

## THÈSE

Pour obtenir le grade de

### DOCTEUR DE L'UNIVERSITÉ GRENOBLE ALPES

École doctorale : MSTII - Mathématiques, Sciences et technologies de l'information, Informatique

Spécialité : Mathématiques Appliquées

Unité de recherche : Laboratoire Jean Kuntzmann

## Optimisation de formes sur des surfaces

## Shape and topology optimization on surfaces

Présentée par :

**Carlos BRITO PACHECO**

### Direction de thèse :

**Charles DAPOGNY**

CHARGE DE RECHERCHE HDR, CNRS DELEGATION ALPES

Directeur de thèse

**Eric BONNETIER**

PROFESSEUR DES UNIVERSITES, UNIVERSITE GRENOBLE ALPES

Co-directeur de thèse

**Rafael ESTEVEZ**

PROFESSEUR DES UNIVERSITES, UNIVERSITE GRENOBLE ALPES

Co-directeur de thèse

### Rapporteurs :

**ULISSE STEFANELLI**

FULL PROFESSOR, UNIVERSITÄT WIEN

**MARC BONNET**

DIRECTEUR DE RECHERCHE, CNRS DELEGATION ILE-DE-FRANCE SUD

### Thèse soutenue publiquement le **12 décembre 2024**, devant le jury composé de :

**EDOUARD OUDET,**

PROFESSEUR DES UNIVERSITES, UNIVERSITE GRENOBLE ALPES

Président

**CHARLES DAPOGNY,**

CHARGE DE RECHERCHE HDR, CNRS DELEGATION ALPES

Directeur de thèse

**ERIC BONNETIER,**

PROFESSEUR DES UNIVERSITES, UNIVERSITE GRENOBLE ALPES

Co-directeur de thèse

**ULISSE STEFANELLI,**

FULL PROFESSOR, UNIVERSITÄT WIEN

Rapporteur

**MARC BONNET,**

DIRECTEUR DE RECHERCHE, CNRS DELEGATION ILE-DE-FRANCE SUD

Rapporteur

**FABIEN CAUBET,**

MAITRE DE CONFERENCES HDR, UNIVERSITE DE PAU ET PAYS DE L'ADOUR

Examineur

**FIORALBA CAKONI,**

FULL PROFESSOR, RUTGERS, THE STATE UNIVERSITY NEW JERSEY

Examinatrice

### Invités :

**RAFAEL ESTEVEZ**

PROFESSEUR DES UNIVERSITES, UNIVERSITE GRENOBLE ALPES



This page was intentionally left blank.

# Shape and topology optimization on surfaces

Carlos BRITO PACHECO  
carlos.brito-pacheco@univ-grenoble-alpes.fr

October 6, 2024

# Contents

---

<b>Acknowledgments</b>	<b>12</b>
<b>Résumé de la thèse</b>	<b>14</b>
<b>Introduction</b>	<b>26</b>
<b>1 A review of shape and topology optimization</b>	<b>31</b>
1.1 Historical context . . . . .	31
1.2 Hadamard’s boundary variation method . . . . .	32
1.2.1 Definitions and generalities . . . . .	33
1.2.2 Techniques and approaches to compute shape derivatives . . . . .	34
1.2.3 Shape derivatives of criteria depending on the solution to a boundary value problem . . . . .	36
1.3 The Hilbertian extension-regularization procedure . . . . .	39
1.4 Topological derivatives . . . . .	41
1.5 Body-fitted shape optimization . . . . .	43
1.5.1 Shape representation via triangulations . . . . .	43
1.5.2 The level set method for tracking the motion of a domain . . . . .	45
1.5.3 Numerical discretization of the advection equations . . . . .	46
1.5.4 The signed distance function . . . . .	47
1.5.5 Discretization of the negative sub-domain of the level-set function . . . . .	48
1.6 The classical cantilever example . . . . .	48
1.6.1 The optimization problem . . . . .	48
1.6.2 A numerical experiment . . . . .	50
1.7 Limitations, variants and the homogenization phenomenon . . . . .	53
1.8 State of the art for shape and topology optimization of surfaces . . . . .	54
1.8.1 Theoretical developments and generalizations to manifolds . . . . .	55
1.8.2 Numerical evolution of domains on hypersurfaces embedded in $\mathbb{R}^d$ . . . . .	55
1.8.3 Optimization of the regions supporting boundary conditions for different PDEs . . . . .	56
1.8.4 Open-source programming tools for developing shape and topology optimization algorithms . . . . .	57
<b>2 Towards a framework for the optimization of domains on manifolds</b>	<b>59</b>
2.1 Shape differentiability on manifolds . . . . .	60
2.1.1 The deformation of space . . . . .	60
2.1.2 Relation to parallel transport . . . . .	62
2.2 Shape representation via the signed distance function . . . . .	66
2.3 Generalities of shape derivatives on manifolds . . . . .	69
2.3.1 Structure of shape derivatives for a regular domain in the ambient space . . . . .	69
2.3.2 Structure of shape derivatives for a regular domain in a hypersurface of the ambient manifold . . . . .	71
2.3.3 Computation of shape derivatives . . . . .	72
2.3.4 Shape functionals constrained by a partial differential equation . . . . .	73

