.396e−5
STEMP	1.100	2.001e−6	4.383e−3	2.215e−3	0.000	0.000
SP-fromINI	1.100	1.920e−6	9.572e−3	9.979e−6	0.000	1.393e−5
PONLY	1.100	1.917e−6	8.848e−3	1.659e−5	0.000	1.993e−5
SP-fromPONLY	1.100	1.918e−6	9.337e−3	1.098e−5	0.000	1.477e−5

Table 9.3: Quantitative results for the different tests, aluminium, cantilever, $V_{tar} = 1.1V_0$

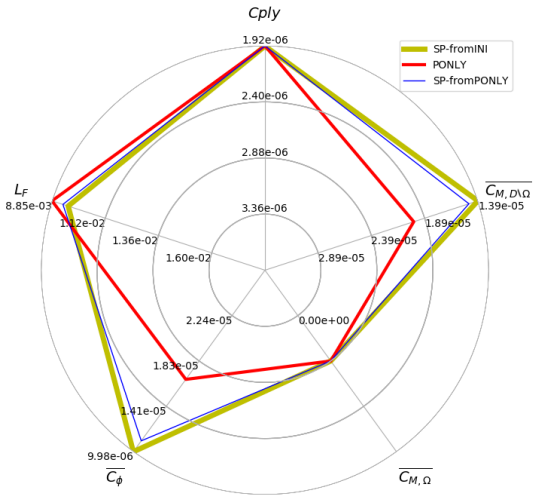


Figure 9.6: Quantitative values visualization, aluminium, cantilever, $V_{tar} = 1.1V_0$

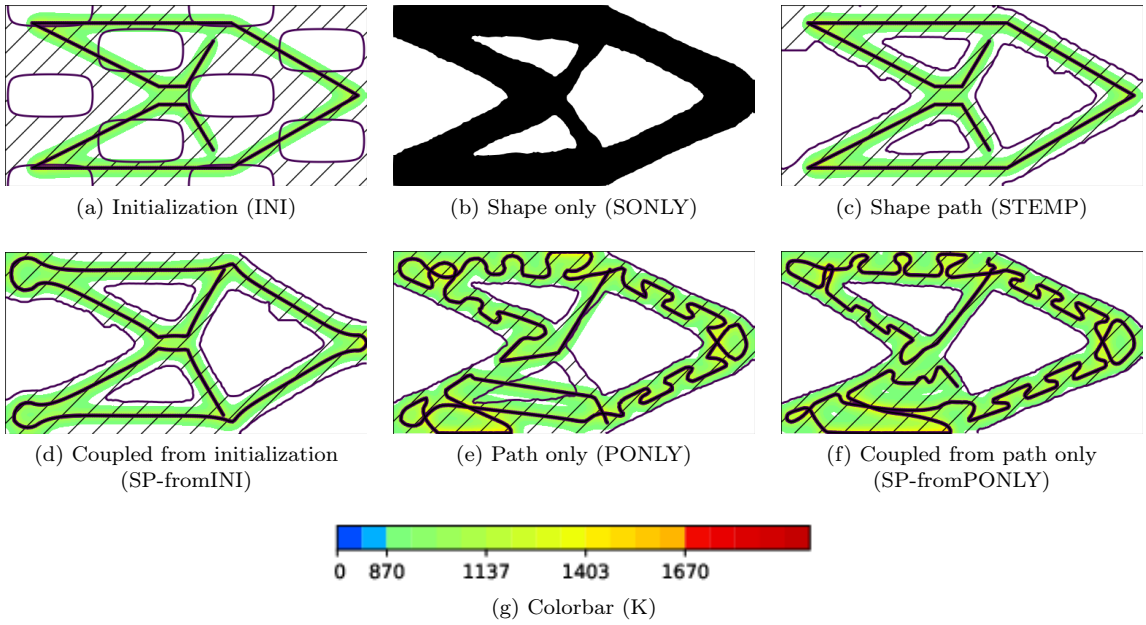
out of the shape to build. Indeed, it seems complicated for the path to build the right lower bar: the conductivity is high and the path cannot cross this bar without violating the maximum temperature constraint out of the domain. Struggling to remedy to this problem, the optimization ends up by being stuck into a local minimum that does not satisfy the temperature constraints. On the contrary in Figure 9.5(e), the bar is thicker and the path optimization satisfies well the constraints. A first criterion could thus be the thickness of the bars to build and an interesting perspective would be to compare the results obtained for design under thickness constraints (see [141]) to the concurrent design of shape and scanning path. In both cases, an interesting pattern must be noted in the upper right bar. The conductivity is not high enough to melt the powder in this bar with a straight line. In both cases, the algorithm designs an Omega-pattern. Note that these patterns could be different if allowing for the modification of the number of path connected components.

Coupled optimizations are run in each case starting from two different initializations: from Figure (a) to Figure (d) and from Figure (e) to Figure (f). In Figure 9.3, the strategies are different from one initialization to another. Starting from (c), the shape is slightly modified with simple bars thickness reduction or increase and the path is adapted to correctly melt the upper left corner. Starting from (e), the bars thickness are not modified much. The modifications simply allow the path to get out of the local minimum and ends the optimization process. Note that in this case, the path does not cross the lower right bar. In Figure 9.5, the results are quite similar. It can be especially noted that concurrent optimization starting from (c) leads to the same Omega-patterns than obtained in (e) and (f).

These first results underline two main points. Once again, the initialization of the path is crucial. Indeed, not only the final path is influenced by this choice, but the shape might also be modified. Choosing this initialization is not intuitive and must be coherent with the optimal design. However, running first shape optimization and then path optimization to get the initializations of the concurrent optimization might lead to results very different from the expectations. The second point is the importance of the bars thickness in the design. Indeed, in both concurrent optimization tests, the thin bars have been thick-

ened until they almost correspond to the thickness of the melted powder domain generated by the source.

We finally run the optimization on the whole working domain to evaluate the impact of the symmetry. For $V_{tar} = 0.9V_0$, the results are shown in Figure 9.7 and Table 9.4.



(a) Initialization (INI) (b) Shape only (SONLY) (c) Shape path (STEMP)

(d) Coupled from initialization (SP-fromINI) (e) Path only (PONLY) (f) Coupled from path only (SP-fromPONLY)

(g) Colorbar (K)

Figure 9.7: Initialization and shape optimization for different tests, aluminium, full cantilever, $V_{tar} = 0.9V_0$

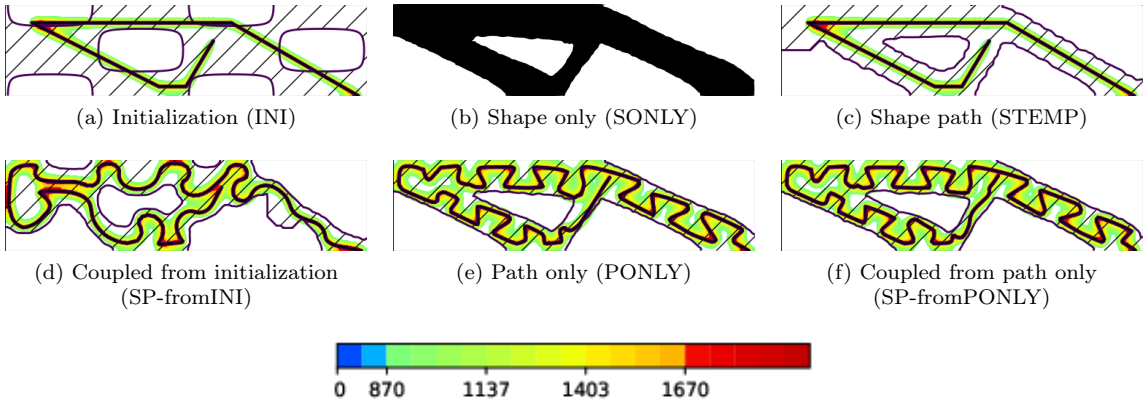
case	$\frac{V}{V^0}$	C_{ply}	L_{F}	$\overline{C}_{\phi}$	$\overline{C}_{\text{M,D}_\text{S}}$	$\overline{C}_{\text{M,D}\setminus\text{D}_\text{S}}$
INI	1.000	1.133e−5	8.765e−3	4.879e−3	0.000	7.296e−3
SONLY	0.900	4.493e−6	8.765e−3	1.628e−3	0.000	1.105e−3
STEMP	0.900	4.715e−6	8.765e−3	1.159e−3	0.000	4.452e−8
SP-fromINI	0.900	4.667e−6	1.024e−2	5.224e−4	0.000	3.769e−8
PONLY	0.900	4.493e−6	1.594e−2	1.437e−4	0.000	1.849e−3
SP-fromPONLY	0.900	4.558e−6	1.604e−2	1.768e−5	0.000	3.045e−5

Table 9.4: Quantitative results for the different tests, aluminium, full cantilever, $V_{tar} = 0.9V_0$

If the results from Figure 9.7(a-b-c-d) are very similar to Figure 9.3(a-b-c-d), it seems in the optimization (of the path only) that the path symmetry has been broken. In Figure 9.3(e), the path has already difficulties to cross the right lower bar. In Figure 9.7(e), it manages to cross it yet encroaching on the lower half of the Figure thus preventing the same method for the symmetric to this right loser bar. Note also that on the right and middle of the Figure (junction location in Figure 9.3(e)), the path is not horizontal anymore but introduces an intersection. As for Figure 9.7(f), it cannot be symmetric because starting from a non symmetric initialization, and induces the loss of the lower hole. Yet, the shape evolution allows for a correct satisfaction of the constraints.

Titanium test case

For the titanium, the results of the five tests for each volume ratio are respectively presented in Figures 9.8, 9.10, by Tables 9.5, 9.6 and by the quantitative results visualization Figures 9.9 and 9.11.
Before getting further in the comments, note that the Young modulus for the titanium is higher than for the aluminium. Thus, with the real values of the Young modulus, the Poisson ratio and for the same loading, the bars obtained for the titanium could be thinner than shown in the Figures 9.8 and 9.10.



(a) Initialization (INI) (b) Shape only (SONLY) (c) Shape path (STEMP)

(d) Coupled from initialization (SP-fromINI) (e) Path only (PONLY) (f) Coupled from path only (SP-fromPONLY)

(g) Colorbar (K)

Figure 9.8: Initialization and shape optimization for different tests, titanium, cantilever, $V_{tar} = 0.9V_0$

case	$\frac{V}{V^0}$	C_{ply}	L_{F}	$\overline{C}_{\phi}$	$\overline{C}_{\text{M,D}_\text{S}}$	$\overline{C}_{\text{M,D}\setminus\text{D}_\text{S}}$
INI	1.000	5.666e−6	4.383e−3	1.692e−1	2.792e−5	2.286e−2
SONLY	0.900	2.245e−6	4.383e−3	8.909e−2	1.545e−4	3.934e−3
STEMP	0.900	2.364e−6	4.383e−3	7.561e−2	1.545e−4	0.000
SP-fromINI	0.900	3.374e−6	7.401e−3	1.910e−2	3.776e−4	1.342e−4
PONLY	0.900	2.245e−6	8.628e−3	2.138e−3	1.096e−4	2.245e−4
SP-fromPONLY	0.900	2.250e−6	8.632e−3	1.656e−3	1.242e−4	1.450e−4

Table 9.5: Quantitative results for the different tests, titanium, cantilever, $V_{tar} = 0.9V_0$

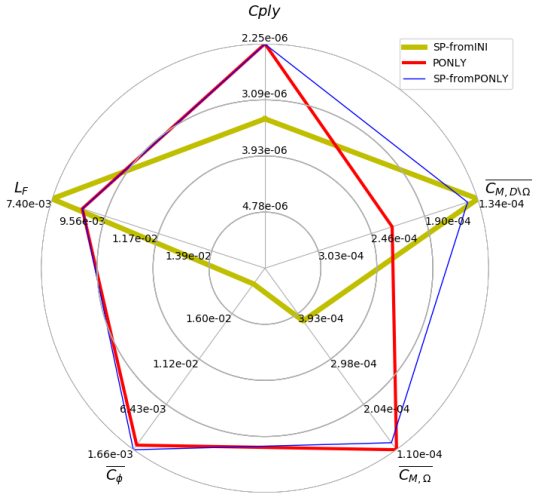
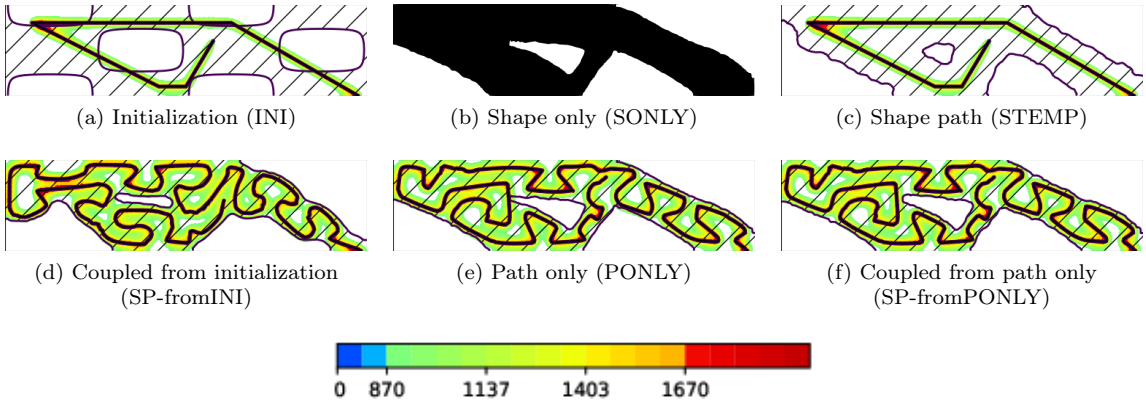


Figure 9.9: Quantitative values visualization, titanium, cantilever, $V_{tar} = 0.9V_0$

First of all, the results obtained corroborate the remarks in the aluminium test case. The thickness of the bars matters. The conductivity of the titanium is lower. If this makes easier for the path to cross thin lines, it complicates the filling of thick ones. In both Figures 9.8 and 9.10, this is clear when running concurrent optimization from (a) to (d). In (d), the shape's boundary really adapts to the path: thickness and curvatures are modified to fit the phase constraint at the cost of the compliance. Another observation is once again the design of an Omega-pattern to fill thick bars. This confirms the interest of this pattern, that had already pointed out for the aluminium in Figure 6.22 in Chapter 6.



