ADDITIVE MANUFACTURING SCANNING PATHS OPTIMIZATION USING SHAPE OPTIMIZATION TOOLS

M Boissier, G. Allaire, C. Tournier

SCANNING PATH OPTIMIZATION USING SHAPE OPTIMIZATION TOOLS

M. BOISSIER[1,2*], G. ALLAIRE[1], C. TOURNIER[2]

[1] *CMAP, École Polytechnique, CNRS UMR7641, Institut Polytechnique de Paris, Palaiseau, France*
[2] *Université Paris-Saclay, ENS Paris-Saclay, Laboratoire Universitaire de Recherche en Production Automatisée, 94235 Cachan, France*

[*] *mathilde.boissier@cmap.polytechnique.fr*

Abstract: This paper investigates path planning strategies for additive manufacturing processes such as powder bed fusion. The state of the art mainly studies trajectories based on existing patterns. Parametric optimization on these patterns or allocating them to the object areas are the main strategies. We propose in this work a more systematic optimization approach without any a priori restriction on the trajectories. 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. First, in a steady-state context, shape optimization tools are applied to trajectories. Second, for time-dependent problems, an optimal control method is considered instead. In both cases, gradient type algorithms are deduced and tested on 2-d examples. Numerical results are discussed, leading to a better understanding of the problem and thus to short- and long-term perspectives.

Keywords. Path planning, Additive manufacturing, Laser based powder bed fusion, Electron beam powder bed fusion, Shape Optimization

CONTENTS

1 Introduction

Additive manufacturing (AM) consists in building structures layer by layer. This concept has been developed for different materials such as plastic and metal, with an adaptation of the process. Among processes, powder bed fusion builds objects according to the following scheme: powder is regularly distributed and a source of thermal energy, moving along a planned trajectory, selectively fuses the material. Solidification comes from the cooling and a new layer of powder is coated to repeat the process. After all layers have been built, removing the powder reveals the structure which, very often, requires some post-processes (like cutting it

from the baseplate and surface treatment). Breaking with traditional manufacturing, such a process presents many advantages in different industrial fields such as aeronautics and biomedical engineering. Reducing the topological constraints, the manufactured shapes can be more complex than before and better achieve functional requirements. Unlike molding, producing a wide range of different items is easy, saving the mold's material and machining time and thus reducing time to market [9, 22, 50, 57].

As promising as this process seems, some issues yet need to be solved. Many thermal, mechanical and metallurgical phenomena occur, affecting the object's quality [14, 19, 33, 34, 38, 46, 58]. The kinematics of the scanning head, and especially the control of the manufacturing time, must also be mastered. Therefore, there are many ongoing research on various topics, including modeling of the physical phenomena and control of the scanning path. With so many nonlinear phenomena at stake and four material phases involved (solid, powder, liquid, gas), high-fidelity simulations are highly problematic in terms of predictability and computational cost. Simplifying ingeniously the problem and developing more efficient tools are thus studied [16, 19, 21, 37, 38, 47, 53]. From these models, the process can be improved by optimizing the powder quality, the process parameters (velocity, power, layer thickness) and the scanning strategy (path, hatch distance) [27, 49]. Finally, the design of the object itself can be adapted. Topology optimization is tightly related to AM. Indeed, this process allows for the building of the complex objects designed. This optimization can also include some process constraints: optimization of supports [4, 30, 39], minimization of the residual stresses [5], dealing with the anisotropy [32].

In this work, we focus on optimizing the heat source path for each layer. In powder bed fusion, the scanning path's design impacts directly the heat distribution and thus the final quality of the part [18, 27, 35]. These questions were studied even before AM, with pocket machining [8, 12, 28]. In these works, paths are studied based on existing patterns, split in different categories [20, 33]: zigzag (simple to program but creating thermal defects [27]), contour (based on an offset of the object boundary), spirals, hybrid (mixing contours and zigzags), continuous (among them fractal paths), medial axis transformation. Strategies in which the object is split in cells were also developed [27, 40] with, for example, genetic algorithms which allocate patterns to each cell [40]. A recent track consists, starting from a chosen initialization, in adapting the path "on-line", while manufacturing the object [48, 54]. Finally, coupled optimization of the object and of a contour path has been considered to deal with anisotropy [32]. These references are not specifically dedicated to powder bed fusion but they present interesting ideas in path design.

Departing from the traditional literature, where paths are based on patterns, the goal of the present study is to optimize paths scanned by a moving source, without any restriction on their shape. To the best of our knowledge, this is a new approach and we know only one other recent work dealing with the topic [1], which applies an optimal control approach to path design. The major difference between our work and [1] is that our building constraint is based on physics (the temperature must attain a melting value in the built structure) while theirs is based on geometry (a tubular neighborhood of the path must cover the built structure). Such an optimization could seem too costly to be used straight in the industry. Indeed, compared to the patterns enumerated before, generating an optimal path would have a higher computational cost and would be difficult to modify on-line. However, it may give some intuition about the paths to choose, validating some patterns or creating new ones related to the object design.

Section 2 presents the model used for this study. As already mentioned, modeling and simulating additive manufacturing processes is a research field in itself. To optimize the path, we need to simulate the manufacturing process. Since it is to be included in an optimization loop, such a simulation must not be too costly. This leads to the choice of a macroscopic scale model, considering only powder and solid and forgetting about the liquid and gaseous states. Whereas this approach should allow for controlling both the thermal expansion and the residual stresses, we choose here to focus on the thermal effects only, avoiding the resolution of the mechanical problem. Yet, we can optimize the built area as well as the thermal expansion, through a control of the maximal temperature. This model is simplified again by considering only a two-dimensional layer (details about this modeling process can be found in [11] or in the Appendix 10). If not fully realistic, this simple formulation allows us to test our optimization algorithms and it could be enriched in the future.

Section 3 simplifies further the problem by considering a steady state version of it, where the path is seen as a hot thread. Representing the path as a line in the plane, it is amenable to classical shape optimization tools, like Hadamard method of shape differentiation. Therefore, for various objective functions or constraints, either geometric (length of the path) or physical (maximum or minimum temperature thresholds), the path can be optimized by means of a gradient descent method. In the steady case, our work is related to various interface optimization problems, as [44] for imperfect interfaces in heat transfers, [7] for crack propagation, [2, 5, 43] for material interfaces. Note, however, that in our model the line is not an interface but just the support of a source term, which is a simpler problem. Our numerical results match physical intuition and validate the algorithm, highlighting the non-uniqueness of the minimum and thus the non-convexity of the optimization problem. The steady state model can be seen as a toy model, far from reality, although it may make sense for electron beam processes where the beam travels much faster than for a laser. However, it is computationally

cheap and allows for quite a fast testing of several technical ingredients. It is thus a convenient first step to get insight before adressing the true unsteady problem.

Section 4 focuses on the more realistic unsteady model. Shape optimization theories [3, 25] are not fully adapted to this problem and they must be complemented by optimal control ideas: the path is now characterized by its tangent direction and the optimization variable is the angle between this tangent and the horizontal axis. The control variable is the path angle and the controlled state is the temperature, solution of the heat equation. Therefore, there is a coupling between a Partial Differential Equation (PDE) for the temperature and an Ordinary Differential Equation (ODE) for the path evolution. If there exist well developed theories about both the optimal control of ODE [23, 51] and optimal control of PDE [26, 31], not so many works focus on mixing both [17, 29, 56, 55]. This coupling leads to the introduction of two adjoint states, one for the PDE and another one for the ODE, which allow us to compute derivatives of objective functions and to propose a gradient-based optimization algorithm. Section 5 discusses the numerical results of this method, enlightening its assets and drawbacks. Finally, Section 6 gives concluding remarks and perspectives.

2 Model presentation of the powder bed fusion process

Modeling the powder bed fusion process is a research problem in its own. Indeed, from the energy absorption by the powder to the liquid melting pool and finally the solidification, many physical and mechanical phenomena occur. The works [19, 37] give a large overview of these different issues. Two distinct modeling approaches could be considered. The first one consists of a high-fidelity model, with a microscopic and complete physical description of the process. It includes phase changes, melting pools and a gaseous phase. The second approach is macroscopic and relies on many simplifications which make it economical in terms of computational cost for simulation. Here, because an optimization loop is involved, we follow the second approach.

2.1 Two dimensional thermal model

The objective of this work is to optimize the scanning paths to improve the quality of the final part, while keeping a relevant manufacturing time. Focusing on the macroscopic scale, this amounts to controlling the thermal expansion and the thermally induced residual stresses. The latter requires the resolution of a mechanical system, which is costly and complicated. However, residual stresses are mainly due to the spatial temperature gradient and to thermal expansion [19, 38, 53]. To simplify the resolution, we thus focus on the thermal problem and thermal expansion only. Considering a full thermo-mechanical problem is part of our perspectives.

We consider one layer at a time and, assuming a vertical build direction e_z, the working domain Σ is chosen as a two dimensional horizontal cross-section of the build chamber. At the initial time $t_0 = 0$, the domain is maintained at a fixed temperature T_{init}. A source q is switched on and moved along a trajectory Γ until the final time $t_F > 0$. Conduction only is taken into account, forgetting about convection and radiation. Because the working domain is surrounded by low conductivity powder, its boundary $\partial\Sigma$ is assumed to be adiabatic. Even if the powder can melt and solidify, the density ρ, the heat capacity c and the conductivity λ are assumed to be constant in time and space. Removing this assumption is part of our perspectives. Finally, because of the restriction to space dimension two, the heat conduction along the vertical axis must be considered and a corrective term (similar to a heat sink) is added to the model. The resulting heat equation is given by (1), with a coefficient $\beta > 0$, defined in Appendix 10, and L a characteristic length related to the vertical direction. More details about this model can be found in Appendix 10. The temperature T is the solution of:

$$\begin{cases} \rho c \partial_t T(t,x) - \nabla \cdot (\lambda \nabla T(t,x)) + \dfrac{\beta}{L}(T(t,x) - T_{init}) = \dfrac{q(t,x)}{L} & \text{in } (0,t_F) \times \Sigma, \\ \lambda \partial_n T(t,x) = 0 & \text{on } (0,t_F) \times \partial\Sigma, \\ T(0,x) = T_{init}(x) & \text{in } \Sigma. \end{cases} \tag{1}$$

To ease the notations, in the following we set $\tilde{\rho} = \rho c$, $\tilde{\beta} = \frac{\beta}{L}$, $\tilde{q} = \frac{q}{L}$.

Following most macroscopic models [19, 37, 53], the source is considered to be a Gaussian beam, given by (2), with $P > 0$ an effective power and $r_C > 0$, a parameter related to the focusing of the beam:

$$\tilde{q}(t,x) = \frac{P}{L} \exp\left(-\frac{1}{r_c^2} r(x,t)^2\right), \quad 0 \le t \le t_F. \tag{2}$$

The quantity $r(x,t) \ge 0$ is the distance of the point x to the source center $u(t)$ which is defined as the solution of the following ordinary differential equation, defining the trajectory:

$$\begin{cases} \dot{u}(t) = V\tau(t), \quad 0 \le t \le t_F, \\ u(0) = \tilde{u}, \end{cases} \tag{3}$$

with constant velocity $V > 0$ and tangent unit vector $\tau(t)$. The initial condition $\tilde{u}$ is the starting point of the path. The family of points $u(t)$ for $0 \leq t \leq t_F$ defines a curve (the scanning path) which is called Γ.

2.2 Objective and constraints

In this work, the source is assumed to move at constant speed V and it is not allowed to be switched off and on during the building process. Therefore, the length of the path is proportional to the final time t_F. There is no reason to fix the path length. Therefore, both the path Γ and the final time t_F are optimization variables. To simplify, in the following, by a slight abuse of language, we shall say that Γ varies when actually both the path itself and the final time t_F vary. Let us introduce the various constraints and objective functions used in the sequel.

- Control of the solid region. Assume that $\Sigma_S \subset \Sigma$ is the part of the layer that must solidify. Then, $\forall x \in \Sigma_S$, there must be a time $t \in (0, t_F)$ (depending on x) such that the temperature is above the fusion, or phase change, temperature T_ϕ, namely $T(t,x) > T_\phi$. Therefore, the following constraint

$$C_\phi(t_F, T) = \int_{\Sigma_S} \left[\left(T_\phi - \max_{t \in (0,t_F)} T(t,x) \right)^+ \right]^2 dx,$$

with the notation $(.)^+ = \max(0,.)$, 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 t_F and T. Since the maximum function in time is not differentiable, it is approximated by a $L^\mathfrak{p}$-norm in time. An effective version of the constraint is given by:

$$C_\phi(t_F, T) = \int_{\Sigma_s} \left[\left(T_\phi - N_\mathfrak{p}(t_F, T)(x) \right)^+ \right]^2 dx, \quad N_\mathfrak{p}(t_F, T)(x) = \left(\frac{1}{t_F} \int_0^{t_F} |T(t,x)]^\mathfrak{p} dt \right)^{1/\mathfrak{p}} \tag{4}$$

- Control of the maximal temperature. If the region Σ_S must solidify, the rest of the working domain should not. This can be controlled by setting a maximum temperature $T_{M,\Sigma\setminus\Sigma_s}$ in the region $\Sigma \setminus \Sigma_s$, such that $T_{M,\Sigma\setminus\Sigma_s} < T_\phi$. Moreover, recall that thermal stresses are typically computed like $\sigma_{th} = C\,(T - T_{init})\,\mathbb{I}_2$ with C a material parameter (related to thermal expansion), T_{init} the initial temperature and $\mathbb{I}_2$ the identity matrix. Hence, to minimize the thermal stresses induced by the source, one can impose a maximum temperature $T_{M,\Sigma_s} > T_\phi$ in the region Σ_s. The maximum temperature T_M is thus space dependent, defined by:

$$T_M(x) = \begin{cases} T_{M,\Sigma_s} & x \in \Sigma_s, \\ T_{M,\Sigma\setminus\Sigma_s} & x \in \Sigma \setminus \Sigma_s. \end{cases} \tag{5}$$

This maximum temperature should be imposed everywhere and at any time. Of course, the choice of the precise numerical values of T_M has a strong influence on the resulting optimized paths. Converting this pointwise constraint into an integral one, the maximum temperature constraint is then:

$$C_M(t_F, T) = \int_\Sigma \int_0^{t_F} \left[(T(t,x) - T_M(x))^+ \right]^2 dx dt \tag{6}$$

Indeed, the choice of these maximum temperature inside the Σ_s (T_{M,Σ_s}) and outside ($T_{M,\Sigma\setminus\Sigma_s}$) will impact the results. The temperature T_{M,Σ_s} must be chosen related to the physics. The temperature $T_{M,\Sigma\setminus\Sigma_s}$ should be taken equal to T_ϕ. However, to make sure that the powder will not melt out of Σ_s, we could also choose $T_{M,\Sigma\setminus\Sigma_s} < T_\phi$.

- Control of the execution time t_F or equivalently of the path length L_F (since the velocity is constant):

$$L_F = V\, t_F\,.$$

Remark 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 2. *In our model, the phase change is instantaneous. This is inaccurate since, in the 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 $T_{\phi,eff}$ such that $T_{\phi,eff} > T_\phi$. Thus, since the temperature varies continuously in time, it would remain above T_ϕ a bit longer.*

Recalling that the notation Γ denotes both the scanning path and the final time, the optimization problem is finally:

$$\min_{\Gamma} J(\Gamma) = L_F \quad \text{such that} \quad \begin{array}{l} C_\phi = C_M = 0, \\ T \text{ solution of (1).} \end{array} \tag{7}$$

3 Steady case: shape optimization of the path

To simplify further the analysis, we consider a toy model, obtained by assuming a steady state configuration. This model is not realistic but it gives, at a very low computational cost, some insights on how to consider the optimization problem as well as some intuition on the results.