2.4	Future work . . . . .	75
<b>3</b>	<b>Body-fitted tracking within a surface via a level set based mesh evolution method</b>	<b>76</b>
3.1	Evolution of a region on a surface in the level set framework . . . . .	78
3.2	Presentation of the numerical strategy . . . . .	80
3.2.1	Outline of the numerical algorithm . . . . .	80
3.2.2	Computation of the signed distance function on a surface . . . . .	81
3.2.3	Resolution of the level set advection equation on the surface $S$ . . . . .	83
3.2.4	Meshing of the negative subdomain of a level set function . . . . .	84
3.3	Numerical framework . . . . .	87
3.4	Motion in the direction of the conormal vector field . . . . .	87
3.5	Wildland fire propagation . . . . .	90
<b>4</b>	<b>Shape and topology optimization of the regions supporting boundary conditions</b>	<b>96</b>
4.1	Mathematical framework . . . . .	98
4.2	Computation of shape derivatives with respect to variations in the support of boundary conditions . . . . .	98
4.2.1	Hadamard's boundary variation method on a domain contained within a surface . . . . .	98
4.2.2	Smoothed interfaces for weakly-singular problems . . . . .	100
4.2.3	Approximate formulas for the numerical implementation of the regularized shape derivative . . . . .	102
4.2.4	A model optimization problem of the regions supporting boundary conditions of a thermal mechanics problem . . . . .	104
4.3	Sensitivity with respect to topological variations in the support of boundary conditions . . . . .	108
4.3.1	The topological derivative of replacing small regions of a set with another set . . . . .	108
4.3.2	Topological derivative of a criterion under the constraint of a boundary value problem . . . . .	109
4.4	Sobolev spaces on the boundary of a domain . . . . .	111
4.5	The single layer potential operator . . . . .	111
4.6	The case of the conductivity equation . . . . .	112
4.6.1	Replacement of homogeneous Neumann boundary conditions by homogeneous Dirichlet conditions . . . . .	113
4.6.2	Replacement of homogeneous Neumann boundary conditions by inhomogeneous Neumann boundary conditions . . . . .	117
4.7	The case of the Helmholtz equation . . . . .	118
4.7.1	Presentation of the Helmholtz model . . . . .	118
4.7.2	Replacement of homogeneous Neumann boundary conditions by Robin boundary conditions . . . . .	119
4.8	The case of the linear elasticity system . . . . .	122
4.8.1	Presentation of the linear elasticity setting . . . . .	122
4.8.2	Replacement of homogeneous Neumann boundary conditions by homogeneous Dirichlet conditions . . . . .	124
4.9	Numerical resolution of singular boundary integral equations . . . . .	127
4.9.1	The Galerkin scheme . . . . .	128
4.9.2	Decomposition of singular integrals through relative coordinates . . . . .	128
4.9.3	Regularization of the variational problem . . . . .	131
4.9.4	Numerical validation . . . . .	131
4.9.5	Extension to the vector-valued case . . . . .	134
4.10	Numerical method . . . . .	141
4.11	Optimization of the repartition of cathode-anode regions for a direct current electroosmotic mixer . . . . .	141
4.11.1	The optimization problem . . . . .	142
4.11.2	The shape derivative . . . . .	143
4.11.3	The topological derivative . . . . .	144
4.11.4	Experiment setup . . . . .	145
4.11.5	Analysis of results . . . . .	145
4.12	Optimization of sound-soft regions on the surface of a sound-hard obstacle for acoustic cloaking . . . . .	152
4.12.1	The scattering problem . . . . .	152