(a) Initialization (INI) (b) Shape only (SONLY) (c) Shape path (STEMP)

(d) Coupled from initialization (SP-fromINI) (e) Path only (PONLY) (f) Coupled from path only (SP-fromPONLY)

(g) Colorbar (K)

Figure 9.10: Initialization and shape optimization for different tests, titanium, cantilever, $V_{tar} = 1.1V_0$

case	$\frac{V}{V^0}$	C_{ply}	L_{F}	$\overline{C}_{\phi}$	$\overline{C}_{\text{M,D}_\text{S}}$	$\overline{C}_{\text{M,D}\setminus\text{D}_\text{S}}$
INI	1.000	5.666e−6	4.383e−3	1.692e−1	2.792e−5	2.286e−2
SONLY	1.100	1.917e−6	4.383e−3	1.125e−1	1.264e−4	0.000
STEMP	1.100	2.115e−6	4.383e−3	1.080e−1	1.264e−4	0.000
SP-fromINI	1.100	2.140e−6	1.021e−2	3.050e−3	1.405e−4	6.077e−5
PONLY	1.100	1.917e−6	1.003e−2	1.394e−3	9.727e−5	3.666e−4
SP-fromPONLY	1.100	1.921e−6	1.004e−2	1.203e−3	9.742e−5	5.010e−5

Table 9.6: Quantitative results for the different tests, titanium, cantilever, $V_{tar} = 1.1V_0$

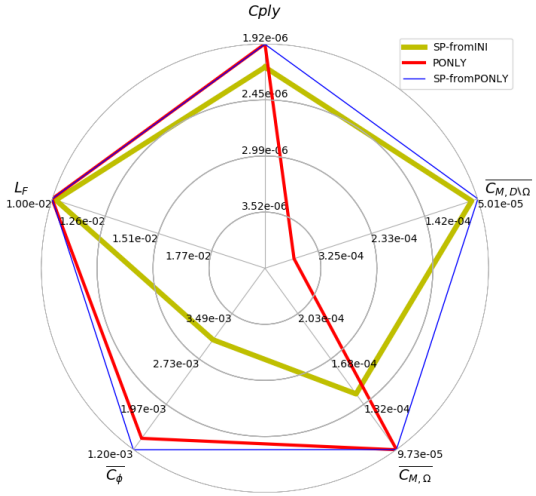
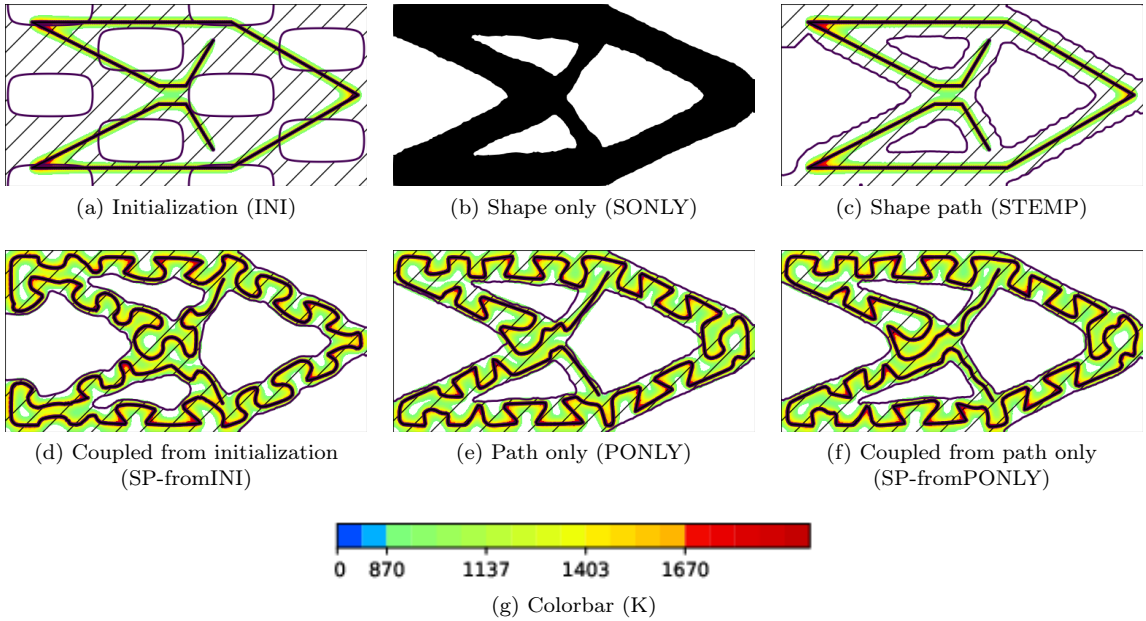


Figure 9.11: Quantitative values visualization, titanium, cantilever, $V_{tar} = 1.1V_0$

In the titanium test case, because the conductivity is small, the shape is highly modified in concurrent optimization starting from (c): the phase constraint is far from being fulfilled at initialization. On the contrary, the shape slightly adapts when starting from PONLY (e) since the phase constraint is well satisfied. The path and shape are only adapted to fit the remaining zones that remain unmelted.

We finally run the optimization on the whole working domain to evaluate the impact of the symmetry. For $V_{tar} = 0.9V_0$, the results are shown in Figure 9.12 and Table 9.7.



(a) Initialization (INI) (b) Shape only (SONLY) (c) Shape path (STEMP)

(d) Coupled from initialization (SP-fromINI) (e) Path only (PONLY) (f) Coupled from path only (SP-fromPONLY)

(g) Colorbar (K)

Figure 9.12: Initialization and shape optimization for different tests, titanium, full cantilever, $V_{tar} = 0.9V_0$

case	$\frac{V}{V^0}$	C_{ply}	L_{F}	$\overline{C}_{\phi}$	$\overline{C}_{\text{M,D}_\text{S}}$	$\overline{C}_{\text{M,D}\setminus\text{D}_\text{S}}$
INI	1.000	1.133e−5	8.765e−3	1.692e−1	2.792e−5	2.286e−2
SONLY	0.900	4.493e−6	8.765e−3	8.696e−2	1.545e−4	2.765e−3
STEMP	0.900	4.960e−6	8.765e−3	7.285e−2	1.545e−4	0.000
SP-fromINI	0.900	4.915e−6	1.756e−2	1.917e−3	1.461e−4	6.759e−5
PONLY	0.900	4.493e−6	1.730e−2	2.682e−3	1.119e−4	8.196e−4
SP-fromPONLY	0.900	4.500e−6	1.735e−2	2.017e−3	1.608e−4	1.446e−4

Table 9.7: Quantitative results for the different tests, titanium, full cantilever, $V_{tar} = 0.9V_0$

As for the aluminium, dealing with the entire geometry induces slight unsymmetries especially to correclty melt the middle zone. Yet, the results remain quite similar with the use of the Omega-pattern and the boundary adaptations to the temperature constraints.

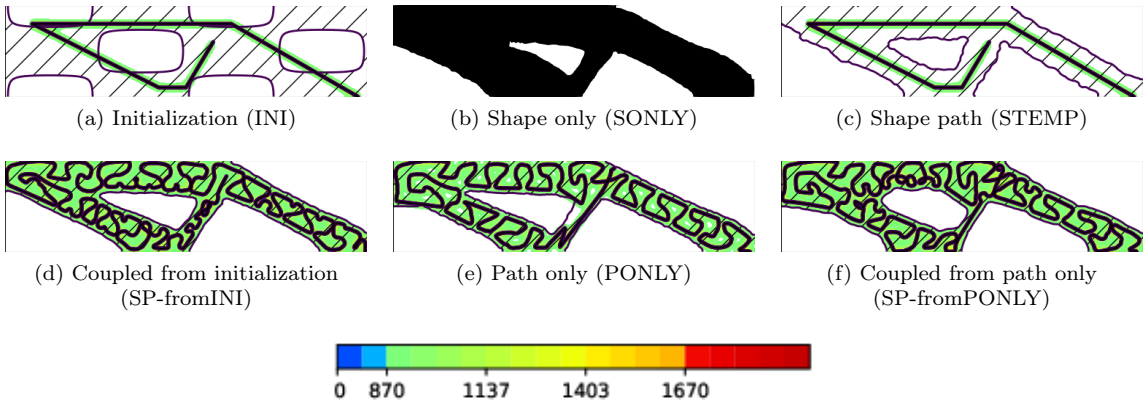
9.4.2 Large cantilever test case

The second test case remains a cantilever yet in bigger working domain which, with the symmetry condition, amounts to $D = [-2.8, 2.8] \times [0, 1.4]$ given in mm. To keep the same accuracy in the source description and the temperature computation, the mesh of this new working domain should be as thin as the previous mesh and thus be composed of 51200 triangular elements. However, with such a number of elements, the computational costs coming from the resolution of the mechanical problem at each outer loop iteration would be far too high. In the inner loop however, the rigidity matrix related to the heat equation remains the same during the whole optimization process and must be inverted only once. We thus use two different meshes: the outer loop and the mechanical computations are run using a mesh with 12800 elements, the inner loop is run with the thin mesh. Then, to allow for the computation of the shape derivatives, the temperature computed on the thin mesh is interpolated (classic interpolation in FreeFem++ [96]) on the outer loop mesh. The mechanical settings remain the same (see Figure 9.2): for both the titanium and aluminium, the Poisson ratio is fixed to $\nu = 0.3$, the Young coefficient is $E = 1$ and the loading $g = (0, -2)$. For both the aluminium and the titanium test cases, two path initializations are tested. In both cases, the initial volume is $V_0 = 4.6289e - 06m^2$.

The interest of this second test case lies in the increasing thickness of the shape bars. Indeed, working on a bigger part, the bars are expected thicker and thus the path longer and adapted to this new thickness.

Aluminium test case

The results for the volume constraint $V_{tar} = 0.9V_0$ are given by Figure 9.13, Table 9.8 and Figure 9.14 for the first initialization and Figure 9.15, Table 9.9 and Figure 9.16 for the second.



(a) Initialization (INI) (b) Shape only (SONLY) (c) Shape path (STEMP)

(d) Coupled from initialization (SP-fromINI) (e) Path only (PONLY) (f) Coupled from path only (SP-fromPONLY)

(g) Colorbar (K)

Figure 9.13: Initialization and shape optimization for different tests, initialization 1, aluminium, large cantilever, $V_{tar} = 0.9V_0$

case	$\frac{V}{V^0}$	C_{ply}	L_{F}	$\overline{C}_{\phi}$	$\overline{C}_{\text{M,D}_\text{S}}$	$\overline{C}_{\text{M,D}\setminus\text{D}_\text{S}}$
INI	1.000	2.266e−5	8.765e−3	8.066e−3	0.000	3.544e−3
SONLY	0.900	9.052e−6	8.765e−3	6.014e−3	0.000	9.194e−4
STEMP	0.900	9.440e−6	8.765e−3	5.453e−3	0.000	0.000
SP-fromINI	0.900	9.006e−6	2.992e−2	7.067e−6	0.000	1.204e−5
PONLY	0.900	9.052e−6	2.630e−2	6.024e−5	0.000	1.279e−4
SP-fromPONLY	0.900	9.369e−6	2.980e−2	5.564e−6	0.000	7.554e−6

Table 9.8: Quantitative results for the different tests, initialization 1, aluminium, large cantilever, $V_{tar} = 0.9V_0$

case	$\frac{V}{V^0}$	C_{ply}	L_{F}	$\overline{C}_{\phi}$	$\overline{C}_{\text{M,D}_\text{S}}$	$\overline{C}_{\text{M,D}\setminus\text{D}_\text{S}}$
INI	1.000	2.266e−5	1.678e−2	5.965e−3	0.000	8.490e−3
SONLY	0.900	9.052e−6	1.678e−2	2.919e−3	0.000	2.718e−3
STEMP	0.900	9.897e−6	1.678e−2	2.029e−3	0.000	8.501e−7
SP-fromINI	0.900	8.987e−6	3.015e−2	8.434e−6	0.000	1.038e−5
PONLY	0.900	9.052e−6	2.586e−2	2.661e−5	0.000	1.808e−5
SP-fromPONLY	0.898	9.310e−6	2.907e−2	6.099e−6	0.000	7.892e−6

Table 9.9: Quantitative results for the different tests, initialization 2, aluminium, large cantilever, $V_{tar} = 0.9V_0$

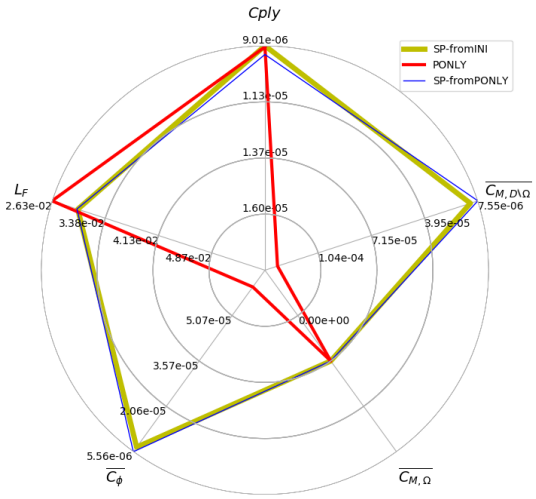
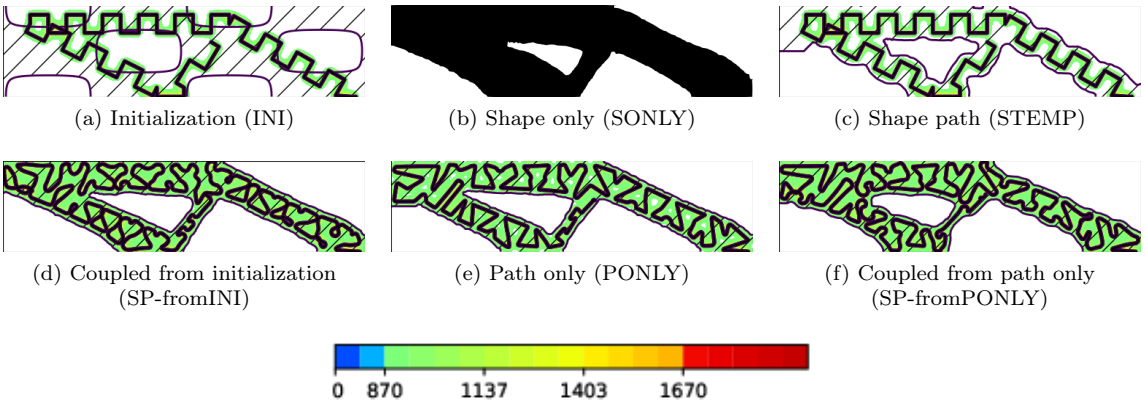


Figure 9.14: Quantitative values visualization, initialization 1, aluminium, large cantilever, $V_{tar} = 0.9V_0$