3.1 Steady case settings

3.1.1 Model

In this time-independent model, the source is no longer moving. Instead, the energy is applied on the whole trajectory Γ as a hot thread, which would be the case in electron beam powder bed fusion with regard to scanning speeds. The steady source is hence given by $Q = P_Q \mathbb{1}_\Gamma$, with P_Q a constant linear power divided by a characteristic path length and $\mathbb{1}_\Gamma$ the Dirac mass for the path Γ (the source is carried by a line and its thickness is neglected). The steady case heat equation reads:

$$\left\{ \begin{array}{ll} -\nabla \cdot (\lambda \nabla T(x)) + \tilde{\beta}(T(x) - T_{init}) = Q(x) = P_Q \mathbb{1}_\Gamma(x) & \text{in } \Sigma, \\ \lambda \partial_n T(x) = 0 & \text{on } \partial\Sigma, \\ T(0,x) = T_{init}(x) & \text{in } \Sigma. \end{array} \right. \tag{8}$$

Equivalently, the temperature $T \in H^1(\Sigma)$ is the solution of the variational formulation: $\forall \phi \in H^1(\Sigma)$,

$$\int_\Sigma \left(\lambda \nabla T \cdot \nabla \phi + \tilde{\beta}\,(T - T_{init})\,\phi \right) dx - \int_\Gamma P_Q \phi \, ds = 0. \tag{9}$$

The corresponding steady objective function and constraints are:

- control of the solid region: the phase-change constraint becomes $\forall x \in \Sigma_S,\ T(x) \geq T_\phi$ resulting in

$$C_\phi(T) = \int_{\Sigma_S} \left[\left(T_\phi - T(x) \right)^+ \right]^2 dx.$$

- control of the maximum temperature: the constraint becomes $\forall x \in \Sigma,\ T(x) \leq T_M(x)$, resulting in:

$$C_M(T) = \int_\Sigma \left[\left(T(x) - T_M(x) \right)^+ \right]^2 dx.$$

- control of 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:

$$L_F = \int_\Gamma ds.$$

3.1.2 Optimization problem

The steady state model leads to the statement of an optimization problem. The path length is minimized under the phase-change and maximal temperature constraints. Since these two constraints are similar, namely they are non-negative quantities which must vanish, they are gathered in a single constraint C, with positive weights $l_\phi, l_M > 0$. The optimization problem boils down to:

$$\min_{\Gamma} L_F = \int_\Gamma ds, \quad \text{such that} \quad \left\{ \begin{array}{l} C(T) = l_\phi C_\phi(T) + l_M C_M(T) = 0, \\ T \text{ solution of (8).} \end{array} \right. \tag{10}$$

3.2 Optimization tools

To solve the optimization problem (10), a gradient descent algorithm shall be used, which requires computing derivatives of the objective function and of the constraints with respect to the line Γ.

3.2.1 Path representation: front tracking methods

Along the iterations, the path is modified and the temperature is evaluated. The choice of both the meshing process and the line representation are crucial since they impact the computational time and the accuracy. Discretizing a line or an interface is a very classical problem in many fields of science and engineering. Following [52], our method is based on two different meshes: one, called "(physical) mesh", for the heat equation in the working domain and one for the path Γ, referred to as "discretized path". The mesh is kept fixed whereas the discretized path (and especially its nodes) moves along the optimization iterations (Figure 2). This method creates oriented trajectories very easy to handle. In the present work, the path is discretized using segments and is thus represented by a broken line (Figure 1). This representation is very convenient because it keeps the mesh fixed, thereby considerably reducing the computational costs. Moreover, the discretized path allows for a full control of the line topology (no uncontrolled changes in the number of connected components). However, there are two technical issues to be discussed [52].

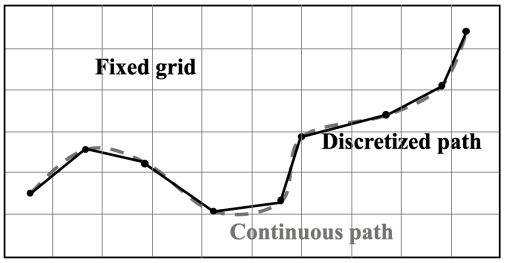


Figure 1: Front tracking meshes

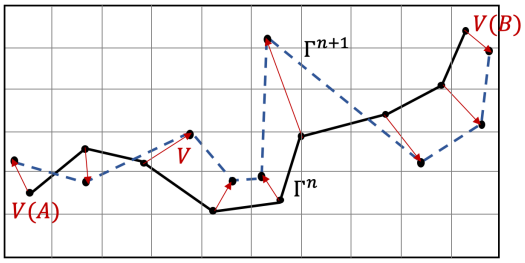


Figure 2: Advection of the discretized path

First issue, related to the path description: at each iteration, the path discretization's nodes are advected, modifying the length of each element. To keep a relevant representation, the discretization must be adapted [36, 52]. 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 the range $[d_{lower}, d_{upper}]$: if the length of a segment is above the upper bound, the middle point is added to the discretization (points added equidistantly here but other means, such as Legendre methods, could be used) whereas for an element size under the lower bound, one of the segment end points is simply removed. These bounds are chosen so that $d_{upper} = 2d_{lower} = 0.7\Delta x$, with Δx the characteristic physical mesh size.

During the optimization process, the tangent, normal and curvature of the line at each node point are required. Yet, considering a broken line, approximations are required. Consider a steady path Γ. Following the broken line representation with N nodes, this path is fully described by the oriented sequence of points $(u_0, ..., u_{N-1})$, or equivalently by the starting point u_0, a sequence of length $(l_0, ..., l_{N-2})$ and an oriented sequence of tangent vectors $(\tau_0, ..., \tau_{N-2})$ $(\tau_i = (u_{i+1} - u_i)/l_i)$. The normal to a segment n_{s_i} is naturally defined as the normalized vector so that the basis (τ_i, n_{s_i}) is orthonormal and positively oriented. The normal to a point, n_{u_i}, is then defined by the average between the normals to both neighboring segments, weighted by their length. The normals to the starting and last points are respectively given by the normals to the first and last segments. As for the curvature, different approaches (Gauss curvature, osculating circles and using the length variations) giving similar results have been tested and Gauss curvature has been chosen [10, 45]. Let α_i be the angle between the horizontal and the vector τ_i and let ψ_i be the angle between the vectors τ_i and τ_{i+1} (see Figure 3). Then, $\forall i \in 1, .., N-2$,

$$\psi_i = \alpha_i - \alpha_{i-1} = \int_{u_{i-1}}^{u_i} \kappa(s) ds.$$

Choosing a linear interpolation for the curvature (for the segment i, $\kappa(t) = (1-t)\kappa_{i-1} + t\kappa_i$), one gets that, $\forall i \in 1, .., N-2$,

$$\psi_i = \int_0^1 (1-t)\kappa_{i-1} + t\kappa_i dt = l_i \frac{\kappa_{i-1} + \kappa_i}{2}$$

Finally, from the closing property of curvature, $\int \kappa dl = \sum_1^{N-2} \psi_i$, one can deduce the curvature $\forall i \in 1, .., N-2$ (see [10])

$$\kappa_i = \frac{2}{l_i + l_{i-1}} \psi_i.$$

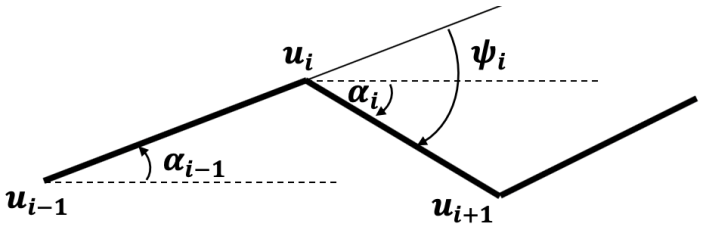
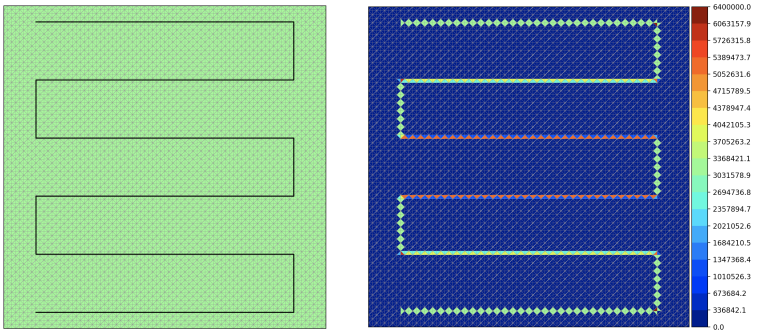


Figure 3: Computation of a discrete curvature

Second issue, interpolation between the physical mesh and the path discretization Both meshes hold information, that must be shared with each other.

- heat source: the path, defined by its discretization, carries the heat source. For each cell of the physical mesh, the length of the path is computed. To obtain a source term in the cell, this length is divided by the cell area and multiplied by the source power P, which yields a $P0-$source function (Figure 4). The thickness of the source's carrier should thus not exceed one element.
- derivative and physical quantities: the advection velocity, (see Section 3.2.2) is determined from physical quantities, defined on the physical mesh. The $P1-$functions are first turned into $P0-$ones, by assigning to each cell of the physical mesh the average of the values at its nodes. These $P0-$functions are used at the nodal values of the path.



(a) Source line (b) Source, $P0-$function

Figure 4: Discretized path (left) and corresponding heat source (right)

3.2.2 Advection method: shape optimization of lines

A representation of the line having been chosen, the optimization problem can be solved, using a descent gradient method. The line nodes are iteratively moved, improving the objective function. The remaining task is to characterize this variation, determining an advection velocity at each node. This is done by using Hadamard method of shape differentiation [25, 41].

Consider a smooth reference set Γ_0. Any admissible shape Γ is assumed to be related to the reference shape through a perturbation θ such that:

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

To have smooth enough shape deformations, the vector field is taken in $W^{1,\infty}\left(\mathbb{R}^2, \mathbb{R}^2\right)$ (the space of Lipschitz functions). A definition of shape differentiability can now be stated [3, 25].

Definition 1. *A functional $J : \mathbb{R}^2 \to \mathbb{R}$ is said to be shape differentiable at $\Gamma_0 \subset \mathbb{R}^2$ if the application $\theta \to J\left((Id+\theta)(\Gamma_0)\right)$ is Fréchet-differentiable at 0 in the Banach space $W^{1,\infty}\left(\mathbb{R}^2, \mathbb{R}^2\right)$, i.e.*

$$\begin{aligned} J\left((Id+\theta)(\Gamma_0)\right) &= J(\Gamma_0) + DJ(\Gamma_0)(\theta) + o(\theta) \quad \textit{with} \\ \lim_{\theta \to 0} \frac{|o(\theta)|}{\|\theta\|} &= 0, \end{aligned} \tag{11}$$

where $DJ(\Gamma_0)$ is a continuous linear form on $W^{1,\infty}\left(\mathbb{R}^2, \mathbb{R}^2\right)$.

Lemma 1. *Let Σ be a bounded open set of $\mathbb{R}^2$, and $\Gamma_0 \subset \Sigma$ an oriented and Lipschitz curve with end points A and B. Its unit tangent vector τ is defined with respect to its orientation, and its normal n is set so that, at each point, the couple (τ, n) is a direct orthonormal basis of $\mathbb{R}^2$. Let J be a differentiable function at Γ_0 and $\theta \in W^{1,\infty}\left(\mathbb{R}^2, \mathbb{R}^2\right)$. Then [25, 41], there exist a function $v_n : \Gamma_0 \to \mathbb{R}$ and real numbers $v_\tau(A), v_\tau(B)$ such that*

$$DJ(\Gamma_0)(\theta) = \int_{\Gamma_0} v_n(s)\theta(s)\cdot n(s)ds + v_\tau(B)\theta(B)\cdot\tau(B) + v_\tau(A)\theta(A)\cdot\tau(A). \tag{12}$$

For $f \in W^{2,1}\left(\mathbb{R}^2\right)$, consider

$$J(\Gamma) = \int_\Gamma f(s)ds.$$

Then, J is differentiable at Γ_0 and, denoting by κ the curvature of Γ_0, for all $\theta \in W^{1,\infty}\left(\mathbb{R}^2, \mathbb{R}^2\right)$,

$$DJ(\Gamma_0)(\theta) = \int_{\Gamma_0} \left[\partial_n f + \kappa f\right](s)\theta(s)\cdot n(s)ds + f(B)\theta(B)\cdot\tau(B) - f(A)\theta(A)\cdot\tau(A). \tag{13}$$

From the very general formulation given by (12), an advection velocity must be chosen. In other words, a vector field θ is looked for such that $DJ(\Gamma_0)(\theta) \leq 0$. Such a θ is a descent direction for the minimization of J. Ignoring smoothness issues, an obvious simple choice is:

$$\begin{cases} \forall s \in \Gamma_0 \setminus A, B, \quad \theta(s) = -v_n(s)n(s), \\ \theta(A) = -v_\tau(A)\tau(A) - v_n(A)n(A), \qquad \theta(B) = -v_\tau(B)\tau(B) - v_n(B)n(B). \end{cases} \tag{14}$$

It is not clear that (14) defines a smooth vector field $\theta \in W^{1,\infty}\left(\mathbb{R}^2, \mathbb{R}^2\right)$. However, we shall see in Subsection 3.3.3 that another choice of the descent direction θ is possible by introducing a different scalar product for identifying the linear form $DJ(\Gamma_0)(\theta)$.

3.3 Application

The path is now discretized following section 3.2.1 and a shape derivative is computed to allow a gradient descent optimization.

3.3.1 Dealing with the constraints, the Augmented Lagrangian Method

To include the equality constraint ($C = 0$, see (10)) in the optimization process, an augmented Lagrangian method is applied. From [42], a penalized function J is introduced and the optimization problem becomes:

$$\min_\Gamma J = L_F - \lambda_{lag}C + \frac{\mu}{2}C^2, \quad \text{such that} \quad T \quad \text{is a solution of (8).} \tag{15}$$

This formulation corresponds to the Lagrangian function of problem (10) in which the penalization term $\frac{\mu}{2}C^2$ has been added to the objective function ($\mu = 1 > 0$ fixed). An algorithm alternatively minimizing this Lagrangian function with respect to Γ and maximizing with respect to λ_{lag} would, at convergence, lead to a result satisfying the constraints. The minimization with respect to Γ is detailed in the following. The maximization of λ_{lag} is easier: at each iteration n, $(-C^n)$ is a maximizing direction and the update (16) yields an increase of the Lagrangian function. In all numerical applications, the multiplier λ_{lag} is initialized to 1.

$$\lambda_{lag}^{n+1} = \lambda_{lag}^n - \mu C^n. \tag{16}$$

3.3.2 Theoretical computation of the shape derivative

Proposition 1. *The shape derivative of the objective function* (15) *is, $\forall \theta \in W^{1,\infty}\left(\mathbb{R}^2, \mathbb{R}^2\right)$:*

$$DJ(\Gamma)(\theta) = \int_\Gamma v_n(s)\theta(s)n(s)ds + v_\tau(B)\theta(B)\tau(B) + v_\tau(A)\theta(A)\tau(A), \tag{17}$$

where

$$\begin{cases} \forall s \in \Gamma, \quad v_n(s) = -P_Q\dfrac{\partial p}{\partial n}(s) + \left(1 - P_Q p(s)\right)\kappa(s), \\ v_\tau(B) = 1 - P_Q p(B), \\ v_\tau(A) = -\left(1 - P_Q p(A)\right), \end{cases} \tag{18}$$

with $\mathbb{1}_{\Sigma_s}$ the characteristic function of Σ_s and the adjoint $p \in H^1(\Sigma)$, solution of

$$\begin{cases} -\nabla\cdot(\lambda\nabla p) + \tilde{\beta}p &= \left(-\lambda_{lag} + \mu C\right)\left(2l_\phi\left(T_\phi - T\right)^+\mathbb{1}_{\Sigma_s} - 2l_M\left(T - T_M\right)^+\right) \quad in\,\Sigma \\ \lambda\partial_n p &= 0 \qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\quad on\,\partial\Sigma. \end{cases} \tag{19}$$

Proof. To compute the shape derivative of this problem, from Céa's method [6, 15], a Lagrangian function $\mathcal{L} : U_{ad} \times H^1(\Sigma) \times H^1(\Sigma)$ is introduced, where U_{ad} is the space of admissible shapes Γ:

$$\begin{aligned}\mathcal{L}(\Gamma,\Phi,q) =& L_F(\Gamma) - \lambda_{lag} C(\Gamma,\Phi) + \frac{\mu}{2} C(\Gamma,\Phi)^2 \\ &+ \int_\Sigma \left(\lambda \nabla\Phi\cdot\nabla q + \tilde{\beta}(\Phi - T_{init})q\right) dx - \int_\Gamma P_Q q ds.\end{aligned}$$

where $C(\Gamma,\Phi)$ is the constraint (10) computed for a field Φ instead of T, which of course depends on Γ. Differentiating this Lagrangian with respect to Φ, $\forall \psi \in H^1(\Sigma)$,

$$\begin{aligned}\frac{\partial \mathcal{L}}{\partial \Phi}(\Gamma,T,q)(\psi) &= \left(-\lambda_{lag} + \mu C\right)\left[-2l_\phi \int_{\Sigma_s} \left(T_\phi - T\right)^+ dx + 2l_M \int_\Sigma (T - T_M)^+\right]\psi dx + \int_\Sigma \lambda \nabla q\cdot\nabla\psi + \tilde{\beta} q\psi dx \\ &= \left(-\lambda_{lag} + \mu C\right)\int_\Sigma \left[-2l_\phi \left(T_\phi - T\right)^+ \mathbb{1}_{\Sigma_s} + 2l_M (T - T_M)^+\right]\psi dx + \int_\Sigma \left(-\nabla\cdot(\lambda\nabla q) + \tilde{\beta} q\right)\psi dx \\ &\quad - \int_{\partial\Sigma} \lambda \partial_n q \psi ds.\end{aligned}$$

Setting to 0 the evaluation of this derivative at $\Phi = T$ (solution of (8)) amounts to solving the variational formulation of the adjoint problem given by (19). Moreover, for $\Phi = T$, one gets that, $\forall q \in H^1(\Sigma)$, $\mathcal{L}(\Gamma,T,q) = J(\Gamma)$. Thus, $\forall \theta \in W^{1,\infty}\left(\Gamma,\mathbb{R}^2\right)$, $\forall q \in H^1(\Sigma)$:

$$DJ(\Gamma)(\theta) = \frac{\partial \mathcal{L}}{\partial \Gamma}(\Gamma,T,q)(\theta) + < \frac{\partial \mathcal{L}}{\partial \Phi}(\Gamma,T,q), \frac{\partial T}{\partial \Gamma}(\theta) > .$$

Evaluating this last equation at $q = p$ (the adjoint given by (19)) gives that differentiating the objective function finally consists in computing the partial derivative with respect to the shape Γ of the Lagrangian function evaluated at $\Phi = T$ and $q = p$. The conclusion comes finally from Lemma 1. □

3.3.3 Regularization on the discretized line

Since a gradient descent method is used, it is necessary to give a Hilbert structure to the space $W^{1,\infty}\left(\mathbb{R}^2,\mathbb{R}^2\right)$. More precisely, one should find a Hilbert space Θ, with its scalar product $< \cdot,\cdot >_\Theta$, such that a gradient $J'(\Gamma)$ could be determined, $\forall \theta \in \Theta \cap W^{1,\infty}\left(\mathbb{R}^2,\mathbb{R}^2\right)$, by

$$DJ(\Gamma)(\theta) = < J'(\Gamma), \theta >_\Theta \tag{20}$$

and $-J'(\Gamma)$ would be a relevant velocity. In Subsection 3.2.2, formula (14) corresponds to the choice of a L^2-scalar product. However, the resulting path could lack smoothness. To regularize the path, an alternative scalar product is employed, corresponding to the Laplace-Beltrami flow [13]. In such a case, with $J'(\Gamma) = J'(\Gamma)_\tau \tau + J'(\Gamma)_n n \in H^1(\Gamma,\mathbb{R}^2)$ (τ and n respectively the tangent and normal to the line) and $\theta = \theta_\tau \tau + \theta_n n \in H^1(\Gamma,\mathbb{R}^2)$:

$$< J'(\Gamma), \theta >_\Theta = \int_\Gamma \eta^2 \partial_\tau J'(\Gamma)\cdot\partial_\tau\theta + J'(\Gamma)\cdot\theta dx. \tag{21}$$

where $\partial_\tau\theta = (\tau\cdot\nabla)\theta$ is the tangential derivative along the path and $\eta > 0$ is a regularization parameter (of the order of a few mesh cell sizes, see Section 3.4). Computing the gradient in this context amounts to finding $J'(\Gamma)$ such that, $\forall W = W_\tau\tau + W_n n \in H^1(\Gamma,\mathbb{R}^2)$

$$\int_\Gamma \eta^2 \partial_\tau J'(\Gamma)\cdot\partial_\tau W + J'(\Gamma)\cdot W dx = \int_\Gamma v_n n\cdot W ds + v(A)\tau(A)\cdot W(A) + v(B)\tau(B)\cdot(B)W(B). \tag{22}$$

The resulting gradient $J'(\Gamma)$ may not belong to $W^{1,\infty}(\Gamma,\mathbb{R}^2)$ anymore but is regular enough for this application [6]. The descent direction chosen is then $\theta = -J'(\Gamma)$.

Discretization of the regularization equation This continuous regularization equation has been adapted to the discretized line using a $P1-$formulation. Let f a $P1-$function and g a $P0-$one. Their integral is given by:

$$\int_\Gamma f(s)ds = \sum_{i=0}^{N-1} l_i \left(\frac{f_i + f_{i+1}}{2}\right), \qquad \int_\Gamma g(s)ds = \sum_{i=0}^{N-1} l_i g_i.$$

3.3.4 Step and projection

At each iteration $n > 0$, the advection velocity V^n is determined at each point and the line is updated by:

$$\forall i \in 0,N, \qquad P_i^{n+1} = \left(x_i^{n+1}, y_i^{n+1}\right) = P_i^n + S_i^n V_i^n.$$

The step S_i^n is given by:

$$S_i^n = \frac{\zeta^n \Delta x}{\max_i(\|V_i^n\|)},$$

with ζ^n a coefficient initialized to 1 and updated at each iteration so that, if the objective function decreases enough, namely $J(\Gamma^{n+1}) < tol^n * J(\Gamma^n)$, $\zeta^{n+1} = \min(1.2 * \zeta^n, 1)$ and else $\zeta^{n+1} = 0.6 * \zeta^n$. After this update, if some points are outside from the domain Σ they are orthogonally projected back to Σ. The tolerance tol^n is an optimization parameter, chosen by the user. Setting it different from 1 allows the objective function for increasing a bit at some iterations. In numerical applications, this tolerance is set to $tol^0 = 2$ at the beginning and is multiplied by 0.9 every 50 iterations.

3.4 Numerical results in the steady case

initialization of the line;
resolution of the heat equation and computation of the objective function and constraints;
computation of the derivatives;
for *each iteration* **do**
    update of the tolerance;
    line variation such that $\Gamma^{n+1} = \Gamma^n - \text{step}J'(\Gamma^n)$;
    resolution of the heat equation, computation of the objective function and constraint;
    if $J(\Gamma^{n+1}) < J(\Gamma^n) * tol^n$ **then**
        iteration accepted;
        Lagrange multiplier λ_{lag} updated;
        step increased ;
        update of the variables;
        computation of the derivatives;
    end
    else
        iteration refused;
        step decreased;
    end
end

Algorithm 1: Iterative algorithm to optimize the steady problem.

This section presents the numerical results for the optimization of Problem (10), using Algorithm 1. For simplicity, there is no stopping criterion in this algorithm, except a prescribed maximal number of iterations. Of course, one could add one, based on the optimality criterion. In our case, in case of convergence, the advection step will get extremely small and, even if not stopped, the algorithm will not modify much the line.

We consider here a working domain $\Sigma = [-10cm, 10cm] \times [-10cm, 10cm]$, discretized with a triangular mesh having 12800 elements (the mean length of an element is $\Delta x = 0.0035$). The accuracy of the discretized path is given by the distance ΔP between two nodes ($d_{lower} = 0.35\Delta x \leq \Delta P \leq 0.7\Delta x = d_{upper}$). In this simulation, we set the conductivity $\lambda = 0.25W.m^{-1}K^{-1}$, $\lambda_{sol} = 15W.m^{-1}K^{-1}$ and the fusion, or phase change, temperature $T_\phi = 1700K$, values corresponding to maraging steel [53]. As for the other values, they have been arbitrarily chosen to enable steady optimization: $\Delta Z = 1m$, $L = 10cm$, $P_Q = 7000W.m^{-2}$, $T_{init} = 500K$. The maximum temperature is $T_{M,\Sigma_s} = 2000K$ in the object and $T_{M,\Sigma\setminus\Sigma_s} = 1600K$ outside. The coefficients l_ϕ and l_M, from (10) are both set to 1. Since the problem is fully symmetric with respect to the (Oy)-axis, we consider, in case of a symmetric initialization, only half of the working domain, adding a sliding constraint on the last point of the path: the advection velocity at point B is projected on the (Ox)-axis. The finite element computations are run with Freefem [24] whereas the descent algorithm is run by Python. To solve this problem, 500 iterations are run on a MacBook laptop equipped with 2,3 GHz Intel Core i5 and a RAM of 16GB. For the example presented on Figure 6, the optimization process took around 3 minutes, with a mean time of 0.3s (half the domain) for each iteration. No specific efforts for optimizing the Python optimization code have been made.

3.4.1 Manufacturing the whole working domain

The first test consists in building the whole working domain Σ. Thus, $\forall x \in \Sigma, T_M(x) = T_{M,\Sigma_s}$. Three different initializations have been tested: zigzag, contour and spiral. The results are presented in Figures 6, 7, 8. A reference result is given in Figure 5. It corresponds to the smallest zigzag path with equidistant horizontal lines, centered horizontally and vertically, with its horizontal line equal to $0.16cm$, such that the phase constraint is reduced to 0. To each figure is associated the path length and the constraint C given by (10). The final numerical normalized values are given by Table 1 with:

$$N(C_\phi) = \sqrt{\frac{C_\phi}{|\Sigma| T_\phi^2}}, \qquad N(C_M) = \sqrt{\frac{C_M}{|\Sigma| T_{M,\Sigma_S}^2}}.$$

The values correspond to the full domain Σ (and not only half of it).

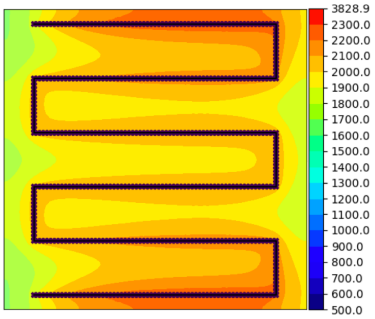


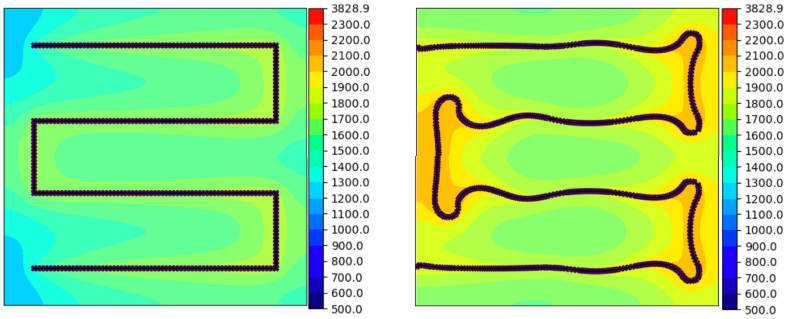
Figure 5: Temperature distribution for the non optimized reference path, steady case $L = 1.14m$, $C = 346m^2K^2$

Case	Length (m)	$N(C_\phi)$	$N(C_M)$
Reference (non optimized)	1.14	$1.97e-5$	$3.29e-2$
Zigzag initialization	1.00	$3.63e-5$	$2.50e-5$
Contour initialization	1.00	$2.20e-4$	$2.92e-4$
Spiral initialization	1.02	$1.67e-9$	$8.66e-7$

Table 1: Comparison of the final optimized results

These first results validate the algorithm with a real improvement of the objective functions and satisfaction of the constraints. Besides this validation, some points must be highlighted. The resulting path differs from one initialization to the other. Indeed, the optimization problem is not convex and there exist local minima. The technique used here is based on small variations which explains that, depending on the initialization, different solutions are found. The final quantities, however, are quite similar for each case. In particular the optimized path length is always approximately equal to 1. The zigzag and contour initializations are symmetric with respect to the horizontal axis, therefore, for the sake of reducing the computational time, a symmetry condition was imposed and the optimization process is applied only to the upper half square. Since the spiral initialization is not symmetric, the algorithm has been run on the whole working domain in such a case. Note that, for all three initializations, the constraints on the fusion and maximum temperatures are not satisfied and the initial path lengths are not the same. We use an Augmented Lagrangian method which is able to deal with such situations. In practice, in the first iterations this algorithm focuses mainly on the constraints' satisfaction, possibly at the price of increasing the objective function. This explains why, in the case of a zigzag and contour initializations, the final length is larger than the initial one, while it is indeed lower for the spiral initialization. Indeed, for the zigzag and contour initializations the fusion constraint is not satisfied, more energy is required and the path must be extended. On the contrary for the spiral case, the maximal temperature constraint is initially not satisfied, meaning that the initial path carries too much energy and the path length has to decrease. Finally, the slightly different behavior of the spiral case (the constraints are better satisfied but the optimized length is a bit longer) may be a result of not applying any symmetry condition.

Finally, let us focus on the importance of the regularization parameter η. The algorithm has been run for three different tests: $\eta = 5d_{lower}$, $\eta = 15d_{lower}$, $\eta = 20d_{lower}$ (recall that $d_{lower} = 0.35\Delta x$). The results, presented by Table 2 and Figure 9, show that this choice of regularization brings different local minima too (the case $\eta = 15d_{lower}$ is the same than in Figure 6). The "regularity" of the final path also differs and adding more industrial constraints would probably help the designer to elect one.

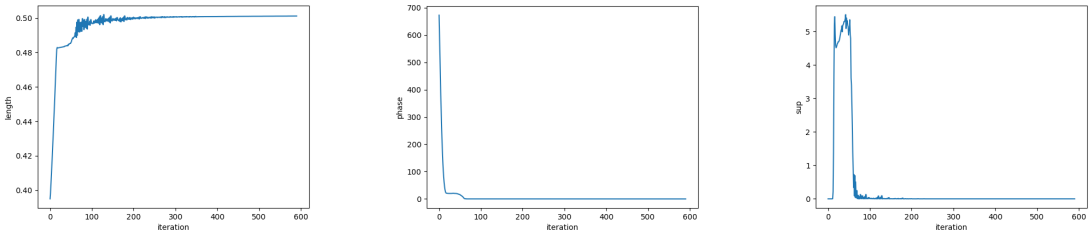


(a) Initial path, $L = 0.790m$, $C = 1345m^2K^2$

(b) Iteration 50, $L = 0.991m$, $C = 161m^2K^2$

(c) Iteration 100, $L = 0.993m$, $C = 4.63e-2m^2K^2$

(d) Iteration 500 (final) $L = 1.00m$, $C = 6.34e-5m^2K^2$



(e) Evolution of length for half of the domain

(f) Evolution of C_ϕ for half of the domain

(g) Evolution of C_M for half of the domain

Figure 6: Temperature distribution and convergence histories during the optimization process, starting from a zigzag initialization, steady case, $\eta = 15d_{lower}$

Case	Length (m)	$N(C_\phi)$	$N(C_M)$
$\eta = 5$	0.988	0	0
$\eta = 15$	1.00	$3.63e-5$	$2.50e-5$
$\eta = 20$	0.999	$1.49e-3$	$1.02e-3$

Table 2: Comparison of the final optimized results depending on the regularization coefficient, with a zigzag initialization

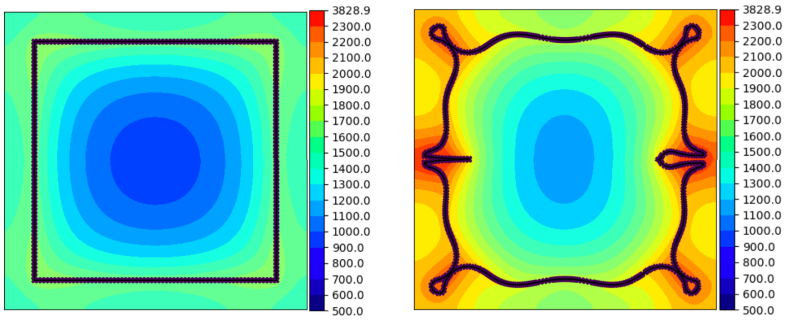
3.4.2 Manufacturing a cantilever shape

The area to build, Σ_s, is now different from the whole working domain Σ. The maximum temperature T_M is thus space dependent. The maximum temperature out of the cantilever shape Σ_S is of high importance and chosen to be favoured. Thus, the maximum temperature constraint is split such that:

$$C_M = C_{M,\Sigma_s} + 10C_{M,\Sigma\setminus\Sigma_s},$$
$$\begin{cases} C_{M,\Sigma_s} = \displaystyle\int_{\Sigma_s} \left[(T(x) - T_{M,\Sigma_s})^+\right]^2 dx, \\ C_{M,\Sigma\setminus\Sigma_s} = \displaystyle\int_{\Sigma\setminus\Sigma_s} \left[\left(T(x) - T_{M,\Sigma\setminus\Sigma_s}\right)^+\right]^2 dx. \end{cases}$$

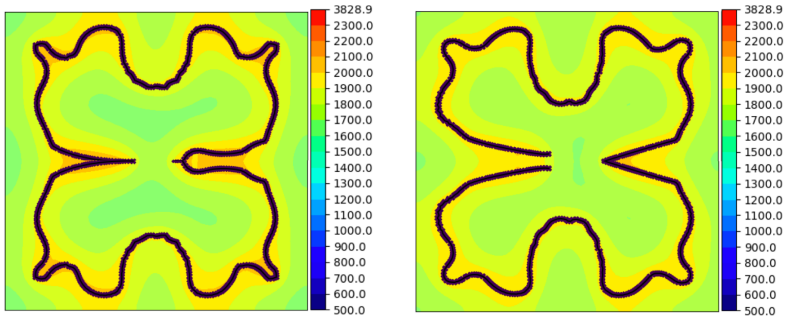
Two application cases, with their temperature constraints, are presented on Figures 10 and 11. In these cases, the mesh is adapted to the cantilever and respectively contains 1097 and 1661 triangles. The mean computational time for one iteration is reduced to 0.15s (for half the domain).