(a) Initialization (INI)
(b) Shape only (SONLY)
(c) Shape path (STEMP)
(d) Coupled from initialization (SP-fromINI)
(e) Path only (PONLY)
(f) Coupled from path only (SP-fromPONLY)
(g) Colorbar (K)

Figure 9.15: Initialization and shape optimization for different tests, initialization 2, aluminium, large cantilever, $V_{tar} = 0.9V_0$

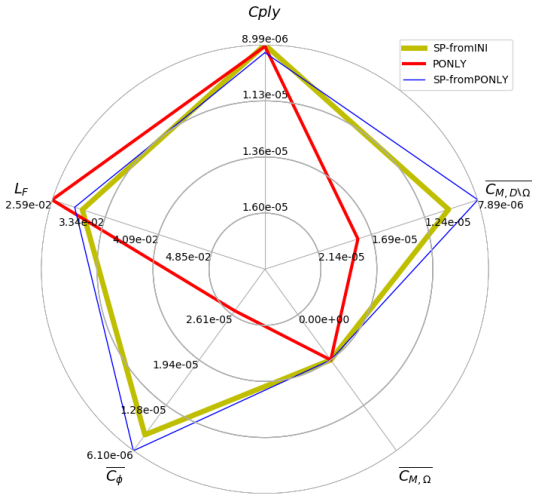
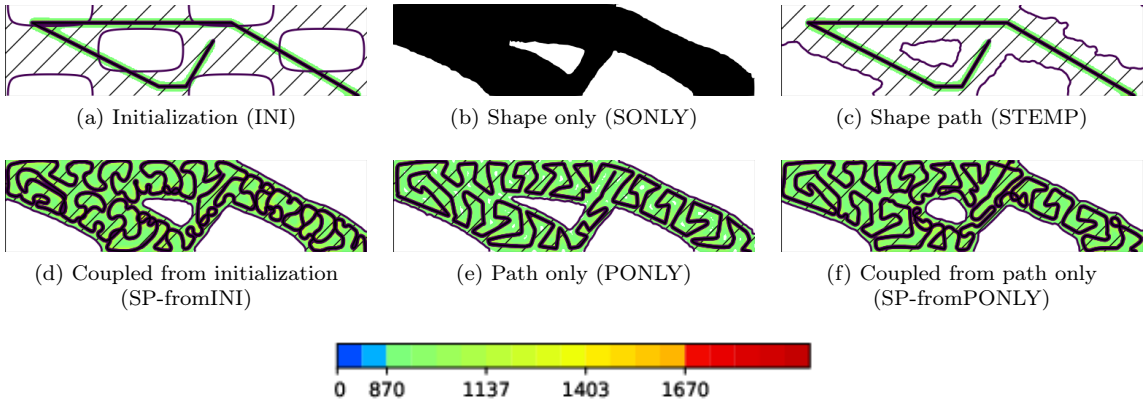


Figure 9.16: Quantitative values visualization, initialization 2, aluminium, large cantilever, $V_{tar} = 0.9V_0$

Increasing the size of the working domain can be intuitively understood as a decrease of the conductivity. Indeed, compared to the working domain, the zone melted by the initialization in Figure 9.13 is thinner than in Figure 9.3. The path and shape must thus adapt more. The results obtained for $V_{tar} = 0.9V_0$ lead to similar conclusions than for the results for the titanium in a small working domain. The bars thickness is once again very important and to cover thick bars, specific patterns appear . In Figure 9.13, the initialization of the path is composed of straight lines. Yet, it seems that once again Omega-patterns are designed (especially in Figure 9.13(d)). The paths resulting from PONLY (Figure 9.13(e)) and concurrent optimization starting from the result of PONLY (Figure 9.13(f)) still present the Omega-pattern especially in the left lower bar. They also seem to introduce a second pattern referred to as "Wave-pattern" in the following. The second initialization (Figure 9.15(a)) is chosen to widen the melted zone at the initial state. It clearly impacts the results by preventing the creation of these Wave-patterns but still deforming the zigzag to introduce once again Omega-patterns or Heart-patterns. In both cases, it is very clear that improvements to a zigzag strategy can be thought of and that the algorithm seem to naturally introduce repetitive patterns which is very interesting for industrial applications and should be further studied. A final observation, a lot clearer in this large domain context than in the classic one, is that adapting the shape improves the results. Completely straightforward in Figure 9.15(c) but also in the different concurrent optimizations and especially Figures 9.13(f) and 9.15(f), the shape boundary fits the melted domain introducing slight oscillations. The damages on the compliance remaining small, this observation confirms the importance of introducing shape constraints to facilitate path designs.

As for $V_{tar} = 1.1V_0$, the results are given by Figure 9.17, Table 9.10 and Figure 9.18 for the first initialization and Figure 9.19, Table 9.11 and Figure 9.20 for the second.



(a) Initialization (INI) (b) Shape only (SONLY) (c) Shape path (STEMP)

(d) Coupled from initialization (SP-fromINI) (e) Path only (PONLY) (f) Coupled from path only (SP-fromPONLY)

(g) Colorbar (K)

Figure 9.17: Initialization and shape optimization for different tests, initialization 1, aluminium, large cantilever, $V_{tar} = 1.1V_0$

case	$\frac{V}{V^0}$	C_{ply}	L_{F}	$\overline{C}_{\phi}$	$\overline{C}_{\text{M,D}_\text{S}}$	$\overline{C}_{\text{M,D}\setminus\text{D}_\text{S}}$
INI	1.000	2.266e−5	8.765e−3	8.066e−3	0.000	3.544e−3
SONLY	1.100	7.666e−6	8.765e−3	6.742e−3	0.000	0.000
STEMP	1.100	8.031e−6	8.765e−3	6.679e−3	0.000	0.000
SP-fromINI	1.100	7.692e−6	3.840e−2	4.410e−6	0.000	5.156e−6
PONLY	1.100	7.666e−6	3.109e−2	1.378e−5	0.000	1.099e−5
SP-fromPONLY	1.101	7.769e−6	3.521e−2	4.306e−6	0.000	4.430e−6

Table 9.10: Quantitative results for the different tests, initialization 1, aluminium, large cantilever, $V_{tar} = 1.1V_0$

With $V_{tar} = 1.1V_0$, the conclusions are quite similar. Since the bars to melt are larger, the patterns are now closer to the Wave-pattern in Figure 9.17(d-e-f) and to hearts in Figure 9.19 (d-e-f). In is once again

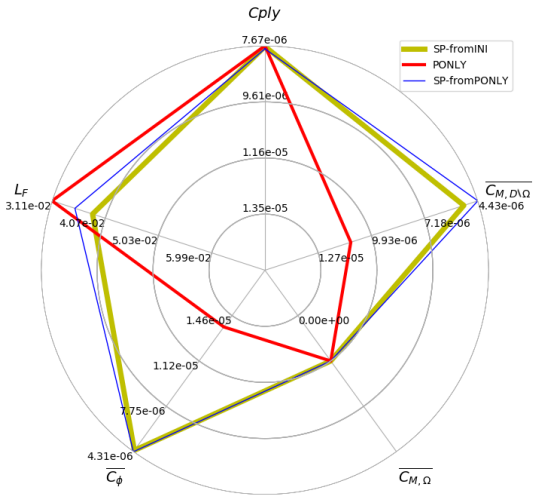
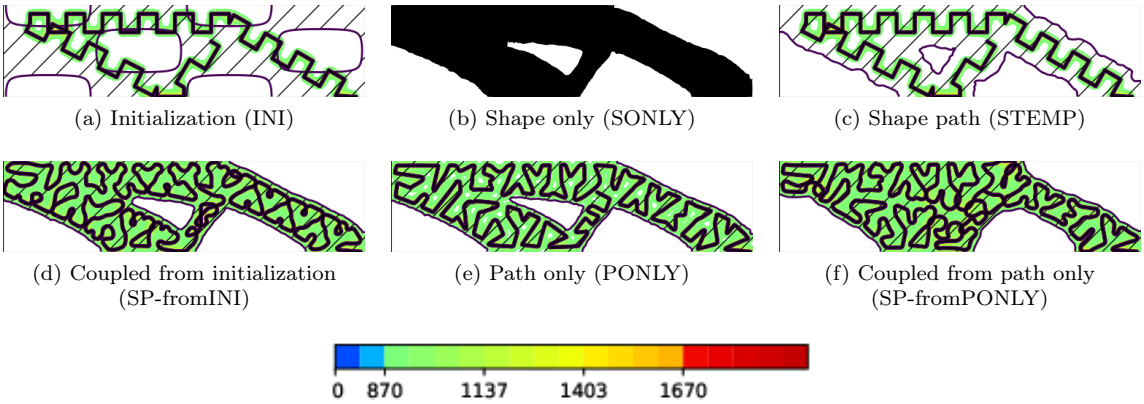


Figure 9.18: Quantitative values visualization, initialization 1, aluminium, large cantilever, $V_{tar} = 1.1V_0$



(a) Initialization (INI) (b) Shape only (SONLY) (c) Shape path (STEMP)

(d) Coupled from initialization (SP-fromINI) (e) Path only (PONLY) (f) Coupled from path only (SP-fromPONLY)

(g) Colorbar (K)

Figure 9.19: Initialization and shape optimization for different tests, initialization 2, aluminium, large cantilever, $V_{tar} = 1.1V_0$

case	$\frac{V}{V^0}$	C_{ply}	L_{F}	$\overline{C}_\phi$	$\overline{C}_{\text{M,D}_\text{S}}$	$\overline{C}_{\text{M,D}\setminus\text{D}_\text{S}}$
INI	1.000	2.266e−5	1.678e−2	5.965e−3	0.000	8.490e−3
SONLY	1.100	7.666e−6	1.678e−2	3.636e−3	0.000	2.901e−4
STEMP	1.100	8.049e−6	1.678e−2	3.422e−3	0.000	0.000
SP-fromINI	1.100	7.667e−6	3.563e−2	4.714e−6	0.000	5.736e−6
PONLY	1.100	7.666e−6	3.091e−2	2.349e−5	0.000	6.289e−6
SP-fromPONLY	1.100	8.041e−6	3.647e−2	2.987e−6	0.000	3.525e−6

Table 9.11: Quantitative results for the different tests, initialization 2, aluminium, large cantilever, $V_{tar} = 1.1V_0$

very clear that the shape adapt to the melted zones with the modification of the inner hole size in Figures 9.17 and 9.19 (d) and (f) (the hole even disappears in Figure 9.19(f)) and with stronger oscillations in the boundary, especially when running concurrent optimization starting from PONLY (Figures 9.17 and 9.19(f)).

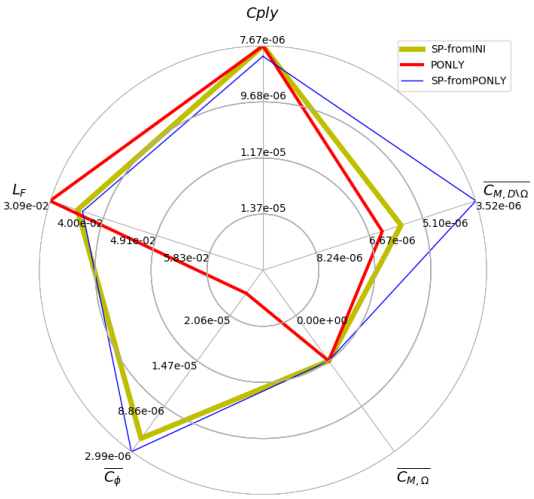
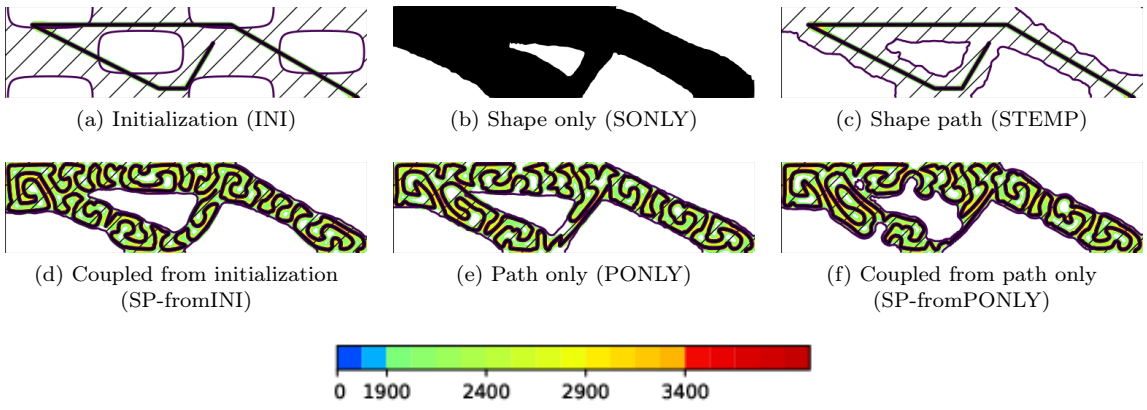


Figure 9.20: Quantitative values visualization, initialization 2, aluminium, large cantilever, $V_{tar} = 1.1V_0$

Titanium test case

To further complicate the optimization, these tests with a larger domain are now run with the titanium. The Figures presenting the results for $V_{tar} = 0.9V_0$ are given by Figure 9.21, Table 9.12 and Figure 9.22 for the first initialization and Figure 9.23, Table 9.13 and Figure 9.24for the second initialization. As for the tests for $V_{tar} = 1.1V_0$, they are given by Figure 9.25, Table 9.14 and Figure 9.26 for the first initialization and Figure 9.27, Table 9.15 and Figure 9.28 for the second initialization.