The results for the first test case (Figure 10) are given by Figures 12 and 13. Here again, symmetry applies



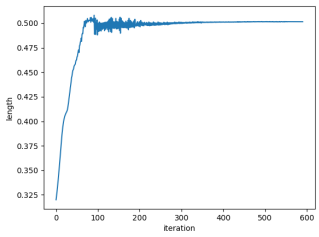
(a) Initial path
$L = 0.64m$, $C = 6537m^2K^2$

(b) Iteration 50
$L = 0.931m$, $C = 2128m^2K^2$

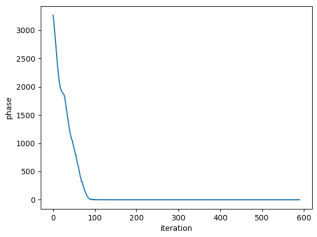


(c) Iteration 100
$L = 0.999m$, $C = 4.99m^2K^2$

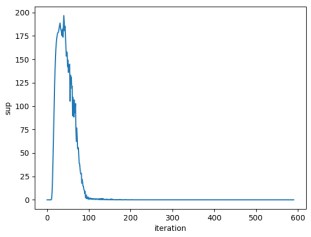
(d) Iteration 500 (final)
$L = 1.00m$, $C = 8.78e-3m^2K^2$



(e) Evolution of length for half of the domain



(f) Evolution of C_ϕ for half of the domain



(g) Evolution of C_M for half of the domain

Figure 7: Temperature distribution and convergence histories during the optimization process, starting from a contour initialization, steady case, $\eta = 15d_{lower}$

and only one half of the domain is considered. The normalized results are compared in Table 3 with

$$N(C_\phi) = \sqrt{\frac{C_\phi}{V_{\Sigma_s} T_\phi^2}}, \qquad N(C_{M,\Sigma_s}) = \sqrt{\frac{C_{M,\Sigma_s}}{V_{\Sigma_s} T_{M,\Sigma_S}^2}}, \qquad N(C_{M,\Sigma\setminus\Sigma_s}) = \sqrt{\frac{C_{M,\Sigma\setminus\Sigma_s}}{V_{\Sigma\setminus\Sigma_s} T_{M,\Sigma\setminus\Sigma_s}^2}}.$$

For this first object, the volume of Σ_s is $V_{\Sigma_s} = 2.75e-02m^2$ and the volume of $\Sigma\setminus\Sigma_s$ is $V_{\Sigma\setminus\Sigma_s} = 1.25e-02m^2$.

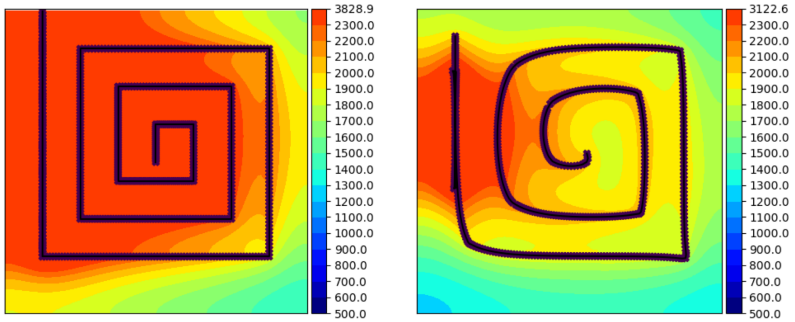
Case	Length (m)	$N(C_\phi)$	$N(C_{M,\Sigma_s})$	$N(C_{M,\Sigma\setminus\Sigma_s})$
Hat	0.941	$9.46e-3$	$8.38e-4$	$2.70e-3$
Contour	0.937	$1.29e-2$	$3.74e-4$	$3.64e-3$

Table 3: Comparison of the final optimized results for the one hole cantilever

The results of the second test case (Figure 11) are given by Figures 14 and 15. Here again, symmetry applies and only one half of the domain is considered. The normalized results are compared in Table 4, with $V_{\Sigma_s} = 2.10e-02m^2$ and $V_{\Sigma\setminus\Sigma_s} = 1.90e-02m^2$.

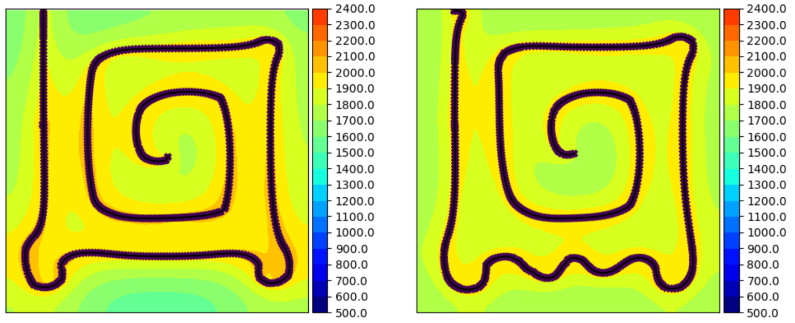
Case	Length (m)	$N(C_\phi)$	$N(C_{M,\Sigma_s})$	$N(C_{M,\Sigma\setminus\Sigma_s})$
Hat	0.828	$5.81e-2$	$4.44e-3$	$1.30e-2$
Contour	0.843	$2.73e-2$	0	$6.12e-3$

Table 4: Comparison of the final results for the three holes cantilever



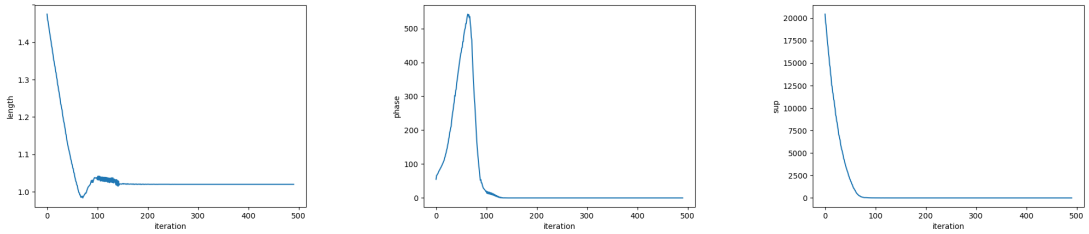
(a) Initial path
$L = 1.47m, C = 20507m^2K^2$

(b) Iteration 50
$L = 1.07m, C = 2475m^2K^2$



(c) Iteration 100
$L = 1.03m, C = 23.0m^2K^2$

(d) Iteration 500 (final)
$L = 1.02m, C = 1.08e-8m^2K^2$

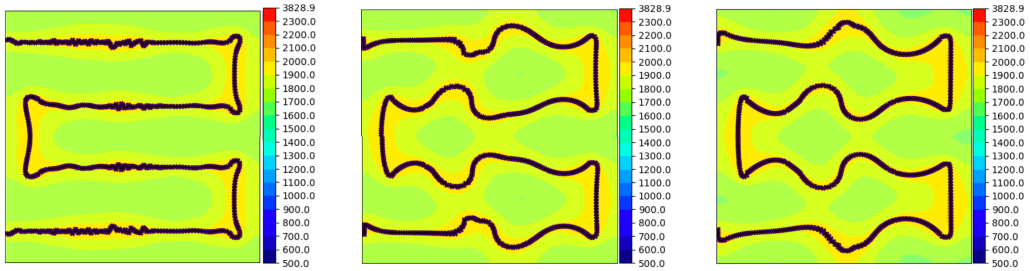


(e) Evolution of length

(f) Evolution of C_ϕ

(g) Evolution of C_M

Figure 8: Temperature distribution and convergence histories during the optimization process, starting from a spiral initialization, steady case, $\eta = 15d_{lower}$



(a) $\eta = 5d_{lower}$

(b) $\eta = 15d_{lower}$

(c) $\eta = 20d_{lower}$

Figure 9: Temperature distribution of the final iteration (iteration 500) from a zigzag initialization depending on the regularization coefficient η

These two cantilever test cases validate quite well the algorithm. Indeed, the path really adapts to the constraints given. Different optimized paths appear depending on the initialization. Some of them seem to fit the constraints whereas others, such as the result presented by Figure 14, obviously do not. However, this result still corresponds to a local minimum. One should change the optimization parameters or even the augmented Lagrangian algorithm to solve this difficulty.

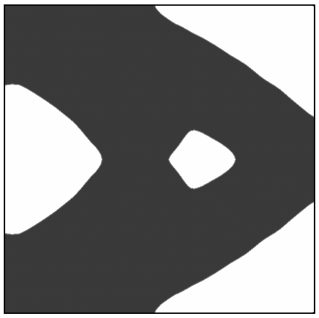


Figure 10: One hole cantilever, steady case

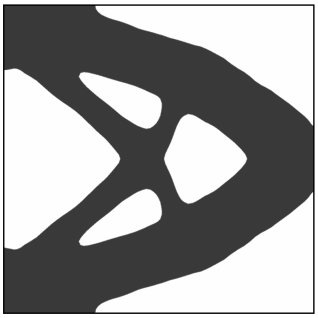
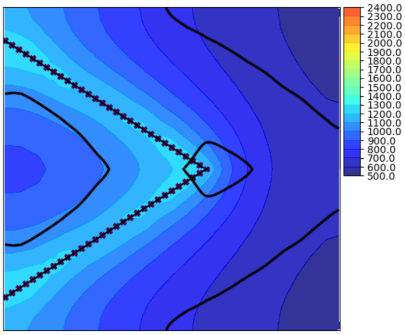
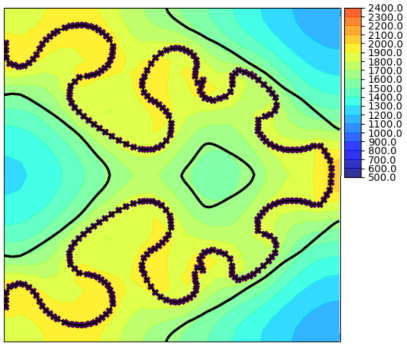


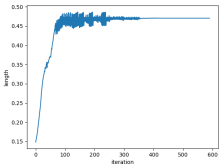
Figure 11: Three holes cantilever, steady case



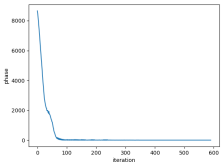
(a) Initial path



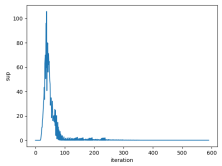
(b) Optimized path



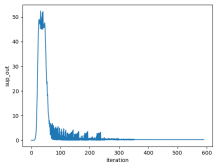
(c) Evolution of length for half of the domain



(d) Evolution of C_ϕ for half of the domain

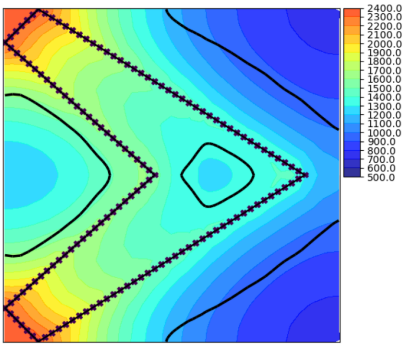


(e) Evolution of C_{M,Σ_s} for half of the domain

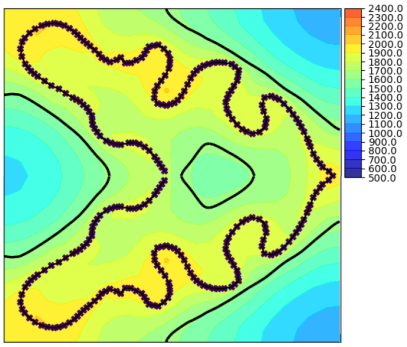


(f) Evolution of $C_{M,\Sigma\setminus\Sigma_s}$ for half of the domain

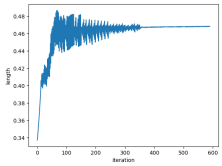
Figure 12: Initial and final temperature distribution as well as convergence histories, path optimization for the one hole cantilever from a hat initialization ($\eta = 10d_{lower}$)



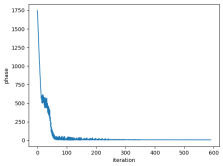
(a) Initial path



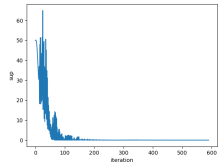
(b) Optimized path



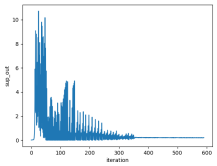
(c) Evolution of length for half of the domain



(d) Evolution of C_ϕ for half of the domain

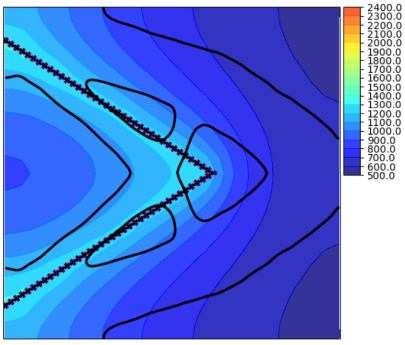


(e) Evolution of C_{M,Σ_s} for half of the domain

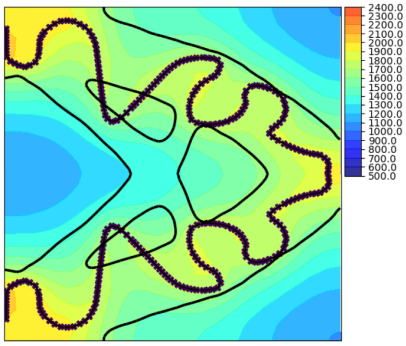


(f) Evolution of $C_{M,\Sigma\setminus\Sigma_s}$ for half of the domain

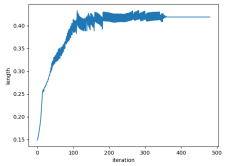
Figure 13: Initial and final temperature distribution as well as convergence histories, path optimization for the one hole cantilever from a contour initialization ($\eta = 5d_{lower}$)



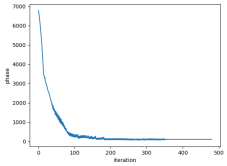
(a) Initial path



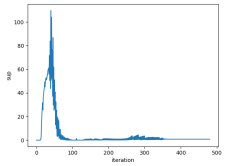
(b) Optimized path



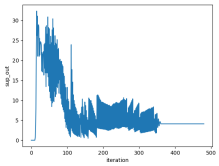
(c) Evolution of length for half of the domain



(d) Evolution of C_ϕ for half of the domain

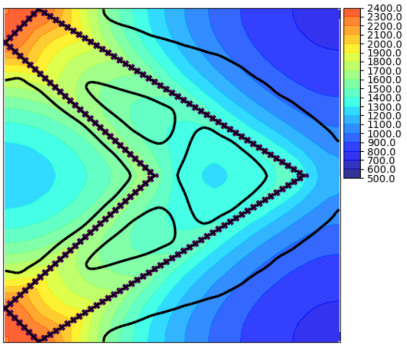


(e) Evolution of C_{M,Σ_s} for half of the domain

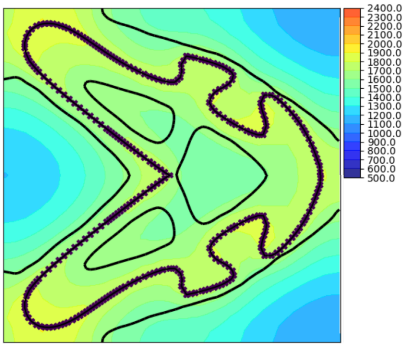


(f) Evolution of $C_{M,\Sigma\setminus\Sigma_s}$ for half of the domain

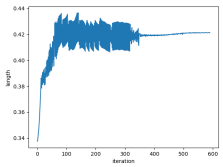
Figure 14: Initial and final temperature distribution as well as convergence histories, path optimization for the three holes cantilever from a hat initialization ($\eta = 5d_{lower}$)



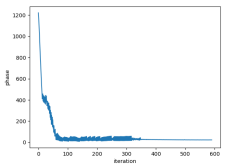
(a) Initial path



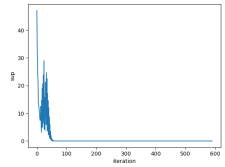
(b) Optimized path



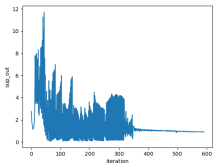
(c) Evolution of length for half of the domain



(d) Evolution of C_ϕ for half of the domain



(e) Evolution of C_{M,Σ_s} for half of the domain



(f) Evolution of $C_{M,\Sigma\setminus\Sigma_s}$ for half of the domain

Figure 15: Initial and final temperature distribution and convergence histories, path optimization for the three holes cantilever from a hat initialization ($\eta = 5d_{lower}$)

4 Unsteady case: optimal control of the path

We now consider the unsteady problem modeled in Section 2 which would be the case in laser based powder bed fusion. The source is a Gaussian function, defined by (2), with center $u(t,x)$ moving along the oriented path Γ, satisfying the trajectory equation (3). We aim at constraining the temperature, solution of the heat equation (1) by controlling the source center and thus the path Γ. The final time t_F is also to be determined, such that $t_F \in [0, T_F]$ (bounding the final time t_F by $T_F > 0$ fixed is an industrial requirement). In the remaining of this work, we assume that both the scalar velocity V and the power P are fixed. As mentioned in Section 2.2, this means that we cannot switch the source on and off during the process, making the final time be the manufacturing time.

4.1 Continuous optimal control of the path

Unlike the steady case, the unsteady problem not only involves a PDE but interweaves it with the trajectory equation, an ODE depending on time. This complicates the problem, preventing from using the very convenient shape optimization theory. The approach is thus modified, relying on optimal control methods for mixed PDE and ODE problems [17, 29, 55, 56] and especially on the approach developed in [56].

4.1.1 Path description

As already mentioned we consider in this case the evolution of the source's position with respect to time. The support of the source term is not the whole line anymore but travels along the line. Thus, the path description is modified: it is characterized by its starting point $\tilde{u}$, its length (or equivalently its final time t_F) and the angle $\alpha(t)$ between its tangent direction and the horizontal axis (see Figure 16). Thus, in the sequel, the optimization variable will be the angle α,the starting point $\tilde{u}$ and the final time t_F. Another possible choice would be the position $x(t)$ on the line: this will be investigated in future work (see Section 6).

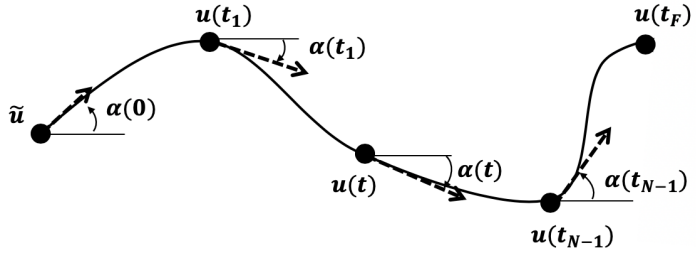


Figure 16: Continuous path description

In this context, the source point trajectory equation becomes:

$$\begin{cases} \dot{u}(t) = V\tau(\alpha(t)) = V(\cos\alpha(t), \sin\alpha(t)), & t \in (0, t_F) \\ u(0) = \tilde{u}. \end{cases} \tag{23}$$

4.1.2 Problem definition

We consider the optimization problem (7). The line Γ is now fully described by the angle $\alpha \in L^2((0,t_F))$, the starting point $\tilde{u} \in \Sigma$ and the final time $t_F \in (0, T_F)$. These three variables constitute the new optimization parameters.

To deal with the constraints, we transform the optimization problem (7) into an unconstrained optimization problem, setting a new objective function J such that:

$$J(t_F, \alpha, \tilde{u}) \quad = l_{t_F} t_F + l_\phi C_\phi + l_M C_M \tag{24}$$

with C_ϕ and C_M defined by (4) and (5) depend on u and T, and where the Lagrange multipliers l_{t_F}, l_ϕ, l_M are kept constant during the optimization process (in a future work the Lagrange multipliers will vary, allowing to treat the original constrained problem (7)). The unconstrained optimization problem is the following:

$$\min_{t_F \in [0,T_F],\, \alpha \in \mathcal{A},\, \tilde{u} \in \Sigma} J(t_F, \alpha, \tilde{u})$$
$$\text{such that} \begin{cases} \alpha \in \mathcal{A} = L^2((0,t_F)) \\ u \in U = \mathcal{C}^0\left([0,t_F], \mathbb{R}^2\right) \text{ solution of (23)}, \\ T \in \mathcal{T} = L^2\left([0,t_F], H^1(\Sigma)\right) \cap \mathcal{C}^0\left([0,t_F], L^2(\Sigma)\right) \text{ solution of (1)}. \end{cases} \tag{25}$$

4.1.3 Differentiation of the objective function

In the unsteady case, the descent gradient algorithm is simple since it is merely a classical parametric optimization with respect to the angle, the starting point and the final time. However, the computation of the derivatives is slightly more complex.

Proposition 2. *The derivatives of the objective function* (24) *are:*

$$D_\alpha J(t_F, \alpha, \tilde{u})(\delta\alpha) = -\int_0^{t_F} V\tau'(\alpha(t)) \cdot w(t)\delta\alpha(t)dt, \tag{26}$$

with $\delta\alpha \in \mathcal{A}$ *a differentiation direction and* $\tau'(\alpha(t)) = (-\sin\alpha(t), \cos\alpha(t))$,

$$D_{\tilde{u}} J(t_F, \alpha, \tilde{u})(\delta\tilde{u}) = -w(0) \cdot \delta\tilde{u}, \tag{27}$$

with $\delta\tilde{u} \in \mathbb{R}^2$ *a differentiation direction, and*

$$\begin{aligned} D_{t_F} J(t_F, \alpha, \tilde{u}) &= l_{t_F} + l_M \int_\Sigma \left[(T(t_F, x) - T_M)^+ \right]^2 dx \\ &+ \frac{l_\phi}{t_F \mathfrak{p}} \int_{\Sigma_s} 2 \left(T_\phi - N_\mathfrak{p}(t_F, T) \right)^+ N_\mathfrak{p}(t_F, T)^{1-\mathfrak{p}} \left(N_\mathfrak{p}(t_F, T)^\mathfrak{p} - T(t_F)^\mathfrak{p} \right) dx. \end{aligned} \tag{28}$$

In the above formulas, $w \in U$ *is the adjoint for the ODE* (23)*, solution of*

$$\begin{cases} \dot{w}(t) = \dfrac{2}{r_c^2} \displaystyle\int_\Sigma \tilde{q}(u(t), x) p(t, x) \left(u(t) - x \right) dx \quad t \in (0, t_F), \\ w(t_F) = 0, \end{cases} \tag{29}$$

with r_c *parameter related to the beam's focusing (see* (2)*) and* $p \in \mathcal{T}$ *is the adjoint of the heat equation, solution of*

$$\begin{cases} -\tilde{\rho}\partial_t p + \nabla \cdot (\lambda \nabla p) + \tilde{\beta} p = -2 l_M (T - T_M)^+ \\ \qquad + \dfrac{2 l_\phi}{t_F} \left(T_\phi - N_\mathfrak{p}(t_F, T) \right)^+ N_\mathfrak{p}(t_F, T)^{1-\mathfrak{p}} T^{\mathfrak{p}-1} \mathbb{1}_{\Sigma_s} & in\,(0, t_F) \times \Sigma \\ \lambda \partial_n p = 0 & on\,(0, t_F) \times \partial\Sigma \\ p(t_F, x) = 0 & in\,\Sigma. \end{cases} \tag{30}$$

Proof. As in the steady context, we rely on Cea's Lagrangian approach to compute the derivatives by using the adjoint method [15]. Recalling the definition (25) of the spaces $\mathcal{A}, U, \mathcal{T}$, a Lagrangian function $\mathcal{L} : [0, T_F] \times \mathcal{A} \times \Sigma \times U \times U \times \mathcal{T} \times \mathcal{T} \to \mathbb{R}$, involving the variational formulations of both the heat and trajectory equations, is introduced:

$$\mathcal{L}(t_F, \alpha, \tilde{u}, v, w, \Phi, p) \quad = J(t_F, \alpha, \tilde{u}) + \tilde{\mathcal{L}}(t_F, \alpha, \tilde{u}, v, w, \Phi, p)$$

with

$$\begin{aligned} \tilde{\mathcal{L}}(t_F, \alpha, \tilde{u}, v, w, \Phi, p) &= \int_0^{t_F} \left(\dot{v} - V\tau(\alpha(t)) \right) \cdot w(t) dt + (v(0) - \tilde{u}) \cdot w(0) \\ &+ \int_0^{t_F} \int_\Sigma \left(\left(\tilde{\rho}\partial_t \Phi + \tilde{\beta}\Phi - \tilde{\beta} T_{init} - \tilde{q}(v(t)) \right) p + \lambda \nabla\Phi \cdot \nabla p \right) dx dt \\ &+ \int_\Sigma \tilde{\rho}(0, x) \left(\Phi(0, x) - T_{init}(x) \right) p(0, x) dx. \end{aligned}$$

Then, $\forall (t_F, \alpha, \tilde{u}) \in [0, T_F] \times \mathcal{A} \times \Sigma$, $\forall (w, p) \in U \times \mathcal{T}$, with $u \in U$ and $T \in \mathcal{T}$ respectively solutions of (23) and (1), $\tilde{\mathcal{L}}(t_F, \alpha, \tilde{u}, u, w, T, p) = 0$ and thus:

$$\mathcal{L}(t_F, \alpha, \tilde{u}, u, w, T, p) = J(t_F, \alpha, \tilde{u}).$$

□

Differentiating with respect to the control parameters $X = (t_F, \alpha, \tilde{u}) \in [0, T_F] \times \mathcal{A} \times \Sigma$, for any differentiation direction δX,

$$\begin{aligned} \frac{d\mathcal{L}}{dX}(X, u, w, T, p)(\delta X) &= \partial_X \mathcal{L}(X, u, w, T, p)(\delta X) \\ &+ < \partial_v \mathcal{L}(X, u, w, T, p), \partial_X u(\delta X) > + < \partial_\Phi \mathcal{L}(X, u, w, T, p), \partial_X T(\delta X) > . \end{aligned} \tag{31}$$

Setting to 0 the derivatives of the Lagragian function $\mathcal{L}$ with respect to v and Φ, evaluated at $\Phi = T$ (solution of (1)) and $v = u$ (solution of (23)), comes down to solving the adjoint equations (29) and (30). Particularizing w to be solution of (29) and p solution of (30), the differentiation finally results in

$$d_X J(X) = \partial_X \mathcal{L}(X, u, w, T, p). \tag{32}$$

Classically, (31) yields the derivatives with respect to α and $\tilde{u}$. As for the differentiation with respect to the

final time t_F, one gets:

$$\begin{aligned}
&\partial_{t_F}\mathcal{L}(t_F,\alpha,\tilde{u},u,w,T,p) = l_{t_F} \\
&+ l_\phi \int_{\Sigma_s} -2\left(T_\phi - N_\mathfrak{p}(t_F,T)\right)^+ N_\mathfrak{p}(t_F,T)^{1-\mathfrak{p}} \left(\frac{-1}{\mathfrak{p}t_F^2}\int_0^{t_F}|T|^\mathfrak{p}dt + \frac{1}{\mathfrak{p}t_F}T(t_F)^\mathfrak{p}\right)dx \\
&+ l_M \int_\Sigma \left[(T(t_F,x)-T_M)^+\right]^2 dx \\
&+ (\dot{u}(t_F) - V\tau(\alpha(t_F))) \cdot w(t_F) \\
&+ \int_\Sigma \left(\left(\tilde{\rho}\partial_t T(t_F,x) + \tilde{\beta}T(t_F,x) - \tilde{q}(u(t_F),x)\right)p(t_F,x) + \lambda\nabla T(t_F,x)\cdot\nabla p(t_F,x)\right)dx.
\end{aligned} \tag{33}$$

It is relevant to notice that for smooth solutions in time, the variational formulation of (1) and (23) are satisfied at the final time t_F. Thus, the two last terms of (33) cancel and we get the result (28).

Remark 3. *Proposition 2 gives the derivative of the problem with respect to the control variables, α, $\tilde{u}$, t_F. The partial derivatives with respect to $t_F \in \mathbb{R}$ and $\tilde{u} \in \mathbb{R}^2$ need no post processing since they belong to a finite dimensional space. However, to get a descent direction for α, the Hilbert structure of $\mathcal{A} = L^2(0,t_F)$ must be recalled. The gradient $\nabla_\alpha J$ is derived from the differential by:*

$$D_\alpha J(\alpha)(\delta\alpha) = \int_0^{t_F} \nabla_\alpha J(\alpha)\delta\alpha dt. \tag{34}$$

4.2 Discrete optimal control of the path

4.2.1 Discrete path description

As in the steady context (Section 3), a fixed physical mesh and a path described by a broken line (Figure 17) are chosen. Yet, if this broken line was previously defined by its node points, it is now characterized by the segments angles with the (Ox)-axis and the starting point $\tilde{u}$ of the line (Figure 17). Each segment element has the same fixed length l. Fixing this size highly simplifies the resolution of the heat equation: its time step corresponds to the time required by the source to go from one node point to the next one on the discretized path and thus, $\Delta t = l/V$. The price to pay is that the final time t_F is no longer a continuous variable but a discrete one proportional to the number of segments.