(a) Initialization (INI)
(b) Shape only (SONLY)
(c) Shape path (STEMP)
(d) Coupled from initialization (SP-fromINI)
(e) Path only (PONLY)
(f) Coupled from path only (SP-fromPONLY)
(g) Colorbar (K)

Figure 9.21: Initialization and shape optimization for different tests, initialization 1, titanium, large cantilever, $V_{tar} = 0.9V_0$

Filling the volume is very hard for the path and the results hint between a relation between the filling pattern and the thickness of the bar. In Figure 9.21(d-e), it seems that thin bars are filled in with a Omega-pattern whereas thicker bars at the upper left corner are filled in with a strong Wave-pattern. On Figure 9.21(f), it also seems that the algorithm can take advantage of the low conductivity to build bars that only require a straight line to be melted. Very thin, they still maintain a correct compliance. These observations can be spread to Figures 9.25. As for zigzag initializations, they lead to slightly different patterns corresponding to a mix between Omega- and Heart-patterns. Yet, the relation between thickness and pattern happens in these tests too with the presence of Wave-patterns especially in the top left corners of Figures 9.23(d-e-f) and 9.27(d-e-f). As in the small working domain, the low conductivity of the titanium emphasizes the adaptation of the shape to the temperature. At the cost of compliance, this is very well illustrated by Figure 9.21(f) in which the inner whole corresponds to a non intuitive shape fitting the path. If this is less explicit in the other results, the algorithm still created boundary oscillations.

case	$\frac{V}{V^0}$	C_{ply}	L_{F}	$\overline{C}_{\phi}$	$\overline{C}_{\text{M,D}_\text{S}}$	$\overline{C}_{\text{M,D}\setminus\text{D}_\text{S}}$
INI	1.000	2.266e−5	8.765e−3	8.066e−3	0.000	3.544e−3
SONLY	0.900	9.052e−6	8.765e−3	6.014e−3	0.000	9.194e−4
STEMP	0.900	9.440e−6	8.765e−3	5.453e−3	0.000	0.000
SP-fromINI	0.900	9.006e−6	2.992e−2	7.067e−6	0.000	1.204e−5
PONLY	0.900	9.052e−6	2.630e−2	6.024e−5	0.000	1.279e−4
SP-fromPONLY	0.900	9.369e−6	2.980e−2	5.564e−6	0.000	7.554e−6

Table 9.12: Quantitative results for the different tests, initialization 1, titanium, large cantilever, $V_{tar} = 0.9V_0$

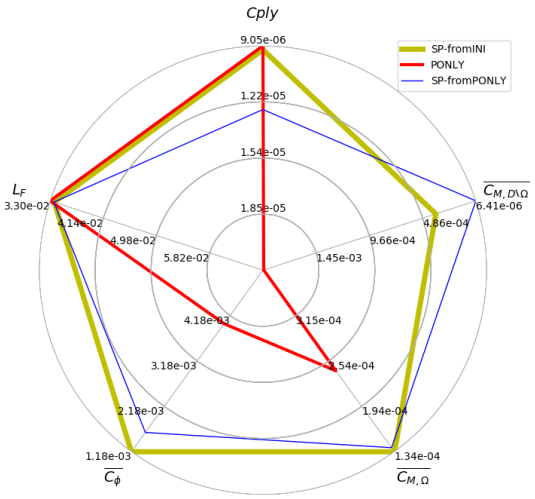
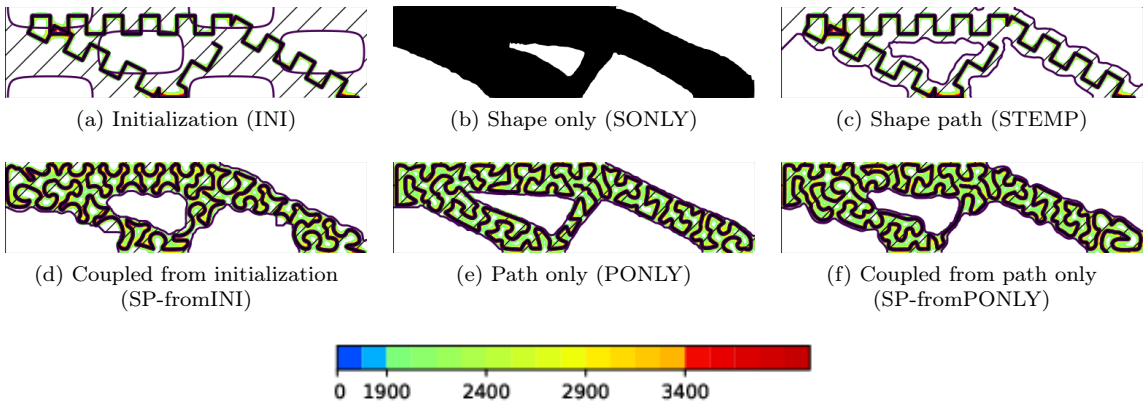


Figure 9.22: Quantitative values visualization, initialization 1, titanium, large cantilever, $V_{tar} = 0.9V_0$



(a) Initialization (INI)
(b) Shape only (SONLY)
(c) Shape path (STEMP)
(d) Coupled from initialization (SP-fromINI)
(e) Path only (PONLY)
(f) Coupled from path only (SP-fromPONLY)
(g) Colorbar (K)

Figure 9.23: Initialization and shape optimization for different tests, initialization 2, titanium, large cantilever, $V_{tar} = 0.9V_0$

9.5 Conclusion

In this chapter, we have adapted the path optimization algorithm developed in this work to handle concurrent optimization of path and design in the steady state. The resulting concurrent algorithm is quite basic: it relies on alternate optimization of the shape and the path at each iteration. Yet, this simple algorithm has interesting features and yields promising results. First of all, the algorithm allows

case	$\frac{V}{V^0}$	C_{ply}	L_{F}	$\overline{C}_{\phi}$	$\overline{C}_{\text{M,D}_\text{S}}$	$\overline{C}_{\text{M,D}\setminus\text{D}_\text{S}}$
INI	1.000	2.266e−5	1.678e−2	5.965e−3	0.000	8.490e−3
SONLY	0.900	9.052e−6	1.678e−2	2.919e−3	0.000	2.718e−3
STEMP	0.900	9.897e−6	1.678e−2	2.029e−3	0.000	8.501e−7
SP-fromINI	0.900	8.987e−6	3.015e−2	8.434e−6	0.000	1.038e−5
PONLY	0.900	9.052e−6	2.586e−2	2.661e−5	0.000	1.808e−5
SP-fromPONLY	0.898	9.310e−6	2.907e−2	6.099e−6	0.000	7.892e−6

Table 9.13: Quantitative results for the different tests, initialization 2, titanium, large cantilever, $V_{tar} = 0.9V_0$

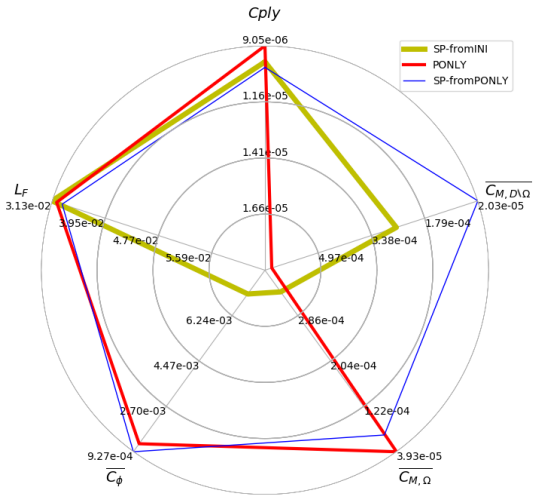
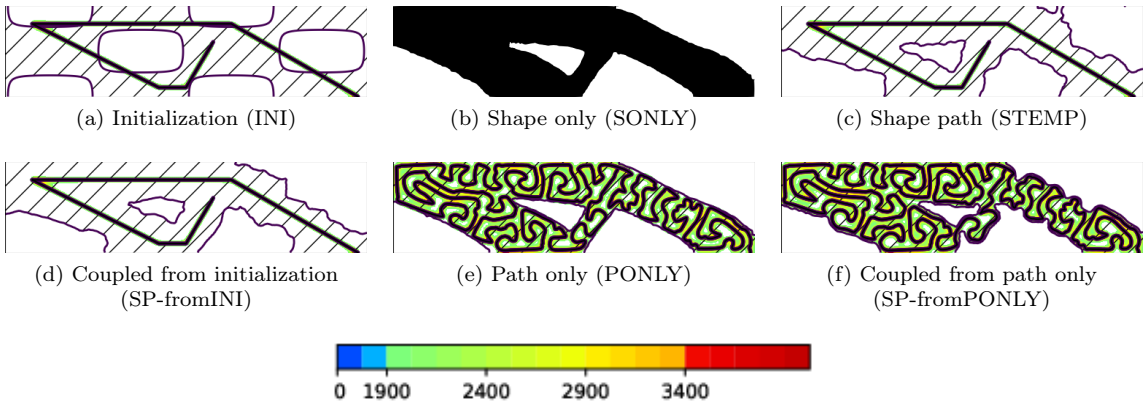


Figure 9.24: Quantitative values visualization, initialization 2, titanium, large cantilever, $V_{tar} = 0.9V_0$



(a) Initialization (INI)
(b) Shape only (SONLY)
(c) Shape path (STEMP)
(d) Coupled from initialization (SP-fromINI)
(e) Path only (PONLY)
(f) Coupled from path only (SP-fromPONLY)
(g) Colorbar (K)

Figure 9.25: Initialization and shape optimization for different tests, initialization 1, titanium, large cantilever, $V_{tar} = 1.1V_0$

for a modification of the part design to facilitate the scanning, which in our case is mainly visible on the bars thickness. Indeed, depending on the material's properties and on the scanning parameters, a specific width of melted powder can be characterized. This width must be related to the bars thickness: too thin, the bar cannot be accurately built and too large, it might be slimmed to prevent any ill-melted powder. The second main feature influencing the algorithm is the initialization, which now influences

case	$\frac{V}{V^0}$	C_{ply}	L_{F}	$\overline{C}_{\phi}$	$\overline{C}_{\text{M,D}_\text{S}}$	$\overline{C}_{\text{M,D}\setminus\text{D}_\text{S}}$
INI	1.000	2.266e−5	8.765e−3	8.066e−3	0.000	3.544e−3
SONLY	1.100	7.666e−6	8.765e−3	6.742e−3	0.000	0.000
STEMP	1.100	8.031e−6	8.765e−3	6.679e−3	0.000	0.000
SP-fromINI	1.100	7.692e−6	3.840e−2	4.410e−6	0.000	5.156e−6
PONLY	1.100	7.666e−6	3.109e−2	1.378e−5	0.000	1.099e−5
SP-fromPONLY	1.101	7.769e−6	3.521e−2	4.306e−6	0.000	4.430e−6

Table 9.14: Quantitative results for the different tests, initialization 1, titanium, large cantilever, $V_{tar} = 1.1V_0$

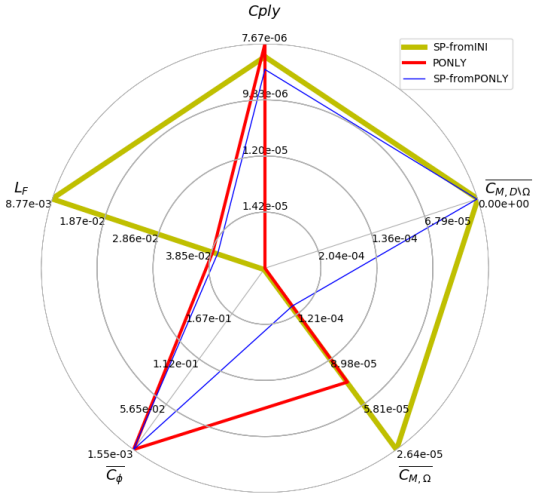
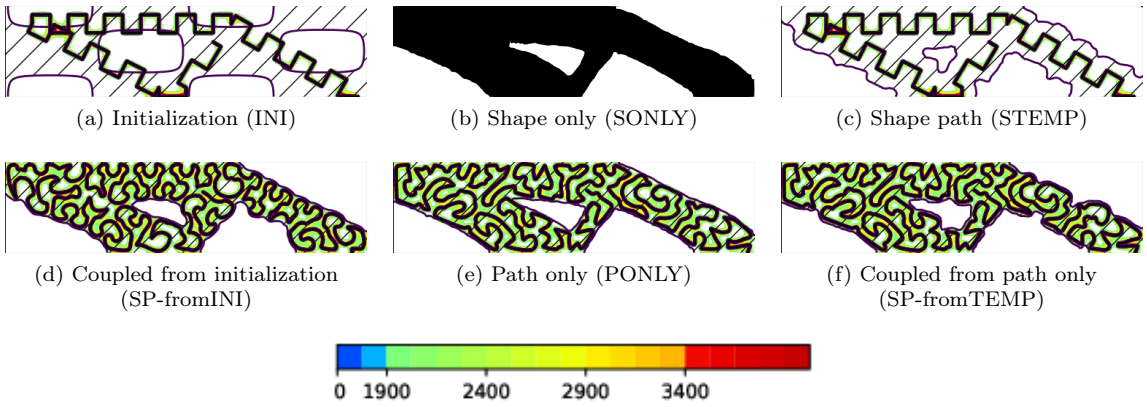


Figure 9.26: Quantitative values visualization, initialization 1, titanium, large cantilever, $V_{tar} = 1.1V_0$



(a) Initialization (INI) (b) Shape only (SONLY) (c) Shape path (STEMP)

(d) Coupled from initialization (SP-fromINI) (e) Path only (PONLY) (f) Coupled from path only (SP-fromTEMP)

(g) Colorbar (K)

Figure 9.27: Initialization and shape optimization for different tests, initialization 2, titanium, large cantilever, $V_{tar} = 1.1V_0$

the part design itself instead of just the scanning path as in the previous chapters. This feature must be further assessed. Indeed, from the preliminary results obtained, it seems that no easy solution exists and that algorithms should be worked on to allow for a computationally efficient initialization algorithm.