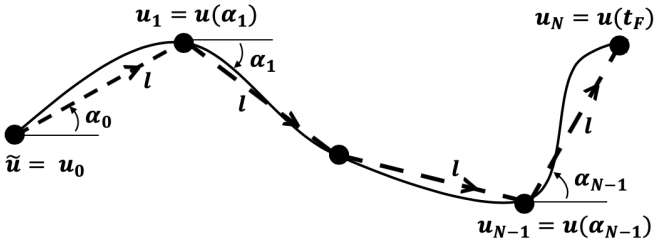


Figure 17: Continuous path (solid line) and its discretization (dotted line)

4.2.2 Discrete problem definition and adjoint equations

Time is thus discretized by a sequence $\{0 = t_0, ..., t_N\}$ with t_0 the initial time, $t_N = t_F$ and $\forall i \in [\![1,N]\!]$, $t_i = t_{i-1} + \Delta_t$ $\left(\Delta t = \frac{l}{V}\right)$. At each time step t_i we associate an angle $\alpha_i = \alpha(t_i)$, a temperature $T_i = T(t_i)$, and a path position $u_i = u(t_i)$. The heat equation (1) is discretized with respect to time by an implicit Euler scheme whereas an Euler forward scheme is used for the trajectory equation (23):

$$\begin{cases} \dfrac{u_i - u_{i-1}}{\Delta t} = F(\alpha_{i-1}) & \forall i \in \{1,..,N-1\} \\ u_0 = \tilde{u} & . \end{cases} \tag{35}$$

and for $i \in \{1,..,N\}$,

$$\begin{cases} \tilde{\rho}\dfrac{T_i(x)-T_{i-1}(x)}{\Delta t} - \nabla\cdot(\lambda\nabla T_i(x)) + \tilde{\beta}(T_i(x)-T_{init}) = \tilde{q}(u_{i-1},x,) & \forall x\in\Sigma \\ \lambda\partial_n T_i(x) = 0 & \forall x\in\partial\Sigma \\ T_0(x) = T_{init}(x) & \forall x\in\Sigma, \end{cases} \tag{36}$$

In numerical practice, we first solve the ODE (35) for index $i-1$ and then the PDE (36) for i. The differentiation process described in the continuous case (Section 4.1) can be also be applied in the discrete case (see [5] for a similar approach in shape optimization). The corresponding adjoint equations are backward. The discrete equations for the heat adjoint $p = (p_0, ..., p_N)$ are, $\forall i \in 0, ...N-1$:

$$\begin{cases} \tilde{\rho}\dfrac{p_i - p_{i+1}}{\Delta t} - \nabla\cdot(\lambda\nabla p_i) + \tilde{\beta}p_i = -2l_M(T_i - T_M)^+ \\ \qquad\qquad + \dfrac{2l_\phi}{t_F}\left(T_\phi - N_\mathfrak{p}(t_F,T)\right)^+ N_\mathfrak{p}(t_F,T)^{1-\mathfrak{p}} T_i^{\mathfrak{p}-1}\mathbb{1}_\Sigma & \text{in}\,\Sigma \\ \lambda\partial_n p_i = 0 & \text{on}\,\partial\Sigma. \end{cases}$$

$$\begin{cases} \dfrac{\tilde{\rho}}{\Delta t}p_N - \nabla\cdot(\lambda\nabla p_N) + \tilde{\beta}p_N = -2l_\phi(T_N - T_M)^+ \\ \qquad\qquad + \dfrac{2l_\phi}{t_F}\left(T_\phi - N_p(t_F,T)\right)^+ N_p(t_F,T)^{1-p} T_N^{p-1} & \text{in}\,\Sigma \\ \lambda\partial_n p_N = 0 & \text{on}\,\partial\Sigma. \end{cases} \tag{37}$$

The discrete equations for the ODE adjoint $w = (w_0, ..., w_{N-1})$ are $\forall i \in \{1,..,N-1\}$

$$\begin{cases} \dfrac{w_{i+1}-w_i}{\Delta t} = \dfrac{2}{r_c^2}\displaystyle\int_\Sigma \tilde{q}(u_i)p_i(u_i - x)dx, \\ \dfrac{w_{N-1}}{\Delta t} = -\dfrac{2}{r_c^2}\displaystyle\int_\Sigma \tilde{q}(u_{N-1})p_N(u_{N-1} - x)dx. \end{cases} \tag{38}$$

4.2.3 Update of the angle α

The derivative with respect to α is given by (34), namely

$$\nabla_\alpha J(\alpha(t)) = -V\tau'(\alpha(t))\cdot w(t).$$

Upon discretization, the angle is updated by, $\forall i \in \{0,...,N-2\}$

$$\alpha_i^{n+1} = \alpha_i^n + S_\alpha^n V\Delta t\tau'(\alpha_i)w_{i+1},$$

with a positive step $S_\alpha^n > 0$. The descent gradient algorithm requires small variations to ensure that the optimized function decreases. The angles's variations must be kept small and the step S_α^n is chosen so that

$$\|\alpha^{n+1} - \alpha^n\|_\infty = \alpha_{ref}^n.$$

The reference angle α_{ref}^n is initialized to $\alpha_{ref}^0 = 3$. At each iteration n, if the objective function satisfies $J(\alpha^{n+1}) < tol^n * J(\alpha^n)$, then,

$$\alpha_{ref}^{n+1} = \min\left(\alpha_{ref}^0, 1.2\alpha_{ref}^n\right),$$

else,

$$\alpha_{ref}^{n+1} = 0.6\alpha_{ref}^n,$$

where $tol^n \geq 1$ is a tolerance which is initialized as 2 and multiplied by 0.9 every 50 iterations.

4.2.4 Update of the starting point $\tilde{u}$

The starting point $\tilde{u}$ is updated after the updates of the other points u_i. In particular, the average displacement of those nodes has been computed as

$$\delta u^n = \frac{1}{N-1}\sum_{i=1}^{N-1}|u_i^{n+1} - u_i^n|,$$

where $|.|$ is the Euclidian distance in $\mathbb{R}^2$. The derivative with respect to $\tilde{u}$ is given by (27). At each iteration n,

$$\tilde{u}^{n+1} = \tilde{u}^n + S_{\tilde{u}}^n\delta u^n\frac{w_0}{\|w_0\|},$$

with a positive step $S_{\tilde{u}}^n > 0$. This descent step is updated as follows: at each iteration n, if the objective function satisfies $J(\tilde{u}^{n+1}) < tol^n * J(\tilde{u}^n)$ (see the definition of the tolerance in Section 4.2.3),

$$S_{\tilde{u}}^{n+1} = \min\left(S_{\tilde{u}}^0, 1.2S_{\tilde{u}}^n\right),$$

else,

$$S_{\tilde{u}}^{n+1} = 0.6S_{\tilde{u}}^n.$$

This update of the starting point is not done at each iteration but only every 3 iterations. Notice that, but for the starting point of the line, each point moves twice. The first move corresponds to the angle's update. The second move corresponds to the starting point's update, which actually induces a rigid body motion.

4.2.5 Update of the final time t_F

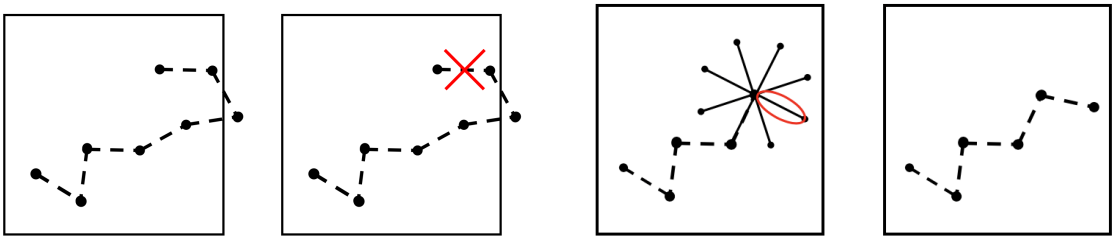
Recall that the final time t_F is not continuous anymore but a discrete variable linked to the number of segments of the path. We define a discrete descent step $S_{t_F}^n$ which is a positive integer number initialized at 5. This number $S_{t_F}^n$ determines the number of segments which are added or removed at the end of the discretized path for each iteration n. If the time derivative (28) is negative or if any node point of the line is out of the domain Σ, $S_{t_F}^n$ segments are removed. Else a straight line of $S_{t_F}^n$ segments (of size l) are added. The direction of the added segments can be chosen within 8 values which are in addition of α_{N-1}^n (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 18). The descent step $S_{t_F}^n$ is updated as follows: if after iteration n, the objective function satisfies $J(\tilde{u}^{n+1}) < tol^n * J(\tilde{u}^n)$ (see the definition of the tolerance in Section 4.2.3), then

$$S_{t_F}^{n+1} = \min(S_{t_F}^n + 1, 5),$$

otherwise,

$$S_{t_F}^{n+1} = \max(S_{t_F}^n - 1, 0).$$

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 node point is out of the working domain

(b) Adding a segment

Figure 18: Final time t_F update process

5 Numerical application in the unsteady context

5.1 Algorithmic details

For our numerical simulations corresponding to Problem (25) we use Algorithm 2. As in the steady case, for simplicity, no stopping criterion has been implemented in this algorithm, except a prescribed maximal number of iterations.

We consider the working domain $\Sigma = [-10cm, 10cm] \times [-10cm, 10cm]$, discretized with a triangular mesh having 3200 elements (the averaged length of an element is $\Delta x = 0.0071$). The length of each line segment is $0.5\Delta x$. In this simulation, the conductivity has been chosen as $\lambda = 1000W.m^{-1} \cdot K^{-1}$, the heat capacity $c = 450J.kg^{-1}.\circ C^{-1}$, $\rho = 8000kg.m^{-3}$, the phase change temperature $T_\phi = 773K$, $T_{M,\Sigma_s} = 3773K$, $T_{M,\Sigma\setminus\Sigma_s} = 673K$ and $T_{init} = 303K$. We set $\Delta Z = 10cm$, $L = 10cm$, $P = 76.8 * 10^9 W$. The coefficient r_c is chosen to be 0.0001 and the power $\mathfrak{p} = 8$. The velocity is $V = 1m.s^{-1}$. If the values of the density and heat capacity correspond to real values for solid maraging steel [5, 53], the other values have been chosen to help validating the algorithm. Considering only realistic values is part of the perspectives described in Section 6. For the numerical applications, the Lagrange multipliers are set to

$$l_{t_F} = 1, \qquad l_\phi = 1, \qquad l_M = 1. \tag{39}$$

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;
    line variation;
    resolution of the heat equation, computation of the objective function and constraint;
    if *improvement* $\left(J^{n+1} < tol^n J^n\right)$ **then**
        iteration accepted;
        step coefficients increased ($\alpha_{ref}^{n+1} = \min\left(\alpha_{ref}^0, 1.2\alpha_{ref}^n\right)$, $S_{\tilde{u}}^{n+1} = \min\left(\S_{tildeu}^0, 1.2S_{\tilde{u}}^n\right)$ if update of the starting point and $S_{t_F}^{n+1} = \min(S_{t_F}^n + 1, 5)$ if update of the final time);
        update of the variables;
        computation of the derivatives;
    end
    else
        iteration refused;
        step coefficient refused ($\alpha_{ref}^{n+1} = 0.6\alpha_{ref}^n$, $S_{\tilde{u}}^{n+1} = 0.6S_{\tilde{u}}^n$ if update of the starting point and $S_{t_F}^{n+1} = \max(S_{t_F}^n - 1, 0)$ if update of the final time);
    end
end

Algorithm 2: Iterative algorithm to optimize the unsteady problem

With such a choice of Lagrange multipliers and since the numerical values of the constraints are by far larger than the typical final time, the optimization process mostly tries to satisfy the two constraints $C_\phi = 0$ and $C_M = 0$ and not so much to minimize the objective function, namely the final time.

5.2 Manufacturing the whole working domain

The first test case consists in building the whole working domain Σ and thus $\forall x \in \Sigma$, $T_M(x) = T_{M,\Sigma_s}$. Four different initializations are tested: zigzag, contour, spiral, and straight line. In each case, 1000 iterations are performed and the path has converged. In each case, two functions are plotted for the first and final iterations. The left plots represent the integrand of the phase constraint

$$\left(T_\phi - N_\mathfrak{p}(t_F, T)\right)^+,$$

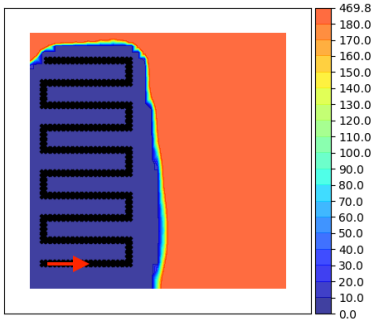
and the right plots the integrand of the maximum temperature constraint

$$\int_0^{t_F} (T - T_M)^+ \, dt.$$

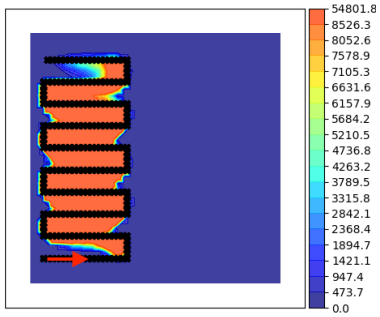
In all figures, a larger box than the working domain is plotted so we can see if the path goes away from the working domain (the working domain corresponds to the colored part). The evolution of the length and the logarithm of both constraints with respect to iterations are also given. As for the computational time, around one hour is required to run on a a MacBook laptop equipped with 2,3 GHz Intel Core i5 and a RAM of 16GB for 1000 iterations, giving Figure 19. The computational time depends on the path since the number of linear problems to solve is twice the number of points constituting the path. Normalized numerical results are summed up in Table 5 with the following definition of normalized constraints:

$$N(C_\phi) = \sqrt{\frac{C_\phi}{|\Sigma| T_\phi^2}}, \qquad N(C_M) = \sqrt{\frac{C_M}{|\Sigma| T_M^2 V L_F}}.$$

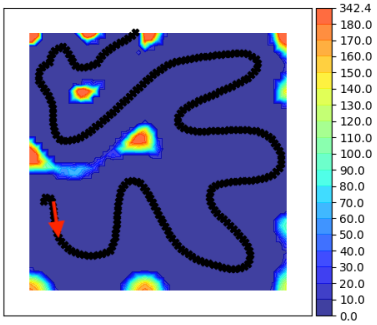
Figures 19, 20, 21 and 22 show the path after 1000 iterations. The constraints are almost fully satisfied (see Table 5). Clearly the path has adapted to the requirements, validating the algorithm. These conclusions lead to interesting perspectives. At short term, the optimization algorithm could be improved and its parameters better tuned in order to speed up convergence. Then, the model could be expanded to more realistic constraints, to three dimensions or to allow for source speed and power optimization.



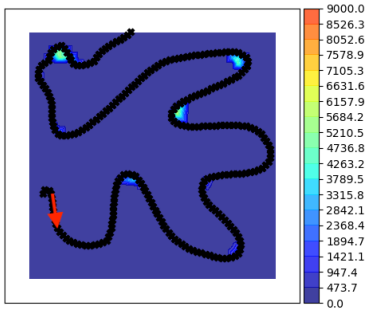
(a) First iteration, phase constraint, $L = 0.83m$, $C_\phi = 4226K^2m^2$



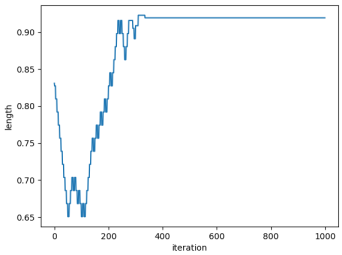
(b) First iteration, maximum constraint, $L = 0.83m$, $C_M = 346K^2m^2s$



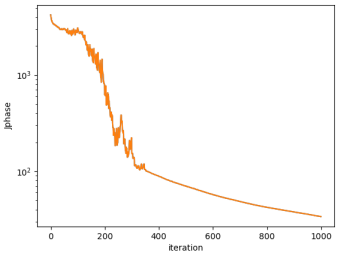
(c) Final iteration, phase constraint, $L = 0.92m$, $C_\phi = 33.7K^2m^2$



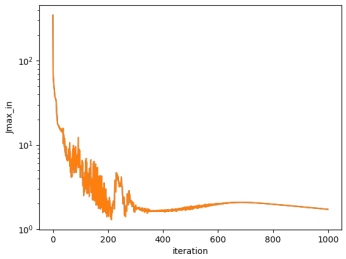
(d) Final iteration, maximum constraint, $L = 0.92m$, $C_M = 1.71K^2m^2s$



(e) Evolution of length



(f) Evolution of $\ln(C_\phi)$



(g) Evolution of $\ln(C_M)$

Figure 19: Zigzag initialization: integrand of the phase (a, c) and maximum (b, d) constraints for the first and last iterations and convergence histories (e, f, g)

Case	Length	Phase constraint $N(C_\phi)$	Max. temperature constraint $N(C_M)$
Zigzag	0.92	$3.76e-02$	$1.81e-03$
Contour	0.97	$1.48e-03$	$5.22e-03$
Spiral	0.97	$3.32e-04$	$4.14e-04$
Straight	0.95	$4.90e-03$	$4.21e-03$

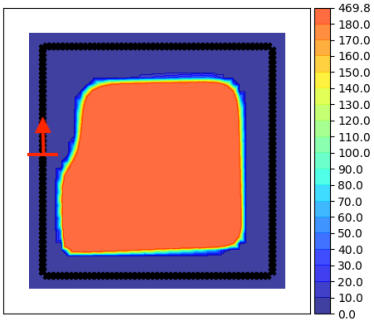
Table 5: Results in the unsteady case

5.3 Manufacturing a cantilever shape

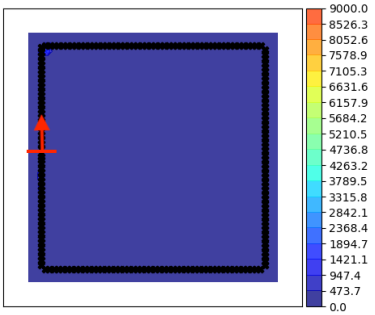
A final example consists in building two shapes. The first shape, a one hole cantilever, is presented by Figure 10 whereas the second one, a three holes cantilever, is given by Figure 11. In these cases, the maximum temperature is space dependent and the maximum temperature constraint C_M is split into two different ones such that $J(t_F, \alpha, \tilde{u}) = t_F + C_\phi + C_{M,\Sigma_s} + C_{M,\Sigma\setminus\Sigma_s}$. All the numerical values remain unchanged (see Section 5.1) but for the conductivity. Indeed, in order to ease the resolution, the conductivity out of the shape has been lowered to $50W \cdot m^{-1} \cdot K^{-1}$ whereas it remains $1000W \cdot m^{-1} \cdot K^{-1}$ in the shape. This is for instance the case when, after the building of some layers, the conductivity changes depending on the phase of the material which is under. The results correspond to iteration 1000.

In Figures 23 and 24, for the first and final iterations, the left plots represent regions of the working domain that have solidified (in blue)

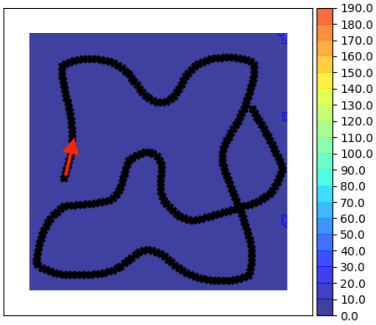
$$\left(T_\phi - N_\mathfrak{p}(t_F, T)\right)^+,$$



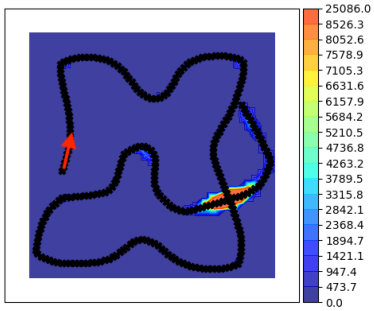
(a) First iteration, phase constraint, $L = 0.71m$, $C_\phi = 3131K^2m^2$



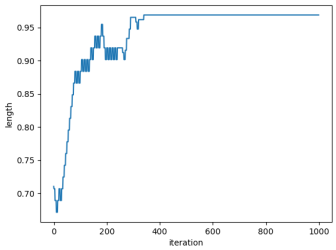
(b) First iteration, maximum constraint, $L = 0.71m$, $C_M = 8.74e-2K^2m^2s$



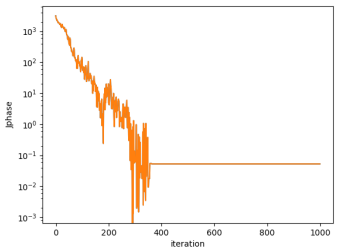
(c) Final iteration, phase constraint, $L = 0.97m$, $C_\phi = 5.27e-2K^2m^2$



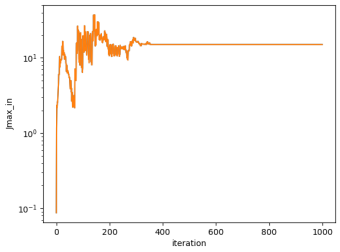
(d) Final iteration, maximum constraint, $L = 0.97m$, $C_M = 5.38e-2K^2m^2s$



(e) Evolution of length



(f) Evolution of $\ln(C_\phi)$



(g) Evolution of $\ln(C_M)$

Figure 20: Contour initialization: integrand of the phase (a, c) and maximum (b, d) constraints for the first and last iterations and convergence histories (e, f, g)

whereas the right plots give the maximum temperature constraint within the shape

$$\int_0^{t_F} (T - T_{M,\Sigma_s})^+ dt.$$

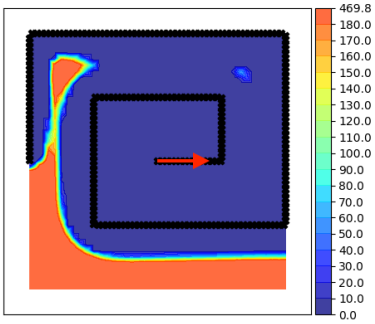
Table 6 presents the normalized numerical values. The results in the cantilever case corroborate the previous ones. Indeed, it is clear that the path is adapting to the constraints.

	Case	Length	$N(C_\phi)$	$N(C_{M,\Sigma_s})$	$N(C_{M,\Sigma\setminus\Sigma_s})$
One hole cantilever	First iteration	0.25	$4.82e-01$	$2.49e-03$	$1.96e-01$
	Last iteration	0.65	$2.70e-02$	$4.92e-03$	$5.90e-02$
Three holes cantilever	First iteration	0.24	$5.04e-01$	$1.15e-03$	$4.48e-01$
	Last iteration	0.78	$6.92e-03$	$3.36e-02$	$4.87e-01$

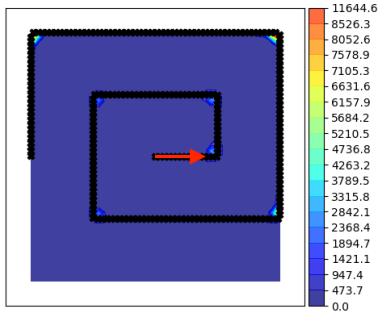
Table 6: Results for the cantilevers in the unsteady case

6 Conclusions and perspectives

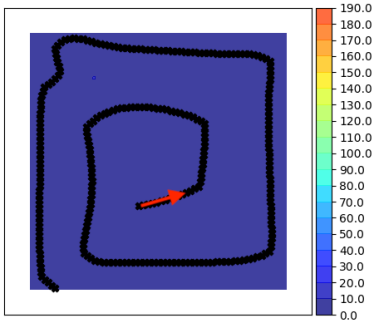
In contrast with the usual approach of pattern based scanning paths, we proposed in this paper a PDE-ODE optimal control method to optimize the trajectory. Before treating the realistic time-dependent problem, a toy model was introduced, based on a steady-state regime which yields a computationally cheap algorithm to test our optimization algorithm. In both cases (steady and unsteady), the numerical results show that large



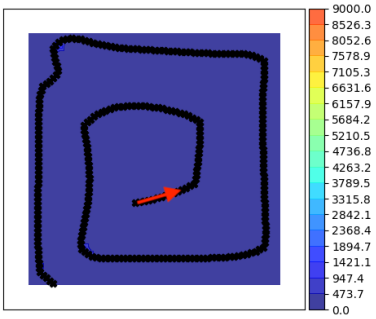
(a) First iteration, phase constraint, $L = 0.89m$, $C_\phi = 1052K^2m^2$



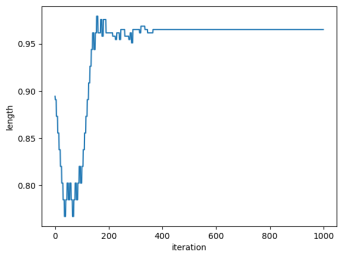
(b) First iteration, maximum constraint, $L = 0.89m$, $C_M = 1.56K^2m^2s$



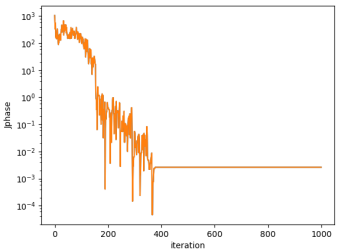
(c) Final iteration, phase constraint, $L = 0.97m$, $C_\phi = 2.63e-03K^2m^2$



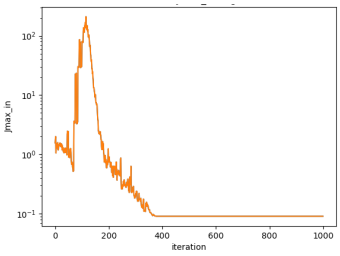
(d) Final iteration, maximum constraint, $L = 0.97m$, $C_M = 9.05e-2K^2m^2s$



(e) Evolution of length



(f) Evolution of $\ln(C_\phi)$

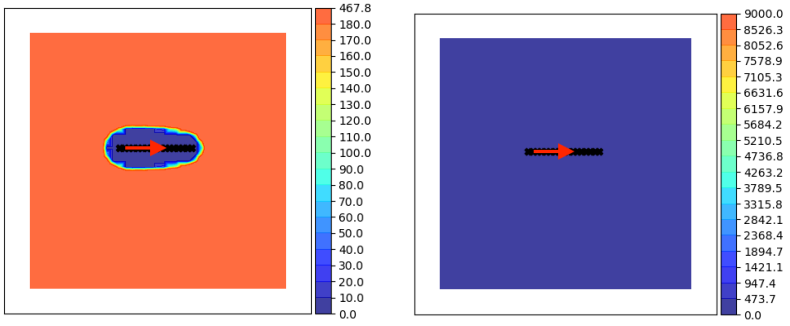


(g) Evolution of $\ln(C_M)$

Figure 21: Spiral initialization: integrand of the phase (a, c) and maximum (b, d) constraints for the first and last iterations and convergence histories (e, f, g)

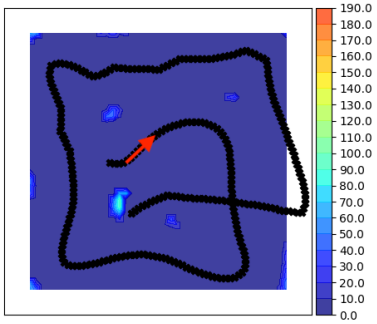
improvements are obtained by optimization. However, some issues appear. First of all, these problems are not convex and thus, many (local or global) minima exist. The optimized path depends on both the optimization parameters and the chosen initialization. In particular, according to the chosen initialization, the optimized path can be very different. A striking feature, unlike pattern based paths, is that all the proposed optimized paths have similar performances in terms of length and temperature constraints. An interesting perspective would be to determine a priori the required path length to then optimize the path keeping its length constant. Second, the optimized paths are strongly influenced by the numerical values of the physical parameters. Investigating this dependence in future works is key to the adaptation of the algorithm to more realistic problems.

Several other perspectives can be considered. Obviously, the physical model could be improved. In particular, the material characteristics should also depend on time, taking into account the advance of the phase change. On the other hand, more complex constraints could be considered (anisotropy for example [32]), and a mechanical state equation could be added to take into account the mechanical behavior of the built part. It would be very interesting to test this optimization algorithm on a sequence of successive layers to get some intuition on the evolution of the paths. A further generalisation is to add the velocity magnitude and the source power as optimization variables. In the present work, the modulus of the velocity is constant, which yields a proportional link between the path length and the final time. In particular, it implies that the final time is equal to the manufacturing time. In reality, the path length, the final time and the manufacturing time are all three different. A model's modification as well as the power and velocity optimization could lead to more realistic results.

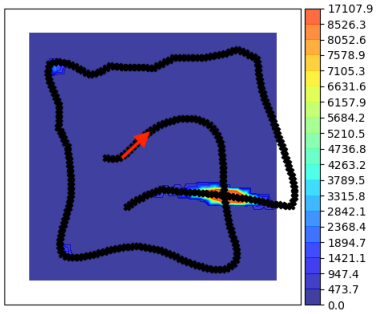


(a) First iteration, phase constraint, $L = 0.060m$, $C_\phi = 8097K^2m^2$

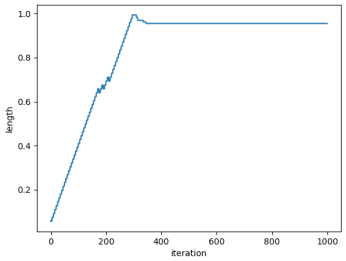
(b) First iteration, maximum constraint, $L = 0.060m$, $C_M = 0.0K^2m^2s$



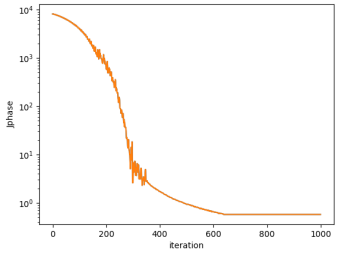
(c) Final iteration, phase constraint, $L = 0.95m$, $C_\phi = 5.75e-01K^2m^2$



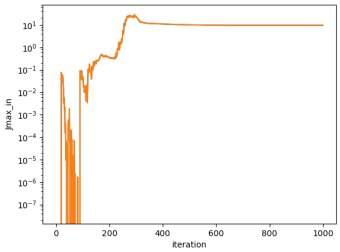
(d) Final iteration, maximum constraint, $L = 0.95m$, $C_M = 9.58K^2m^2s$



(e) Evolution of length



(f) Evolution of $\ln(C_\phi)$



(g) Evolution of $\ln(C_M)$

Figure 22: Straight line initialization: integrand of the phase (a, c) and maximum (b, d) constraints for the first and last iterations and convergence histories (e, f, g)

7 Acknowledgments

This work is supported by the SOFIA project, funded by Bpifrance (Banque Publique d'Investissement). We thank Forian Feppon for providing some of his routines.

8 Replication of results

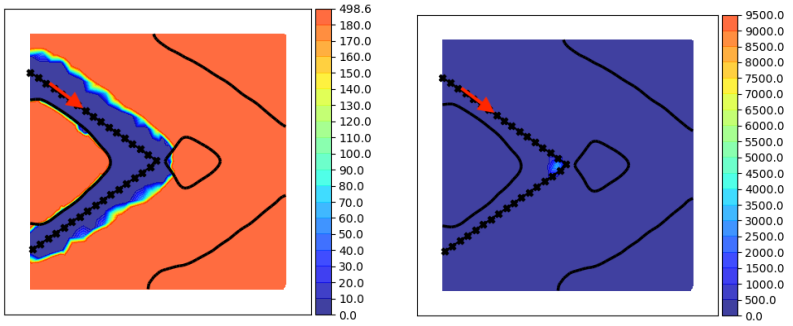
All algorithmic details are given in the paper. If any further information is required, readers can directly ask the authors.

9 Conflicts of interest

The authors declare no conflict of interest.

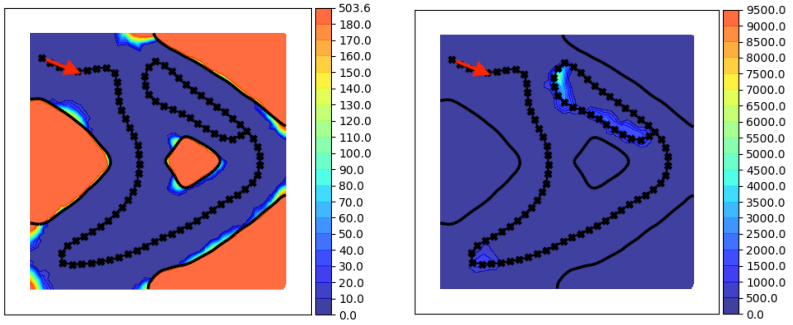
10 Appendix

In this Appendix some details are given about the derivation of the thermal model of Subsection 2.1. Let H be the height of the already built part of the object. The layer being processed corresponds to the coordinate $z = 0$ whereas the base plate is at $z = -H$ (each new layer corresponds to $z = 0$ whereas H increases during the process). The working box $D = \Sigma \times (-H, 0)$ corresponds to the already processed layers (with solid and



(a) First iteration, solid regions (blue)

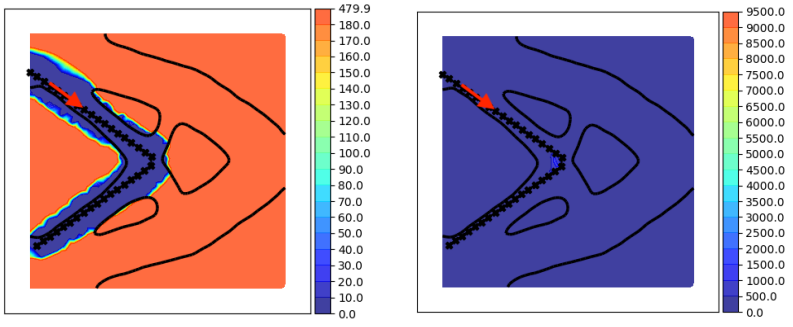
(b) First iteration, maximum constraint



(c) Final iteration, solid regions (blue)

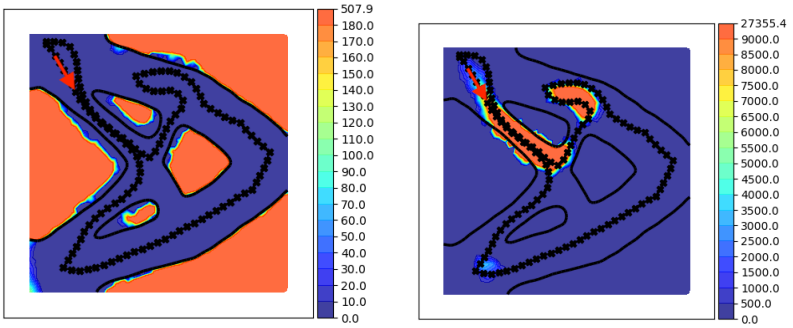
(d) Final iteration, maximum constraint

Figure 23: Results of path optimization for the one hole cantilever



(a) First iteration, solid regions (blue)

(b) First iteration, maximum constraint



(c) Final iteration, solid regions (blue)