These promising results open up perspectives. First of all, the problem could be considered with other

case	$\frac{V}{V^0}$	C_{ply}	L_{F}	$\overline{C}_{\phi}$	$\overline{C}_{\text{M,D}_\text{S}}$	$\overline{C}_{\text{M,D}\setminus\text{D}_\text{S}}$
INI	1.000	2.266e−5	1.678e−2	5.965e−3	0.000	8.490e−3
SONLY	1.100	7.666e−6	1.678e−2	3.636e−3	0.000	2.901e−4
STEMP	1.100	8.049e−6	1.678e−2	3.422e−3	0.000	0.000
SP-fromINI	1.100	7.667e−6	3.563e−2	4.714e−6	0.000	5.736e−6
PONLY	1.100	7.666e−6	3.091e−2	2.349e−5	0.000	6.289e−6
SP-fromPONLY	1.100	8.041e−6	3.647e−2	2.987e−6	0.000	3.525e−6

Table 9.15: Quantitative results for the different tests, initialization 2, titanium, large cantilever, $V_{tar} = 1.1V_0$

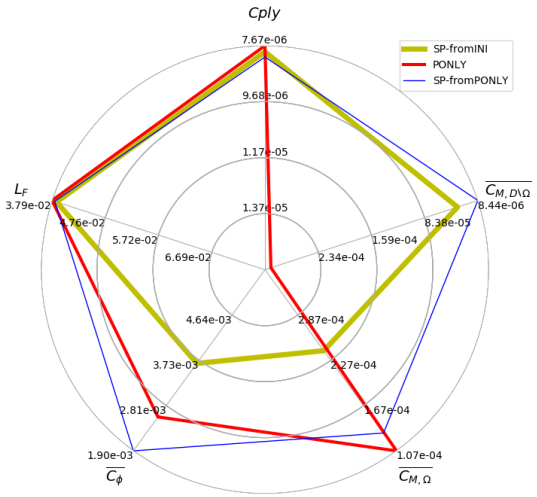


Figure 9.28: Quantitative values visualization, initialization 2, titanium, large cantilever, $V_{tar} = 1.1V_0$

optimization algorithms. Indeed, finding the perfect balance between the many constraints is complicated and replacing the Augmented Lagrangian method by the null space gradient one could increase the accuracy of this balance. Then, an in-depth study of the impact of thickness on the scanning path should be conducted to answer the following questions: is there an optimal bar thickness? Should it belong to a specific range? Do the good thicknesses constitute a discrete set (for example if "good" thicknesses were a multiple of a specific thickness)? Further tests should also be run to test the impact of the object boundary's curvature on the scanning design. Then, as mentioned, a deeper consideration of the initialization might lead to the development of a systematic technique: further tests could be run or a scanning of the contour could be included to the paths to copy the industrial process. Finally, this algorithm should be run with a more complex model and, to start with, using the transient algorithm detailed in Chapter 7. No specific difficulties should arise of this adaptation, but for the extra computational time required for path optimization. It would also be very interesting to optimize the compliance under temperature and volume constraints without any level set anymore but through the path only, considering that mechanical stresses arise in the melted part only.

CONCLUSION AND PERSPECTIVES

We argue in this thesis that parts conception can include criteria to ease each layer's scanning, which is part of the "Design for Additive Manufacturing" approach. This requires the definition of design constraints representative of the generated path's quality for the part. It however appears that the notion of "good path" is complicated, still under study and almost never related in the literature to the geometry to build. We have thus decided to develop a mathematical optimization of the scanning path: without any a priori fixed scanning pattern, the results from such an algorithm increase the intuition on the "good path" notion, propose new scanning solutions and link the path results to the geometry to build.

To do so, we have first develop a simulation model and set an optimization problem. Since the literature focusing on systematic path optimization is rare and recent (only [6, 7, 51]), assumptions have been made to simplify the simulations. We decided to focus on conduction effects only, in two dimensions. In the corresponding heat equation, the material characteristics are constant and the source is modeled by a Gaussian beam traveling along the path at constant speed. The mechanical and thermal phenomena are controlled by bounds on the temperatures. A second steady state model has also been developed to further simplify the problem (the steady state assumption is also present in [51]). Both models have been calibrated to fit data temperatures for the transient one and maximum temperature during the scanning for the steady state. The adaptation of these models to numerical computations has also required choices. Base on a front tracking method, the discrete path is a broken line which characteristic element size is controlled.
We have first developed a path optimization algorithm in the steady state context. Using shape differentiation theory, descent directions, corresponding to different first order optimization algorithms have been computed. Besides the usefulness of the numerical results to analyze the discretization, algorithmic and even modeling choices made all along this work, these results appeared interesting and promising. For example, the results obtained in Figure 6.2 corroborate existing literature [77] and could be used in industrial applications, whereas others which have not been studied yet could be experimentally tested. Moreover, they point out the interest of further considering the volumetric energy, already defined in experimental and simulation works (see [43, 48, 83] and references therein).
We then adapted the algorithms to the transient model giving up shape optimization in aid for control based techniques. If the first tests in which the optimization variables were the path tangent direction, the first path point and the final time, were not conclusive and should be further worked on, the point based optimization gave interesting results. In addition to a rapid analysis of the discretization and algorithmic choices, we have tested several initializations to build different geometries. In the aluminium test case (which can be extended for high conductivity materials), the number of local minima seemed quite restrained with some patterns appearing several times. These patterns should be experimentally tested and analyzed to further understand in what way their characteristics are optimal. As for the titanium, the results do not provide perfect patterns but the path adapts well to the geometries which is very encouraging for future works. Once again, these results confirm that a very interesting perspective could be to focus on volumetrical energy.
In the steady state case, we have added the number of connected components as an optimization feature. A first technique, following discussions with Tonia Maria ALAM and Serge NICAISE (from the LAMAV, Université Polytechnique des Hauts de France), consists in adding a power variable to the optimization variables: this variable is 1 when the laser in on and 0 else. We then used a relaxation-penalization algorithm as well as a total variation constraint on the power variable. The numerical results are in agreement with the literature [191, 192] and are promising for further work, starting from an adaptation to the transient model to then include the velocity in the optimization. A second method, based on topological gradient notions, has also been set: topological gradients have been computed to determine where cutting the path and where adding a new connected component. This method fixes the maximum number of connected components as an optimization parameter facilitating the control of the source jumps. Adapting this method to the transient case is a natural perspective that will require further investigations in optimal control techniques.
Finally, again in the steady state case, a concurrent optimization of the path and the shape has been set. Involving mechanical and thermal effects, the objective was to minimize the shape's compliance and the path length while satisfying volume and temperature constraints. The results are preliminary and the double loop algorithm set could be improved. However, they already point out very interesting features. First of all, the algorithm adapts the shape to the heat problem and it is very clear that the design is modified when taking the path into account. Then, the notion of volumetric energy once again appears.

However, it is now a localized notion. Indeed, the temperature constraints impact the thickness of the structure and sometimes force the boundary to oscillate to adapt to the melting.

A step back on the results obtained open up further work. First of all, the final time of a PhD is bounded and the work asks for further developments. Among them, some technical issues must be dealt with. Industrial applications involve a hierarchy in the temperature constraints: respecting the geometry is more important than mechanical defects: the phase and maximum temperature constraint out of the domain, determining the built shape, must be satisfied at the cost of the maximum temperature in the domain, related to the mechanical defects, that can remain approached at convergence. To do so, we could differently choose the Lagrange multipliers, we could further adapt the null space gradient algorithm or we could think of different constraints (for example by working on the formulation of a projection on the geometry to build for the maximum temperature out of the domain constraint). This algorithmic study could be extended to the concurrent optimization in which the volume constraint should never be violated. A second issue consists in developing more the transient model. Indeed, the power optimization work, the topological optimization work and the concurrent optimization for now only apply to the steady state model. The results being already interesting and encouraging, numerical applications in the transient case could be even more informative. Further studying optimal control literature could also help improving the current algorithm and speed up the optimization.

Beyond these direct work continuation ideas, the results obtained also raise several perspectives. Among the short term ones, it would be relevant to compare the different path discretization possibilities. We chose in this work a broken line representation to remain close to the source movement. However, we could also use splines (as in [6]) inducing smoother trajectories or level sets (as in [51]). Comparisons between the different approaches would help understanding their advantages and drawbacks and allow for informed representation choices.

The model could then be made more complex. The first feature to add, and maybe the most crucial, consists in involving the kinematics into the path considerations: in this work, the velocity is considered constant on the whole path, which is of course a simplification. The source movements depend on the path geometry and especially on the curvature: the higher this curvature is, the slower goes the path and, if the power cannot be regulated, the higher locally is the temperature. To first include the velocity without modifying the simulations, the curvature could be included in the optimization: instead of optimizing the path length (in the steady state case) or the final time (in the transient case), we could minimize the integral of the curvature on the path. This could be supplemented by curvature constraints that might require replacing the broken line path discretization by splines. The model could also be modified by choosing a non constant velocity. The actual heat equation simulation with the point based path discretization could easily be modified to allow such simulations. We could then model the velocity depending on the kinematics constraints or add it as an optimization variable. Among the many possibilities, we could imagine directly expressing the velocity as a function of the curvature resulting in, in broken lines, attributing a velocity to each segment depending on the angle between this segment and the previous one [76].

To be confirmed by tests including kinematics but already pointed out by this work, a second promising lead consists in studying the volumetric energy. This notion has been defined by experimental and numerical studies (see [43, 48, 83] and references therein) and gathers the kinematics, the source and material properties to sum it up into a single parameter corresponding to an amount of energy passed to the powder. Such a notion seems to appear in many results: in the steady and transient cases with constant final length and time whatever path initialization is chosen, in the concurrent optimization with the structure thickness and boundary locally impacted by the temperature. An energy could be related to the geometry first at a global scale (energy for the part) and then at a local scale with the elaboration of a "thermal skeleton". Then, the path could be optimized at constant energy. This could also make already existing approaches available for this application (further considering [64, 65] would for example be very interesting).
Then of course, the limitations related to the physical model must be relaxed: the temperature constraints could be made more realistic by for example taking into account temperature gradients, the material coefficients could depend on space and time at first by simply depending on the phase and then depending on the temperature (which will require their differentiation in the optimization problems), convection and radiation could be included. Going back to a three dimensional model would also helps the physical

interpretation of the results with an analysis of the temperature effects on the vertical axis too and an understanding of the optimal scanning paths evolution from one layer to another. Finally, a mechanical model could be introduced with a full thermo-mechanical resolution or alternative methods such as inherent strain for example [51, 124].

BIBLIOGRAPHY

[1] *Addup*. https://www.addupsolutions.com/en/machines-and-lines/.

[2] *Sofia*. https://www.sofia-3d.fr/.

[3] R. Acharya, J. A. Sharon, and A. Staroselsky, *Prediction of microstructure in laser powder bed fusion process*, Acta Materialia, 124 (2017), pp. 360–371.

[4] P. Ahangar, M. E. Cooke, M. H. Weber, and D. H. Rosenzweig, *Current biomedical applications of 3d printing and additive manufacturing*, Applied Sciences, 9 (2019), p. 1713.

[5] A. Ahrari, K. Deb, S. Mohanty, and J. H. Hattel, *Multi-objective optimization of cellular scanning strategy in selective laser melting*, in 2017 IEEE Congress on Evolutionary Computation (CEC), IEEE, 2017, pp. 2730–2737.

[6] T. M. Alam, *Some optimal control problem of partial differential equations and applications to the selective laser melting process (SLM)*, PhD thesis, Université Polytechnique Hauts-de-France, INSA Hauts-De-France, 2020.

[7] T. M. Alam, S. Nicaise, and L. Paquet, *An optimal control problem governed by the heat equation with nonconvex constraints applied to the selective laser melting process*, Minimax Theory and its Applications, 6 (2021).

[8] G. Allaire, *Shape optimization by the homogenization method*, vol. 146 of Applied Mathematical Sciences, Springer-Verlag, New York, 2002.

[9] ——, *Conception optimale de structures*, Mathématiques & Applications (Berlin) [Mathematics & Applications], Springer-Verlag, Berlin, 2007.

[10] ——, *Numerical analysis and optimization*, Numerical Mathematics and Scientific Computation, Oxford University Press, Oxford, 2007.

[11] G. Allaire, M. Bihr, and B. Bogosel, *Support optimization in additive manufacturing for geometric and thermo-mechanical constraints*, Struct. Multidiscip. Optim., 61 (2020), pp. 2377–2399.

[12] G. Allaire and B. Bogosel, *Optimizing supports for additive manufacturing*, Struct. Multidiscip. Optim., 58 (2018), pp. 2493–2515.

[13] G. Allaire, C. Dapogny, R. Estevez, A. Faure, and G. Michailidis, *Structural optimization under overhang constraints imposed by additive manufacturing technologies*, J. Comput. Phys., 351 (2017), pp. 295–328.

[14] G. Allaire, C. Dapogny, A. Faure, and G. Michailidis, *Shape optimization of a layer by layer mechanical constraint for additive manufacturing*, C. R. Math. Acad. Sci. Paris, 355 (2017), pp. 699–717.

[15] G. Allaire, C. Dapogny, and F. Jouve, *Shape and topology optimization*, to appear in Handbook of Numerical Analysis 22 – Geometric PDES (preprint hal-02496063), (2020).

[16] G. Allaire and L. Jakabčin, *Taking into account thermal residual stresses in topology optimization of structures built by additive manufacturing*, Math. Models Methods Appl. Sci., 28 (2018), pp. 2313–2366.

[17] G. Allaire, F. Jouve, and A.-M. Toader, *Structural optimization using sensitivity analysis and a level-set method*, J. Comput. Phys., 194 (2004), pp. 363–393.

[18] S. Amstutz, *Convex optimization and optimal control*, Lecture notes, Ecole Polytechnique, (2020).

[19] I. E. Anderson, E. M. White, and R. Dehoff, *Feedstock powder processing research needs for additive manufacturing development*, Current Opinion in Solid State and Materials Science, 22 (2018), pp. 8–15.

[20] E. M. Arkin, M. Held, and C. L. Smith, *Optimization problems related to zigzag pocket machining*, Algorithmica, 26 (2000), pp. 197–236.

[21] R. C. Aster, B. Borchers, and C. H. Thurber, *Parameter estimation and inverse problems*, Elsevier/Academic Press, Amsterdam, second ed., 2013.

[22] D. Auroux and M. Masmoudi, *Image processing by topological asymptotic expansion*, J. Math. Imaging Vision, 33 (2009), pp. 122–134.

[23] D. Axinte and J. Billingham, *Time-dependent manufacturing processes lead to a new class of inverse problems*, Proceedings of the National Academy of Sciences, 116 (2019), pp. 5341–5343.

[24] A. Barbero and S. Sra, *Fast newton-type methods for total variation regularization*, in Proceedings of the 28th International Conference on Machine Learning (ICML-11), Citeseer, 2011, pp. 313–320.

[25] ——, *Modular proximal optimization for multidimensional total-variation regularization*, The Journal of Machine Learning Research, 19 (2018), pp. 2232–2313.

[26] R. d. S. Barros, *Laser powder bed fusion of INCONEL® 718: optimization of process parameters and residual stress analysis before and after heat treatment*, PhD thesis, Instituto Superior de Engenharia do Porto, 2019.

[27] J. L. Bartlett and X. Li, *An overview of residual stresses in metal powder bed fusion*, Additive Manufacturing, 27 (2019), pp. 131–149.

[28] K. Bartsch, D. Herzog, C. Emmelmann, and F. Lange, *A novel approach to support structures optimized for heat dissipation in slm by combining process simulation with topology optimization*, in Proceedings of the NAFEMS World Congress, 2019, p. 25.

[29] I. Baturynska, O. Semeniuta, and K. Martinsen, *Optimization of process parameters for powder bed fusion additive manufacturing by combination of machine learning and finite element method: A conceptual framework*, Procedia CIRP, 67 (2018), pp. 227–232.

[30] T. Bayen, *Cours d'optimisation numérique.*, Lecture notes, Université Montpellier 2, (2017).

[31] C. Baykasoglu, O. Akyildiz, D. Candemir, Q. Yang, and A. C. To, *Predicting microstructure evolution during directed energy deposition additive manufacturing of ti-6al-4v*, Journal of Manufacturing Science and Engineering, 140 (2018).

[32] M. Bendsøe and O. Sigmund, *Topology optimization*, Springer-Verlag, Berlin, 2003. Theory, methods and applications.

[33] B. Berman, *3-d printing: The new industrial revolution*, Business horizons, 55 (2012), pp. 155–162.

[34] M. Bertoldi, M. Yardimci, C. Pistor, and S. Guceri, *Domain decomposition and space filling curves in toolpath planning and generation*, in 1998 International Solid Freeform Fabrication Symposium, 1998.

[35] V. Bhavar, P. Kattire, V. Patil, S. Khot, K. Gujar, and R. Singh, *A review on powder bed fusion technology of metal additive manufacturing*, in Additive Manufacturing Handbook, CRC Press, 2017, pp. 251–253.

[36] M. Bihr, *Join topology optimisation of part and support for metal additive manufacturing*, PhD thesis, Ecole Polytechnique, to appear in 2021.

[37] A. Bobenko, *Geometry ii: Discrete differential geometry*, 2015.

[38] M. Boissier, J. Deaton, P. Beran, and N. Vermaak, *Elastoplastic topology optimization and cyclically loaded structures via direct methods for shakedown.* HAL preprint hal-02935509, 2020.

[39] M. Bouard, V. Pateloup, and P. Armand, *Pocketing toolpath computation using an optimization method*, Computer-Aided Design, 43 (2011), pp. 1099–1109.

[40] S. E. Brika, Y. F. Zhao, M. Brochu, and J. Mezzetta, *Multi-objective build orientation optimization for powder bed fusion by laser*, Journal of Manufacturing Science and Engineering, 139 (2017).

[41] C. Bui, C. Dapogny, and P. Frey, *An accurate anisotropic adaptation method for solving the level set advection equation*, Internat. J. Numer. Methods Fluids, 70 (2012), pp. 899–922.

[42] M. Burger, *A framework for the construction of level set methods for shape optimization and reconstruction*, Interfaces Free Bound., 5 (2003), pp. 301–329.

[43] F. Caiazzo, V. Alfieri, and G. Casalino, *On the relevance of volumetric energy density in the investigation of inconel 718 laser powder bed fusion*, Materials, 13 (2020), p. 538.

[44] F. Calignano and P. Minetola, *Influence of process parameters on the porosity, accuracy, roughness, and support structures of hastelloy x produced by laser powder bed fusion*, Materials, 12 (2019), p. 3178.

[45] S. Catchpole-Smith, N. Aboulkhair, L. Parry, C. Tuck, I. Ashcroft, and A. Clare, *Fractal scan strategies for selective laser melting of 'unweldable'nickel superalloys*, Additive Manufacturing, 15 (2017), pp. 113–122.

[46] J. Céa, *Conception optimale ou identification de formes: calcul rapide de la dérivée directionnelle de la fonction coût*, RAIRO Modél. Math. Anal. Numér., 20 (1986).

[47] J. Céa, S. Garreau, P. Guillaume, and M. Masmoudi, *The shape and topological optimizations connection*, vol. 188, 2000. IV WCCM, Part II (Buenos Aires, 1998).

[48] C. Cepeda-Jiménez, F. Potenza, E. Magalini, V. Luchin, A. Molinari, and M. Pérez-Prado, *Effect of energy density on the microstructure and texture evolution of ti-6al-4v manufactured by laser powder bed fusion*, Materials Characterization, (2020), p. 110238.

[49] A. Chambolle, V. Caselles, D. Cremers, M. Novaga, and T. Pock, *An introduction to total variation for image analysis*, Theoretical foundations and numerical methods for sparse recovery, 9 (2010), p. 227.

[50] Q. Chen, X. Liang, D. Hayduke, J. Liu, L. Cheng, J. Oskin, R. Whitmore, and A. C. To, *An inherent strain based multiscale modeling framework for simulating part-scale residual deformation for direct metal laser sintering*, Additive Manufacturing, 28 (2019), pp. 406–418.

[51] Q. Chen, J. Liu, X. Liang, and A. C. To, *A level-set based continuous scanning path optimization method for reducing residual stress and deformation in metal additive manufacturing*, Computer Methods in Applied Mechanics and Engineering, 360 (2020), p. 112719.

[52] B. Cheng, S. Shrestha, and K. Chou, *Stress and deformation evaluations of scanning strategy effect in selective laser melting*, Additive Manufacturing, 12 (2016), pp. 240–251.

[53] L. Cheng, X. Liang, J. Bai, Q. Chen, J. Lemon, and A. To, *On utilizing topology optimization to design support structure to prevent residual stress induced build failure in laser powder bed metal additive manufacturing*, Additive Manufacturing, 27 (2019), pp. 290–304.

[54] J. Chetboun, *Conception de formes aérodynamiques en présence d'écoulements décollés : contrôle et optimisation*, PhD thesis, Ecole Polytechnique, 2010.

[55] J. Chetboun and G. Allaire, *Flow control of curved air ducts using topological derivatives*, in Proceedings of the 8th World Congress on Structural and Multidisciplinary Optimization, H.C. Rodrigues et al. eds., ISSMO, 2009.

[56] K. Chudej, H. J. Pesch, M. Wächter, G. Sachs, and F. Le Bras, *Instationary heat-constrained trajectory optimization of a hypersonic space vehicle by ODE-PDE-constrained optimal control*, in Variational analysis and aerospace engineering, vol. 33 of Springer Optim. Appl., Springer, New York, 2009, pp. 127–144.

[57] A. Clausen, *Topology Optimization for Additive Manufacturing*, PhD thesis, Technical University of Denmark, 2016.

[58] G. Cohen, *Convexité et optimisation*, Lecture notes, Ecole Nationale des Ponts et Chaussées et INRIA, (2006).

[59] P. S. Cook and A. B. Murphy, *Simulation of melt pool behaviour during additive manufacturing: Underlying physics and progress*, Additive Manufacturing, 31 (2020), p. 100909.

[60] L. E. Criales, Y. M. Arisoy, B. Lane, S. Moylan, A. Donmez, and T. Özel, *Predictive modeling and optimization of multi-track processing for laser powder bed fusion of nickel alloy 625*, Additive Manufacturing, 13 (2017), pp. 14–36.

[61] D. Dai and D. Gu, *Tailoring surface quality through mass and momentum transfer modeling using a volume of fluid method in selective laser melting of tic/alsi10mg powder*, International Journal of Machine Tools and Manufacture, 88 (2015), pp. 95–107.

[62] K. Dai and L. Shaw, *Distortion minimization of laser-processed components through control of laser scanning patterns*, Rapid Prototyping Journal, 8 (2002), pp. 270–276.

[63] C. Dapogny, *Shape optimization, level set methods on unstructured meshes and mesh evolution*, PhD thesis, Université Pierre et Marie Curie, 2013.

[64] ——, *A connection between topological ligaments in shape optimization and thin tubular inhomogeneities*, Comptes Rendus. Mathématique, 358 (2020), pp. 119–127.

[65] ——, *The topological ligament in shape optimization: a connection with thin tubular inhomogeneities*. HAL preprint hal-02924929, 2020.

[66] C. Dapogny, R. Estevez, A. Faure, and G. Michailidis, *Shape and topology optimization considering anisotropic features induced by additive manufacturing processes*, Comput. Methods Appl. Mech. Engrg., 344 (2019), pp. 626–665.

[67] F. de Gournay, *Velocity extension for the level-set method and multiple eigenvalues in shape optimization*, SIAM J. Control Optim., 45 (2006), pp. 343–367.

[68] T. DebRoy, H. Wei, J. Zuback, T. Mukherjee, J. Elmer, J. Milewski, A. M. Beese, A. Wilson-Heid, A. De, and W. Zhang, *Additive manufacturing of metallic components–process, structure and properties*, Progress in Materials Science, 92 (2018), pp. 112–224.

[69] D. Delgado Camacho, P. Clayton, W. J. O'Brien, C. Seepersad, M. Juenger, R. Ferron, and S. Salamone, *Applications of additive manufacturing in the construction industry–a forward-looking review*, Automation in construction, 89 (2018), pp. 110–119.

[70] O. Desmaison, P.-A. Pires, G. Levesque, A. Peralta, S. Sundarraj, A. Makinde, V. Jagdale, and M. Megahed, *Influence of computational grid and deposit volume on residual stress and distortion prediction accuracy for additive manufacturing modeling*, in Proceedings of the 4th World Congress on Integrated Computational Materials Engineering (ICME 2017), 2017, pp. 365–374.

[71] D. Ding, Z. Pan, D. Cuiuri, and H. Li, *A practical path planning methodology for wire and arc additive manufacturing of thin-walled structures*, Robotics and Computer-Integrated Manufacturing, 34 (2015), pp. 8–19.

[72] D. Ding, Z. Pan, D. Cuiuri, H. Li, and N. Larkin, *Adaptive path planning for wire-feed additive manufacturing using medial axis transformation*, Journal of Cleaner Production, 133 (2016), pp. 942–952.

[73] D. Ding, Z. Pan, D. Cuiuri, H. Li, and S. van Duin, *Advanced design for additive manufacturing: 3d slicing and 2d path planning*, New trends in 3d printing, (2016), pp. 1–23.

[74] D. Ding, Z. Pan, D. Cuiuri, H. Li, S. van Duin, and N. Larkin, *Bead modelling and implementation of adaptive mat path in wire and arc additive manufacturing*, Robotics and Computer-Integrated Manufacturing, 39 (2016), pp. 32–42.

[75] C. Druzgalski, A. Ashby, G. Guss, W. King, T. Roehling, and M. Matthews, *Process optimization of complex geometries using feed forward control for laser powder bed fusion additive manufacturing*, Additive Manufacturing, (2020), p. 101169.

[76] T. Duong, N. Ferrier, S. Lavernhe, and C. Tournier, *A kinematic simulation software for additive manufacturing metal deposition processes*, in ICWAM'19, International Congress On Welding, Additive Manufacturing And Associated Non Destructive Testing, 2019.

[77] K. Ettaieb, *Contribution à l'optimisation des stratégies de lagase en fabrication additive LPBF*, PhD thesis, Université Paris-Saclay, 2019.

[78] L. C. Evans, *Partial differential equations*, vol. 19 of Graduate Studies in Mathematics, American Mathematical Society, Providence, RI, 1998.

[79] Z.-C. Fang, Z.-L. Wu, C.-G. Huang, and C.-W. Wu, *Review on residual stress in selective laser melting additive manufacturing of alloy parts*, Optics & Laser Technology, 129 (2020), p. 106283.

[80] R. Farouki, T. Koenig, K. Tarabanis, J. Korein, and J. Batchelder, *Path planning with offset curves for layered fabrication processes*, Journal of Manufacturing Systems, 14 (1995), pp. 355–368.

[81] F. Feppon, *Shape and topology optimization of multiphysics systems*, PhD thesis, Ecole Polytechnique, 2019.

[82] F. Feppon, G. Allaire, and C. Dapogny, *Null space gradient flows for constrained optimization with applications to shape optimization.* HAL preprint hal-01972915, 2020.

[83] P. Ferro, R. Meneghello, G. Savio, and F. Berto, *A modified volumetric energy density–based approach for porosity assessment in additive manufacturing process design*, The International Journal of Advanced Manufacturing Technology, (2020), pp. 1–11.

[84] E. Foroozmehr and R. Kovacevic, *Effect of path planning on the laser powder deposition process: thermal and structural evaluation*, The International Journal of Advanced Manufacturing Technology, 51 (2010), pp. 659–669.

[85] M. M. Francois, A. Sun, W. E. King, N. J. Henson, D. Tourret, C. A. Bronkhorst, N. N. Carlson, C. K. Newman, T. S. Haut, J. Bakosi, et al., *Modeling of additive manufacturing processes for metals: Challenges and opportunities*, Current Opinion in Solid State and Materials Science, 21 (2017).

[86] M. Galati, L. Iuliano, A. Salmi, and E. Atzeni, *Modelling energy source and powder properties for the development of a thermal fe model of the ebm additive manufacturing process*, Additive Manufacturing, 14 (2017), pp. 49–59.

[87] W. Gao, Y. Zhang, D. Ramanujan, K. Ramani, Y. Chen, C. B. Williams, C. C. Wang, Y. C. Shin, S. Zhang, and P. D. Zavattieri, *The status, challenges, and future of additive manufacturing in engineering*, Computer-Aided Design, 69 (2015), pp. 65–89.

[88] T. Gatsos, K. A. Elsayed, Y. Zhai, and D. A. Lados, *Review on computational modeling of process–microstructure–property relationships in metal additive manufacturing*, JOM, 72 (2020), pp. 403–419.

[89] P. Geoffroy Donders, *Homogenization method for topology optmization of structures built with lattice materials.*, PhD thesis, Ecole Polytechnique, 2018.

[90] M. Gerdts, *Optimal control of ODEs and DAEs*, De Gruyter Textbook, Walter de Gruyter & Co., Berlin, 2012.

[91] F. Gibou, R. Fedkiw, and S. Osher, *A review of level-set methods and some recent applications*, J. Comput. Phys., 353 (2018), pp. 82–109.

[92] K. Godineau, S. Lavernhe, and C. Tournier, *Calibration of galvanometric scan heads for additive manufacturing with machine assembly defects consideration*, Additive Manufacturing, 26 (2019), pp. 250–257.

[93] M. Grasso and B. M. Colosimo, *Process defects and in situ monitoring methods in metal powder bed fusion: a review*, Measurement Science and Technology, 28 (2017), p. 044005.

[94] A. V. Gusarov, S. N. Grigoriev, M. A. Volosova, Y. A. Melnik, A. Laskin, D. V. Kotoban, and A. A. Okunkova, *On productivity of laser additive manufacturing*, Journal of Materials Processing Technology, 261 (2018), pp. 213–232.

[95] J. Hadamard, *Mémoire sur le problème d'analyse relatif à l'équilibre des plaques élastiques encastrées*, vol. 33, Imprimerie nationale, 1908.

[96] F. Hecht, *New development in FreeFem++*, J. Numer. Math., 20 (2012), pp. 251–265.

[97] A. Henrot and M. Pierre, *Shape variation and optimization*, EMS Tracts in Mathematics, European Mathematical Society (EMS), Zürich, 2018.

[98] D. Herzog, V. Seyda, E. Wycisk, and C. Emmelmann, *Additive manufacturing of metals*, Acta Materialia, 117 (2016), pp. 371–392.