(d) Final iteration, maximum constraint

Figure 24: Results of path optimization for the three holes cantilever

powder) and the new layer. We take into account the conduction and the heat source only, and, since the powder's conductivity is low, we assume adiabatic boundary conditions (Neumann). The three dimensional heat equation, based on the scheme presented by Figure 25 is stated by (40) (with ρ the density, c the heat capacity, λ the conductivity):

$$\begin{cases} \rho(t,x)c(t,x)\partial_t T(t,x) - \nabla\cdot(\lambda(t,x)\nabla T(t,x)) = 0, & (t,x)\in(0,t_F)\times D, \\ \lambda(t,x)\partial_n T(t,x) = q(t,x), & (t,x)\in(0,t_F)\times \partial D_{top}, \\ \lambda(t,x)\partial_n T(t,x) = 0, & (t,x)\in(0,t_F)\times \partial D_{side}, \\ T(t,x) = T_{ini}, & (t,x)\in(0,t_F)\times \partial D_{bottom}, \\ T(0,x) = T_{ini}, & x\in D, \end{cases} \tag{40}$$

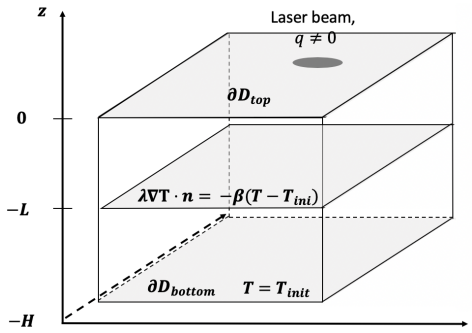


Figure 25: Three dimensional model

A first step consists in truncating the working domain to focus on the last layer. We consider a new working domain $\tilde{D}$ with thickness L ($0 < L < H$ and $\tilde{D} = \Sigma \times (-L, 0)$). We assume that, in this new working domain $\tilde{D}$, 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 β is set on the bottom boundary such that, $\forall (t,x) \in (0,t_F) \times \partial\tilde{D}_{bottom}$,

$$\lambda \partial_n T = -\beta \left(T(t,x) - T_{ini} \right).$$

This models the heat loss by conduction from the domain $\tilde{D}$ to the domain $D \setminus \tilde{D}$. The coefficient β measures this heat transmission and must be related to the conduction at the surface and to a characteristic thickness ΔZ. We choose $\beta = \frac{\lambda}{\Delta Z}$. This leads to a modified heat equation:

$$\begin{cases} \rho(t,x)c(t,x)\partial_t T(t,x) - \nabla \cdot (\lambda(t,x)\nabla T(t,x)) = 0, & (t,x) \in (0,t_F) \times \tilde{D}, \\ \lambda(t,x)\partial_n T(t,x) = q(t,x), & (t,x) \in (0,t_F) \times \partial\tilde{D}_{top}, \\ \lambda(t,x)\partial_n T(t,x) = 0, & (t,x) \in (0,t_F) \times \partial\tilde{D}_{side}, \\ \lambda\partial_n T(t,x) = -\beta \left(T(t,x) - T_{ini} \right), & (t,x) \in (0,t_F) \times \partial\tilde{D}_{bottom}, \\ T(0,x) = T_{ini}, & x \in \tilde{D}, \end{cases} \tag{41}$$

In a second step, we average (41) in the vertical direction in order to deduce a two dimensional model. Whereas we had $x = (X,Y,Z)$ and ∇ the three gradients operator in (40), 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 Σ. We set $\Delta' = \partial_X^2 + \partial_Y^2$ such that $\Delta T = \Delta' T + \partial_Z^2 T$. An integration along the (Oz)-axis, between $(-L)$ and 0 gives:

$$\begin{aligned} \int_{-L}^{0} \rho c \partial_t T - \nabla' \cdot \left(\lambda \nabla' T \right) dZ &= \rho c \partial_t \left(\int_{-L}^{0} T dz \right) - \nabla \cdot \left(\lambda \nabla \left(\int_{-L}^{0} T dz \right) \right) + \int_{-L}^{0} \lambda \partial_Z^2 T dZ \\ &= \rho c L \partial_t \tilde{T} - L \nabla' \cdot \left(\lambda \nabla' \tilde{T} \right) + \lambda \left[\partial_Z T \right]_{-L}^{0}, \\ &= \rho c L \partial_t \tilde{T} - L \nabla' \cdot \left(\lambda \nabla' \tilde{T} \right) - q + \beta \left(T(-L) - T_{ini} \right). \end{aligned} \tag{42}$$

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

11 References

References

[1] Tonia-Maria Alam, Serge Nicaise, and Luc Paquet. "An optimal control problem governed by heat equation with nonconvex constraints applied to selective laser melting process". In: *Hal preprint: https://hal.archives-ouvertes.fr/hal-02302403* (2019).

[2] G. Allaire et al. "Multi-phase structural optimization *via* a level set method". In: *ESAIM Control Optim. Calc. Var.* 20.2 (2014), pp. 576–611.

[3] Grégoire Allaire. *Conception Optimale de Structures*. Vol. 58. Mathématiques & Applications (Berlin) [Mathematics & Applications]. Springer-Verlag, Berlin, 2007.

[4] Grégoire Allaire and Beniamin Bogosel. "Optimizing supports for additive manufacturing". In: *Structural and Multidisciplinary Optimization* 58.6 (2018), pp. 2493–2515.

[5] Grégoire Allaire and Lukas Jakabčin. "Taking into Account Thermal Residual Stresses in Topology Optimization of Structures Built by Additive Manufacturing". In: *Mathematical Models and Methods in Applied Sciences* 28.12 (2018), pp. 2313–2366.

[6] Grégoire Allaire, François Jouve, and Anca-Maria Toader. “Structural Optimization Using Sensitivity Analysis and a Level-Set Method”. In: *Journal of Computational Physics* 194.1 (2004), pp. 363–393.

[7] Grégoire Allaire, François Jouve, and Nicolas Van Goethem. “Damage and fracture evolution in brittle materials by shape optimization methods”. In: *J. Comput. Phys.* 230.12 (2011), pp. 5010–5044.

[8] E. M. Arkin, M. Held, and C. L. Smith. “Optimization Problems Related to Zigzag Pocket Machining”. In: *Algorithmica* 26.2 (Feb. 2000), pp. 197–236.

[9] Claude Barlier and Alain Bernard. *Fabrication Additive - Du Prototypage Rapide à l’impression 3D.* Dunod, 2016.

[10] Alexander I Bobenko. *Geometry II: Discrete Differential Geometry*.

[11] Mathilde Boissier. “Coupling topology optimization methods and trajectory optimization methods in additive manufacturing”. PhD thesis. 2020 (to appear).

[12] Michel Bouard, Vincent Pateloup, and Paul Armand. “Pocketing Toolpath Computation Using an Optimization Method”. In: *Computer-Aided Design* 43.9 (Sept. 2011), pp. 1099–1109.

[13] Martin Burger. “A Framework for the Construction of Level Set Methods for Shape Optimization and Reconstruction”. In: *Interfaces and Free Boundaries* 5.3 (2002), pp. 301–329.

[14] Luke N. Carter, Moataz M. Attallah, and Roger C. Reed. “Laser Powder Bed Fabrication of Nickel-Base Superalloys: Influence of Parameters; Characterisation, Quantification and Mitigation of Cracking”. In: *Superalloys* 2012 (2012), pp. 577–586.

[15] Jean Céa. “Conception Optimale Ou Identification de Formes: Calcul Rapide de La Dérivée Directionnelle de La Fonction Coût”. In: *RAIRO Modélisation Mathématique et Analyse Numérique* 20.3 (1986), pp. 371–402.

[16] Qiang Chen et al. “Three-Dimensional Finite Element Thermomechanical Modeling of Additive Manufacturing by Selective Laser Melting for Ceramic Materials”. In: *Additive Manufacturing* 16 (2017), pp. 124–137.

[17] Kurt Chudej et al. “Instationary Heat-Constrained Trajectory Optimization of a Hypersonic Space Vehicle by ODE-PDE-Constrained Optimal Control”. In: *Variational Analysis and Aerospace Engineering*. Vol. 33. Springer, New York, 2009, pp. 127–144.

[18] K. Dai and L. Shaw. “Distortion Minimization of Laser-processed Components through Control of Laser Scanning Patterns”. In: *Rapid Prototyping Journal* 8.5 (Dec. 2002), pp. 270–276.

[19] T. DebRoy et al. “Additive Manufacturing of Metallic Components Process, Structure and Properties”. In: *Progress in Materials Science* 92 (Mar. 2018), pp. 112–224.

[20] Donghong Ding et al. “A practical path planning methodology for wire and arc additive manufacturing of thinwalled structures”. In: *Robotic and Computer-Integrated Manufacturing* 34 (2015), pp. 8–19.

[21] Kamel Ettaieb, Sylvain Lavernhe, and Christophe Tournier. “Thermal simulation of scanning paths in LBM additive manufacturing”. In: *Second International Conference on Simulation for Additive Manufacturing, Sim-AM 2019, Pavia (Italy)* (2019).

[22] Wei Gao et al. “The Status, Challenges, and Future of Additive Manufacturing in Engineering”. In: *Computer-Aided Design* 69 (2015), pp. 65–89.

[23] Matthias Gerdts. *Optimal Control of ODEs and DAEs*. De Gruyter, 2011.

[24] F. Hecht. “New Development in Freefem++”. In: *Journal of Numerical Mathematics* 20.3-4 (Jan. 2012), pp. 251–265.

[25] Antoine Henrot and Michel Pierre. *Shape Variation and Optimization*. Vol. 28. EMS Tracts in Mathematics. European Mathematical Society (EMS), Zürich, 2018.

[26] Roland Herzog and Karl Kunisch. “Algorithms for PDE-Constrained Optimization”. In: *GAMM-Mitteilungen* 33.2 (2010), pp. 163–176.

[27] Jamasp Jhabvala et al. “On the Effect of Scanning Strategies in the Selective Laser Melting Process”. In: *Virtual and Physical Prototyping* 5.2 (June 2010), pp. 99–109.

[28] Yu-an Jin et al. “Optimization of Tool-Path Generation for Material Extrusion-Based Additive Manufacturing Technology”. In: *Additive Manufacturing*. Inaugural Issue 1 (Oct. 2014), pp. 32–47.

[29] 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”. In: *Mathematical and Computer Modelling of Dynamical Systems. Methods, Tools and Applications in Engineering and Related Sciences* 24.2 (2018), pp. 182–206.

[30] Matthijs Langelaar. “Combined Optimization of Part Topology, Support Structure Layout and Build Orientation for Additive Manufacturing”. In: *Structural and Multidisciplinary Optimization* 57.5 (2018), pp. 1985–2004.

[31] J.-L. Lions. *Contrôle Optimal de Systèmes Gouvernés Par Des Équations Aux Dérivées Partielles*. Dunod, Paris; Gauthier-Villars, Paris, 1968.

[32] Jikai Liu and Albert C. To. “Deposition Path Planning-Integrated Structural Topology Optimization for 3D Additive Manufacturing Subject to Self-Support Constraint”. In: *Computer-Aided Design* 91 (2017), pp. 27–45.

[33] Jikai Liu et al. “Current and Future Trends in Topology Optimization for Additive Manufacturing”. In: *Structural and Multidisciplinary Optimization* 57.6 (June 2018), pp. 2457–2483.

[34] E. Liverani et al. “Effect of Selective Laser Melting (SLM) Process Parameters on Microstructure and Mechanical Properties of 316L Austenitic Stainless Steel”. In: *Journal of Materials Processing Technology* 249 (2017), pp. 255–263.

[35] Liang Ma and Hongzan Bin. “Temperature and Stress Analysis and Simulation in Fractal Scanning-Based Laser Sintering”. In: *The International Journal of Advanced Manufacturing Technology* 34.9-10 (Oct. 2007), pp. 898–903.

[36] Emmanuel Maitre. “Review of numerical methods for free interfaces”. In: *Les Houches* 27 (2006), p. 31.

[37] Mustafa Megahed et al. “Metal Additive-Manufacturing Process and Residual Stress Modeling”. In: *Integrating Materials and Manufacturing Innovation* 5.1 (Dec. 2016), pp. 61–93.

[38] Peter Mercelis and Jean-Pierre Kruth. “Residual Stresses in Selective Laser Sintering and Selective Laser Melting”. In: *Rapid prototyping journal* 12.5 (2006), pp. 254–265.

[39] Amir M. Mirzendehdel and Krishnan Suresh. “Support Structure Constrained Topology Optimization for Additive Manufacturing”. In: *Computer-Aided Design* 81 (2016), pp. 1–13.

[40] Sankhya Mohanty and Jesper Hattel. “Cellular Scanning Strategy for Selective Laser Melting: Capturing Thermal Trends with a Low-Fidelity, Pseudo-Analytical Model”. In: *Mathematical Problems in Engineering* 2014 (2014).

[41] François Murat and Jacques Simon. “Etude de Problemes d’optimal Design”. In: *Optimization Techniques Modeling and Optimization in the Service of Man Part 2*. Ed. by G. Goos et al. Vol. 41. Berlin, Heidelberg: Springer Berlin Heidelberg, 1976, pp. 54–62.

[42] Jorge Nocedal and Stephen J. Wright. *Numerical Optimization*. Second. Springer Series in Operations Research and Financial Engineering. Springer, New York, 2006.

[43] Olivier Pantz. “Sensibilité de l’équation de la chaleur aux sauts de conductivité”. In: *C. R. Math. Acad. Sci. Paris* 341.5 (2005), pp. 333–337.

[44] X. Peng, K. Niakhai, and B. Protas. “A method for geometry optimization in a simple model of two-dimensional heat transfer”. In: *SIAM J. Sci. Comput.* 35.5 (2013), B1105–B1131.

[45] Pascal Romon. *Introduction à La Géométrie Différentielle Discrète*. Références Sciences. 2013.

[46] Frédéric Royer. “Fonctionnement et singularités du procédé de fusion laser sélective : Illustration par application à deux superalliages à base nickel et considérations énergétiques”. PhD thesis. Ecole Nationale Supérieure des Mines de Paris, 2014.

[47] Babis Schoinochoritis, Dimitrios Chantzis, and Konstantinos Salonitis. “Simulation of Metallic Powder Bed Additive Manufacturing Processes with the Finite Element Method: A Critical Review”. In: *Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture* 231.1 (Jan. 2017), pp. 96–117.

[48] Emmanuel Stathatos and George-Christopher Vosniakos. “Real-Time Simulation for Long Paths in Laser-Based Additive Manufacturing: A Machine Learning Approach”. In: *The International Journal of Advanced Manufacturing Technology* (2019).

[49] Jun Hao Tan, Wai Leong Eugene Wong, and Kenneth William Dalgarno. “An Overview of Powder Granulometry on Feedstock and Part Performance in the Selective Laser Melting Process”. In: *Additive Manufacturing* 18 (2017), pp. 228–255.

[50] Mary Kathryn Thompson et al. “Design for Additive Manufacturing: Trends, Opportunities, Considerations, and Constraints”. In: *CIRP Annals* 65.2 (2016), pp. 737–760.

[51] Emmanuel Trélat. *Contrôle Optimal*. Vuibert, Paris, 2008.

[52] G. Tryggvason et al. “A Front-Tracking Method for the Computations of Multiphase Flow”. In: *Journal of Computational Physics* 169.2 (2001), pp. 708–759.

[53] Laurent Van Belle, Guillaume Vansteenkiste, and Jean Claude Boyer. “Investigation of residual stresses induced during the selective laser melting process”. In: 554 (2013), pp. 1828–1834.

[54] A. Vandone, S. Baraldo, and A. Valente. “Multisensor Data Fusion for Additive Manufacturing Process Control”. In: *IEEE Robotics and Automation Letters* 3.4 (2018), pp. 3279–3284.

[55] S. Wendl, H. J. Pesch, and A. Rund. “On a State-Constrained PDE Optimal Control Problem Arising from ODE-PDE Optimal Control”. In: (2010). Ed. by Moritz Diehl et al., pp. 429–438.

[56] Stefan Wendl. “On a Prototype of an Optimal Control Problem Governed by Ordinary and Partial Differential Equations”. PhD thesis. 2014.

[57] Kaufui V. Wong and Aldo Hernandez. “A Review of Additive Manufacturing”. In: *ISRN Mechanical Engineering* (2012).

[58] Amanda S. Wu et al. “An Experimental Investigation into Additive Manufacturing-Induced Residual Stresses in 316L Stainless Steel”. In: *Metallurgical and Materials Transactions A* 45.13 (2014), pp. 6260–6270.