[99] R. Herzog and K. Kunisch, *Algorithms for PDE-constrained optimization*, GAMM-Mitt., 33 (2010), pp. 163–176.

[100] R. Huang, M. Riddle, D. Graziano, J. Warren, S. Das, S. Nimbalkar, J. Cresko, and E. Masanet, *Energy and emissions saving potential of additive manufacturing: the case of lightweight aircraft components*, Journal of Cleaner Production, 135 (2016), pp. 1559–1570.

[101] ISO/ASTM52900-15, *Standard Terminology for Additive Manufacturing – General Principles – Terminology*, standard, ASTM International, West Conshohocken, PA, USA, 2015.

[102] A. Jafari and N. Ashgriz, *Numerical techniques for free surface flows: Interface capturing and interface tracking*, Encyclopedia of Microfluidics and Nanofluidics, (2015), pp. 2458–2479.

[103] M. Jamshidinia and R. Kovacevic, *The influence of heat accumulation on the surface roughness in powder-bed additive manufacturing*, Surface Topography: Metrology and Properties, 3 (2015), p. 014003.

[104] J. Jhabvala, E. Boillat, T. Antignac, and R. Glardon, *On the effect of scanning strategies in the selective laser melting process*, Virtual and physical prototyping, 5 (2010), pp. 99–109.

[105] Y. Jin, J. Du, Z. Ma, A. Liu, and Y. He, *An optimization approach for path planning of high-quality and uniform additive manufacturing*, The International Journal of Advanced Manufacturing Technology, 92 (2017), pp. 651–662.

[106] Y.-a. Jin, Y. He, J.-z. Fu, W.-f. Gan, and Z.-w. Lin, *Optimization of tool-path generation for material extrusion-based additive manufacturing technology*, Additive manufacturing, 1 (2014), pp. 32–47.

[107] C. Kanzow, N. Yamashita, and M. Fukushima, *Levenberg-Marquardt methods with strong local convergence properties for solving nonlinear equations with convex constraints*, J. Comput. Appl. Math., 172 (2004), pp. 375–397.

[108] J.-H. Kao and F. B. Prinz, *Optimal motion planning for deposition in layered manufacturing*, in Proceedings of DETC, vol. 98, Citeseer, 1998, pp. 13–16.

[109] S. Kapil, P. Joshi, H. V. Yagani, D. Rana, P. M. Kulkarni, R. Kumar, and K. Karunakaran, *Optimal space filling for additive manufacturing*, Rapid Prototyping Journal, 22 (2016), pp. 660–675.

[110] G. Kasperovich, J. Haubrich, J. Gussone, and G. Requena, *Correlation between porosity and processing parameters in tial6v4 produced by selective laser melting*, Materials & Design, 105 (2016), pp. 160–170.

[111] S.-J. Kimmerle, M. Gerdts, and R. Herzog, *Optimal control of an elastic crane-trolley-load system—a case study for optimal control of coupled ODE-PDE systems*, Math. Comput. Model. Dyn. Syst., 24 (2018), pp. 182–206.

[112] W. King, A. T. Anderson, R. M. Ferencz, N. E. Hodge, C. Kamath, and S. A. Khairallah, *Overview of modelling and simulation of metal powder bed fusion process at lawrence livermore national laboratory*, Materials Science and Technology, 31 (2015), pp. 957–968.

[113] W. King, A. T. Anderson, R. M. Ferencz, N. E. Hodge, C. Kamath, S. A. Khairallah, and A. M. Rubenchik, *Laser powder bed fusion additive manufacturing of metals; physics, computational, and materials challenges*, Applied Physics Reviews, 2 (2015), p. 041304.

[114] G. L. Knapp, N. Raghavan, A. Plotkowski, and T. Debroy, *Experiments and simulations on solidification microstructure for inconel 718 in powder bed fusion electron beam additive manufacturing*, Additive Manufacturing, 25 (2019), pp. 511–521.

[115] J.-P. Kruth, J. Deckers, E. Yasa, and R. Wauthlé, *Assessing and comparing influencing factors of residual stresses in selective laser melting using a novel analysis method*, Proceedings of the institution of mechanical engineers, Part B: Journal of Engineering Manufacture, 226 (2012), pp. 980–991.

[116] L. J. Kumar and C. K. Nair, *Current trends of additive manufacturing in the aerospace industry*, in Advances in 3d printing & additive manufacturing technologies, Springer, 2017, pp. 39–54.

[117] M. Langelaar, *Combined optimization of part topology, support structure layout and build orientation for additive manufacturing*, Struct. Multidiscip. Optim., 57 (2018), pp. 1985–2004.

[118] T. Larimian, M. Kannan, D. Grzesiak, B. AlMangour, and T. Borkar, *Effect of energy density and scanning strategy on densification, microstructure and mechanical properties of 316l stainless steel processed via selective laser melting*, Materials Science and Engineering: A, 770 (2020).

[119] Y. Lee, M. M. Kirka, J. Ferguson, and V. C. Paquit, *Correlations of cracking with scan strategy and build geometry in electron beam powder bed additive manufacturing*, Additive Manufacturing, 32 (2020), p. 101031.

[120] G. Leugering, J. Sokołowski, and A. Żochowski, *Control of crack propagation by shape-topological optimization*, Discrete Contin. Dyn. Syst., 35 (2015), pp. 2625–2657.

[121] J. J. Lewandowski and M. Seifi, *Metal additive manufacturing: a review of mechanical properties*, Annual Review of Materials Research, 46 (2016), pp. 151–186.

[122] C. Li, J. Liu, X. Fang, and Y. Guo, *Efficient predictive model of part distortion and residual stress in selective laser melting*, Additive Manufacturing, 17 (2017), pp. 157–168.

[123] H. Li, Z. Dong, and G. W. Vickers, *Optimal toolpath pattern identification for single island, sculptured part rough machining using fuzzy pattern analysis*, Computer-Aided Design, 26 (1994), pp. 787–795.

[124] X. Liang, Q. Chen, L. Cheng, D. Hayduke, and A. C. To, *Modified inherent strain method for efficient prediction of residual deformation in direct metal laser sintered components*, Computational Mechanics, 64 (2019), pp. 1719–1733.

[125] J.-L. Lions, *Contrôle optimal de systèmes gouvernés par des équations aux dérivées partielles*, Dunod, Paris; Gauthier-Villars, Paris, 1968.

[126] J. Liu, A. T. Gaynor, S. Chen, Z. Kang, K. Suresh, A. Takezawa, L. Li, J. Kato, J. Tang, C. C. Wang, et al., *Current and future trends in topology optimization for additive manufacturing*, Structural and Multidisciplinary Optimization, 57 (2018), pp. 2457–2483.

[127] J. Liu and A. C. To, *Deposition path planning-integrated structural topology optimization for 3d additive manufacturing subject to self-support constraint*, Computer-Aided Design, 91 (2017), pp. 27–45.

[128] J. Liu and A. C. To, *Quantitative texture prediction of epitaxial columnar grains in additive manufacturing using selective laser melting*, Additive Manufacturing, 16 (2017), pp. 58–64.

[129] J. Liu, Y. Zheng, Y. Ma, A. Qureshi, and R. Ahmad, *A topology optimization method for hybrid subtractive–additive remanufacturing*, International Journal of Precision Engineering and Manufacturing-Green Technology, (2019), pp. 1–15.

[130] S. Liu and Y. C. Shin, *Additive manufacturing of ti6al4v alloy: A review*, Materials & Design, 164 (2019), p. 107552.

[131] Y.-L. Lo, B.-Y. Liu, and H.-C. Tran, *Optimized hatch space selection in double-scanning track selective laser melting process*, The International Journal of Advanced Manufacturing Technology, 105 (2019), pp. 2989–3006.

[132] W. E. Lorensen and H. E. Cline, *Marching cubes: A high resolution 3d surface construction algorithm*, ACM siggraph computer graphics, 21 (1987), pp. 163–169.

[133] Y. Lu, S. Wu, Y. Gan, T. Huang, C. Yang, L. Junjie, and J. Lin, *Study on the microstructure, mechanical property and residual stress of slm inconel-718 alloy manufactured by differing island scanning strategy*, Optics & Laser Technology, 75 (2015), pp. 197–206.

[134] X. Luo, J. Li, and M. Lucas, *Galvanometer scanning technology for laser additive manufacturing*, in Laser 3D Manufacturing IV, vol. 10095, International Society for Optics and Photonics, 2017, p. 1009512.

[135] L. Ma and H. Bin, *Temperature and stress analysis and simulation in fractal scanning-based laser sintering*, The International Journal of Advanced Manufacturing Technology, 34 (2007), pp. 898–903.

[136] E. Maitre, *Review of numerical methods for free interfaces*, Ecole thématique: Modèles de champ de phase pour l' évolution de structures complexes, (2006).

[137] M. Marrey, E. Malekipour, H. El-Mounayri, and E. J. Faierson, *A framework for optimizing process parameters in powder bed fusion (pbf) process using artificial neural network (ann)*, Procedia Manufacturing, 34 (2019), pp. 505–515.

[138] Y. Mayi, M. Dal, P. Peyre, M. Bellet, C. Metton, C. Moriconi, and R. Fabbro, *Laser-induced plume investigated by finite element modelling and scaling of particle entrainment in laser powder bed fusion*, Journal of Physics D: Applied Physics, 53 (2019), p. 075306.

[139] M. Megahed, H.-W. Mindt, N. N'Dri, H. Duan, and O. Desmaison, *Metal additive-manufacturing process and residual stress modeling*, Integrating Materials and Manufacturing Innovation, 5 (2016), pp. 61–93.

[140] P. Mercelis and J.-P. Kruth, *Residual stresses in selective laser sintering and selective laser melting*, Rapid prototyping journal, 12 (2006), pp. 254–265.

[141] G. Michailidis, *Manufacturing Constraints and Multi-Phase Shape and Topology Optimization via a Level-Set Method*, PhD thesis, Ecole Polytechnique, 2014.

[142] J. O. Milewski, *Additive manufacturing of metals*, From Fundamental Technology to Rocket Nozzles, Medical Implants, and Custom Jewelry, (2017), pp. 134–157.

[143] H.-W. Mindt, O. Desmaison, M. Megahed, A. Peralta, and J. Neumann, *Modeling of powder bed manufacturing defects*, Journal of Materials Engineering and Performance, 27 (2018), pp. 32–43.

[144] M. Mirzaali, F. Bobbert, Y. Li, and A. Zadpoor, *Additive manufacturing of metals using powder bed-based technologies*, in Additive Manufacturing (Second Edition), CRC Press, 2019, pp. 93–146.

[145] A. M. Mirzendehdel and K. Suresh, *Support structure constrained topology optimization for additive manufacturing*, Computer-Aided Design, 81 (2016), pp. 1–13.

[146] S. Mohanty and J. H. Hattel, *Reducing residual stresses and deformations in selective laser melting through multi-level multi-scale optimization of cellular scanning strategy*, in Laser 3D Manufacturing III, vol. 9738, International Society for Optics and Photonics, 2016.

[147] T. Mukherjee, H. Wei, A. De, and T. DebRoy, *Heat and fluid flow in additive manufacturing–part i: Modeling of powder bed fusion*, Computational Materials Science, 150 (2018), pp. 304–313.

[148] ——, *Heat and fluid flow in additive manufacturing–part ii: Powder bed fusion of stainless steel, and titanium, nickel and aluminum base alloys*, Computational Materials Science, 150 (2018), pp. 369–380.

[149] F. Murat and J. Simon, *Etude de problemes d'optimal design*, in Optimization Techniques Modeling and Optimization in the Service of Man Part 2, 1976, pp. 54–62.

[150] A. Nassehi, W. Essink, and J. Barclay, *Evolutionary algorithms for generation and optimization of tool paths*, CIRP Annals, 64 (2015), pp. 455–458.

[151] Y. Nesterov, *Introductory lectures on convex optimization: A basic course*, vol. 87, Springer Science & Business Media, 2013.

[152] J. Nocedal and S. J. Wright, *Numerical optimization*, Springer Series in Operations Research and Financial Engineering, Springer, New York, second ed., 2006.

[153] A. A. Novotny and J. Sokłølowski, *Topological derivatives in shape optimization*, Interaction of Mechanics and Mathematics, Springer, Heidelberg, 2013.

[154] J. Oliveira, A. LaLonde, and J. Ma, *Processing parameters in laser powder bed fusion metal additive manufacturing*, Materials & Design, (2020), p. 108762.

[155] S. Osher and R. Fedkiw, *Level set methods and dynamic implicit surfaces*, vol. 153 of Applied Mathematical Sciences, Springer-Verlag, New York, 2003.

[156] S. Osher and J. A. Sethian, *Fronts propagating with curvature-dependent speed: algorithms based on Hamilton-Jacobi formulations*, J. Comput. Phys., 79 (1988), pp. 12–49.

[157] N. Parikh and S. Boyd, *Proximal algorithms*, Foundations and Trends in optimization, 1 (2013), pp. 123–231.

[158] S. C. Park and B. K. Choi, *Tool-path planning for direction-parallel area milling*, Computer-Aided Design, 32 (2000), pp. 17–25.

[159] L. Parry, I. Ashcroft, and R. D. Wildman, *Understanding the effect of laser scan strategy on residual stress in selective laser melting through thermo-mechanical simulation*, Additive Manufacturing, 12 (2016), pp. 1–15.

[160] S. Pereira, A. Vaz, and L. Vicente, *On the optimal object orientation in additive manufacturing*, The International Journal of Advanced Manufacturing Technology, 98 (2018), pp. 1685–1694.

[161] C. Pleass and S. Jothi, *Influence of powder characteristics and additive manufacturing process parameters on the microstructure and mechanical behaviour of inconel 625 fabricated by selective laser melting*, Additive Manufacturing, 24 (2018), pp. 419–431.

[162] C. Qiu, M. Al Kindi, A. S. Aladawi, and I. Al Hatmi, *A comprehensive study on microstructure and tensile behaviour of a selectively laser melted stainless steel*, Scientific reports, 8 (2018), pp. 1–16.

[163] V. Rajan, V. Srinivasan, and K. A. Tarabanis, *The optimal zigzag direction for filling a two-dimensional region*, Rapid Prototyping Journal, 7 (2001), pp. 231–241.

[164] L. Rakotondrainibe, *Join topology optimisation of part and support for metal additive manufacturing*, PhD thesis, Ecole Polytechnique, 2020.

[165] F. Rasoanarivo, P. Rodriguez-Aycrbe, and D. Dumur, *Galvanometer scanner modeling for selective laser melting deflection system simulation*, in 2018 15th International Conference on Control, Automation, Robotics and Vision (ICARCV), IEEE, 2018, pp. 1170–1175.

[166] P.-H. Renard, *Les mystères de la fusion par faisceau d'électrons (ebm).*

[167] I. A. Roberts, C. Wang, R. Esterlein, M. Stanford, and D. Mynors, *A three-dimensional finite element analysis of the temperature field during laser melting of metal powders in additive layer manufacturing*, International Journal of Machine Tools and Manufacture, 49 (2009), pp. 916–923.

[168] J. Robinson, I. Ashton, P. Fox, E. Jones, and C. Sutcliffe, *Determination of the effect of scan strategy on residual stress in laser powder bed fusion additive manufacturing*, Additive Manufacturing, 23 (2018), pp. 13–24.

[169] P. Romon, *Introduction à la géométrie différentielle discrète*, Ellipses, 2013.

[170] A. Rondepierre, *Méthodes numériques pour l'optimisationnon linéaire déterministe.*, Lecture notes: INSA Toulouse, (2017).

[171] J. A. Sethian and P. Smereka, *Level set methods for fluid interfaces*, in Annual review of fluid mechanics, Vol. 35, vol. 35 of Annu. Rev. Fluid Mech., Annual Reviews, Palo Alto, CA, 2003, pp. 341–372.

[172] H. Shipley, D. McDonnell, M. Culleton, R. Coull, R. Lupoi, G. O'Donnell, and D. Trimble, *Optimisation of process parameters to address fundamental challenges during selective laser melting of ti-6al-4v: A review*, International Journal of Machine Tools and Manufacture, 128 (2018), pp. 1–20.

[173] S. Sillars, C. Sutcliffe, A. Philo, S. Brown, J. Sienz, and N. Lavery, *The three-prong method: a novel assessment of residual stress in laser powder bed fusion*, Virtual and Physical Prototyping, 13 (2018), pp. 20–25.

[174] S. Singh and S. Ramakrishna, *Biomedical applications of additive manufacturing: present and future*, Current opinion in biomedical engineering, 2 (2017), pp. 105–115.

[175] J. Sokołowski and A. Żochowski, *On the topological derivative in shape optimization*, SIAM J. Control Optim., 37 (1999), pp. 1251–1272.

[176] J. Sokołowski and A. Żochowski, *Shape and topology sensitivity analysis for elastic bodies with rigid inclusions and cracks*, in Computer Methods in Mechanics, Springer, 2010, pp. 75–98.

[177] J. C. Steuben, A. P. Iliopoulos, and J. G. Michopoulos, *Implicit slicing for functionally tailored additive manufacturing*, Computer-Aided Design, 77 (2016), pp. 107–119.

[178] Z.-j. Tang, W.-w. Liu, Y.-w. Wang, K. M. Saleheen, Z.-c. Liu, S.-t. Peng, Z. Zhang, and H.-c. Zhang, *A review on in situ monitoring technology for directed energy deposition of metals*, The International Journal of Advanced Manufacturing Technology, (2020), pp. 1–27.

[179] Y. W. D. Tay, B. Panda, S. C. Paul, N. A. Noor Mohamed, M. J. Tan, and K. F. Leong, *3d printing trends in building and construction industry: a review*, Virtual and Physical Prototyping, 12 (2017), pp. 261–276.

[180] M. K. Thompson, G. Moroni, T. Vaneker, G. Fadel, R. I. Campbell, I. Gibson, A. Bernard, J. Schulz, P. Graf, B. Ahuja, et al., *Design for additive manufacturing: Trends, opportunities, considerations, and constraints*, CIRP annals, 65 (2016), pp. 737–760.

[181] I. toolbox, *Iscdtoolbox: Advection.* https://github.com/ISCDtoolbox/Advection.

[182] E. Trélat, *Contrôle optimal*, Mathématiques Concrètes. [Concrete Mathematics], Vuibert, Paris, 2005. Théorie & applications. [Theory and applications].

[183] G. Tryggvason, B. Bunner, A. Esmaeeli, D. Juric, N. Al-Rawahi, W. Tauber, J. Han, S. Nas, and Y.-J. Jan, *A front-tracking method for the computations of multiphase flow*, Journal of computational physics, 169 (2001), pp. 708–759.

[184] A. Uriondo, M. Esperon-Miguez, and S. Perinpanayagam, *The present and future of additive manufacturing in the aerospace sector: A review of important aspects*, Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering, 229 (2015), pp. 2132–2147.

[185] Z.-D. Z. Usman, I. Osezua, A. Dibia, F. Chinedu, R. Pouyan, A. Bonakdar, and E. Toyserkani, *Topology optimization parallel-computing framework based on the inherent strain method for support structure design in laser powder-bed fusion additive manufacturing*, International Journal of Mechanics and Materials in Design, (2020).

[186] L. Van Belle, *Analyse, modélisation et simulation de l'apparition de contraintes en fusion laser métallique*, PhD thesis, INSA Lyon, 2013.

[187] L. Van Belle, G. Vansteenkiste, and J. C. Boyer, *Investigation of residual stresses induced during the selective laser melting process*, in Key Engineering Materials, vol. 554, Trans Tech Publ, 2013, pp. 1828–1834.

[188] N. Van Goethem and A. A. Novotny, *Crack nucleation sensitivity analysis*, Math. Methods Appl. Sci., 33 (2010), pp. 1978–1994.

[189] G. Vastola, G. Zhang, Q. Pei, and Y.-W. Zhang, *Controlling of residual stress in additive manufacturing of ti6al4v by finite element modeling*, Additive Manufacturing, 12 (2016), pp. 231–239.

[190] N. Vermaak, M. Boissier, L. Valdevit, and R. McMeeking, *Some graphical interpretations of melan's theorem for shakedown design*, in Advances in Direct Methods for Materials and Structures, Springer, 2018, pp. 179–198.

[191] G. Walrand, C. Tournier, S. Lavernhe, and K. Ettaieb, *Fabrication additive par modulation de puissance laser*, patent FR3092020, (2019).

[192] ——, *Fabrication additive par modulation de vitesse de balayage*, patent FR3092012, (2019).

[193] Q. Wang, P. P. Michaleris, A. R. Nassar, J. E. Irwin, Y. Ren, and C. B. Stutzman, *Model-based feedforward control of laser powder bed fusion additive manufacturing*, Additive Manufacturing, 31 (2020), p. 100985.

[194] T. Wasser, A. D. Jayal, and C. Pistor, *Implementation and evaluation of novel buildstyles in fused deposition modeling (fdm)*, in 1999 International Solid Freeform Fabrication Symposium, 1999.

[195] B. Wen, X. Chen, and T. P. Pong, *Linear convergence of proximal gradient algorithm with extrapolation for a class of nonconvex nonsmooth minimization problems*, SIAM Journal on Optimization, 27 (2017), pp. 124–145.

[196] S. Wendl, *On a prototype of an optimal control problem governed by ordinary and partial differential equations*, PhD thesis, University of Bayreuth, Faculty of Mathematics, Physics and Computer Sciences, 2014.

[197] S. Wendl, H. J. Pesch, and A. Rund, *On a state-constrained pde optimal control problem arising from ode-pde optimal control*, in Recent Advances in Optimization and its Applications in Engineering, Springer, 2010, pp. 429–438.

[198] H. Wessels, T. Bode, C. Weissenfels, P. Wriggers, and T. Zohdi, *Investigation of heat source modeling for selective laser melting*, Computational Mechanics, 63 (2019), pp. 949–970.

[199] B. Wysocki, P. Maj, R. Sitek, J. Buhagiar, K. J. Kurzydłowski, and W. Świeszkowski, *Laser and electron beam additive manufacturing methods of fabricating titanium bone implants*, Applied Sciences, 7 (2017), p. 657.

[200] Y. Xiong, A. Van Campen, A. Van Vlierberghe, K. Kempen, and J.-P. Kruth, *Time-optimal scan path planning based on analysis of sliced geometry*, in Proceedings of the 28th Annual International Solid Freeform Fabrication Symposium. Austin, Texas, USA, 2017.

[201] A. Yadollahi, N. Shamsaei, S. M. Thompson, A. Elwany, and L. Bian, *Effects of building orientation and heat treatment on fatigue behavior of selective laser melted 17-4 ph stainless steel*, International Journal of Fatigue, 94 (2017), pp. 218–235.

[202] Q. Yang, P. Zhang, L. Cheng, Z. Min, M. Chyu, and A. C. To, *Finite element modeling and validation of thermomechanical behavior of ti-6al-4v in directed energy deposition additive manufacturing*, Additive Manufacturing, 12 (2016), pp. 169–177.

[203] Y. Yang, H. T. Loh, J. Fuh, and Y. Wang, *Equidistant path generation for improving scanning efficiency in layered manufacturing*, Rapid Prototyping Journal, 8 (2002), pp. 30–37.

[204] Y. Yang, F. van Keulen, and C. Ayas, *A computationally efficient thermal model for selective laser melting*, Additive Manufacturing, 31 (2020), p. 100955.

[205] H. Yeung, B. M. Lane, M. Donmez, J. C. Fox, and J. Neira, *Implementation of advanced laser control strategies for powder bed fusion systems*, Procedia Manufacturing, 26 (2018), pp. 871–879.

[206] H. Yeung, Z. Yang, and L. Yan, *A meltpool prediction based scan strategy for powder bed fusion additive manufacturing*, Additive Manufacturing, 35 (2020), p. 101383.

[207] Y.-L. Yu, *On decomposing the proximal map*, in Advances in Neural Information Processing Systems, 2013, pp. 91–99.

[208] P. Zhang, J. Liu, and A. C. To, *Role of anisotropic properties on topology optimization of additive manufactured load bearing structures*, Scripta Materialia, 135 (2017), pp. 148–152.

[209] S. Zhang, G. Guillemot, C.-A. Gandin, and M. Bellet, *A partitioned two-step solution algorithm for concurrent fluid flow and stress–strain numerical simulation in solidification processes*, Computer Methods in Applied Mechanics and Engineering, 356 (2019), pp. 294–324.

[210] Y. Zhang, Q. Chen, G. Guillemot, M. Bellet, and C.-A. Gandin, *Macro-and meso-scale finite element modelling of the laser beam melting process*, in 2nd Int. Conf. on Metallic Materials and Processes (MMP2017), 2017.

[211] Y. Zhang, Q. Chen, G. Guillemot, C.-A. Gandin, and M. Bellet, *Numerical modelling of fluid and solid thermomechanics in additive manufacturing by powder-bed fusion: Continuum and level set formulation applied to track-and part-scale simulations*, Comptes Rendus Mécanique, 346 (2018), pp. 1055–1071.

[212] C. Zhuang, Z. Xiong, and H. Ding, *High speed machining tool path generation for pockets using level sets*, International journal of production research, 48 (2010), pp. 5749–5766.

[213] J. Zielinski, S. Vervoort, H.-W. Mindt, and M. Megahed, *Influence of powder bed characteristics on material quality in additive manufacturing*, BHM Berg-und Hüttenmännische Monatshefte, 162 (2017), pp. 192–198.

ECOLE
DOCTORALE
DE MATHEMATIQUES
HADAMARD

Titre : Couplage de méthodes d'optimisation de formes et d'optimisation de trajectoires en fabrication additive

Mots clés : Génération et contrôle de trajectoires, fabrication additive, fusion sur lit de poudre métallique, optimisation de forme

Résumé : Cette thèse porte sur l'optimisation des trajectoires de lasage pour la fabrication additive sur lit de poudre, ainsi que leur lien avec la géométrie de la pièce à construire. L'état de l'art est principalement constitué par des trajectoires basées sur des motifs, dont l'impact sur les propriétés mécaniques des objets finaux est quantifié. Cependant, peu d'analyses permettent de relier leur pertinence à la forme de la pièce elle-même. Nous proposons dans ce travail une approche systématique visant à optimiser la trajectoire sans restriction a priori. Le problème d'optimisation consiste à fusionner la structure en évitant de surchauffer (ce qui induirait des contraintes résiduelles) tout en minimisant le temps de fabrication. L'équation d'état est donc l'équation de la chaleur, dont le terme source dépend de la trajectoire. Deux modèles 2-d sont proposés pour contrôler la température : l'un transitoire et le second stationnaire (pas de dépendance en temps). Basés sur des techniques d'optimisation de forme pour le stationnaire et sur des outils de contrôle pour le transitoire, des algorithmes d'optimisation sont développés. Les applications numériques qui en découlent permettent une analyse critique des différents choix effectués. Afin de laisser plus de liberté dans la conception, l'algorithme stationnaire est adapté à la modification du nombre de composantes connexes de la trajectoire lors de l'optimisation. Deux méthodes sont comparées. Dans la première, la puissance de la source est ajoutée aux variables d'optimisation et un algorithme impliquant une relaxation-pénalisation et un contrôle de la variation totale est proposé. Dans la seconde, la notion de dérivation topologique sont adaptés à la source. Enfin, dans le cadre stationnaire, nous détaillons le couplage de l'optimisation de la forme de la pièce pour améliorer ses performances mécaniques et de la trajectoire de lasage. Ce problème multiphysique ouvre des perspectives d'applications et de généralisations futures.

Title : Coupling structural optimization and trajectory optimization methods in additive manufacturing

Keywords : Path planning and control, additive manufacturing, metallic powder bed fusion, structural optimization

Abstract : This work investigates path planning optimization for powder bed fusion additive manufacturing processes, and relates them to the design of the built part. The state of the art mainly studies trajectories based on existing patterns and, besides their mechanical evaluation, their relevance has not been related to the object's shape. We propose in this work a systematic approach to optimize the path without any a priori restriction. The typical optimization problem is to melt the desired structure, without over-heating (to avoid thermally induced residual stresses) and possibly with a minimal path length. The state equation is the heat equation with a source term depending on the scanning path. Two physical 2-d models are proposed, involving temperature constraint: a transient and a steady state one (in which time dependence is removed). Based on shape optimization for the steady state model and control for the transient model, path optimization algorithms are developed. Numerical results are then performed allowing a critical assessment of the choices we made. To increase the path design freedom, we modify the steady state algorithm to introduce path splits. Two methods are compared. In the first one, the source power is added to the optimization variables and an algorithm mixing relaxation-penalization techniques and the control of the total variation is set. In a second method, notion of topological derivative are applied to the path to cleverly remove and add pieces. eventually, in the steady state, we conduct a concurrent optimization of the part's shape and of the scanning path. This multiphysics optimization problem raises perspectives gathering direct applications and future generalizations.

Institut Polytechnique de Paris
91120 Palaiseau, France