4.12.2	The optimization problem . . . . .	155
4.12.3	The shape derivative . . . . .	155
4.12.4	The topological derivative . . . . .	155
4.12.5	Experiment setup . . . . .	156
4.12.6	Analysis of results . . . . .	156
4.13	Optimization of the repartition of structural supports for a water tank . . . . .	168
4.13.1	The optimization problem . . . . .	169
4.13.2	The shape derivative . . . . .	169
4.13.3	The topological derivative . . . . .	170
4.13.4	Experiment setup . . . . .	170
4.13.5	Analysis of results . . . . .	170
4.14	Optimization of a clamp-locator system . . . . .	178
4.14.1	The optimization problem . . . . .	178
4.14.2	The shape derivative . . . . .	179
4.14.3	The topological derivative . . . . .	180
4.14.4	Experiment setup . . . . .	180
4.14.5	Analysis of results . . . . .	181
<b>5</b>	<b>Rodin: A numerical C++20 library for shape and topology optimization</b>	<b>188</b>
5.1	Introduction . . . . .	188
5.1.1	Philosophy . . . . .	188
5.1.2	Similar libraries to Rodin . . . . .	189
5.2	Modules . . . . .	190
5.3	Theoretical basis . . . . .	191
5.3.1	A primer on meshes . . . . .	191
5.3.2	A primer on finite elements . . . . .	193
5.3.3	From weak formulations to linear algebra . . . . .	195
5.4	DSL embedding via template metaprogramming . . . . .	196
5.5	Code examples and illustrations . . . . .	196
5.5.1	The Poisson equation . . . . .	196
5.5.2	The Helmholtz equation . . . . .	198
5.5.3	The linear elasticity equation . . . . .	201
5.5.4	The level-set cantilever optimization in 2D . . . . .	203
	<b>Conclusions and perspectives</b>	<b>212</b>
<b>A</b>	<b>Supporting material</b>	<b>231</b>
A.1	Useful identities for the computation of shape derivatives . . . . .	231
A.2	Proofs . . . . .	231
A.2.1	Proof of Proposition 4.4 . . . . .	231
A.2.2	Proof of Proposition 4.6 . . . . .	235
<b>B</b>	<b>Differentiability</b>	<b>237</b>
B.1	The Fréchet derivative . . . . .	237
B.2	Differentials and gradients . . . . .	237
B.3	Jacobi's formula . . . . .	238
B.4	Differentiation through a minimum . . . . .	238
B.5	Implicit function theorem . . . . .	239
B.6	The space $W^{1,\infty}$ . . . . .	239
<b>C</b>	<b>Useful integration formulas</b>	<b>240</b>
C.1	Green's identities . . . . .	240
C.2	Tangential calculus . . . . .	240

<b>D</b>	<b>Riemannian geometry</b>	<b>242</b>
D.0.1	Riemannian manifolds . . . . .	242
D.0.2	Connections and covariant derivatives . . . . .	245
D.0.3	Geodesics, the exponential map, and tubular neighborhoods . . . . .	246
D.0.4	Curvature, families of curves, and Jacobi fields . . . . .	248
D.0.5	Distance functions . . . . .	250
<b>E</b>	<b>Fréchet spaces, the Gateaux derivative, and continuous differentiability</b>	<b>251</b>

# List of Figures

---

1	Examples of optimal designs obtained via different shape and topology optimization techniques. . . . .	28
1.2	Domain $\Omega$ undergoing boundary variations characterized by the deformation vector field $\theta$ . The perturbed domain $\Omega_\theta$ shows smooth modifications of the boundary. . . . .	33
1.3	Domain $\Omega$ with a small hole nucleated at point $x$ , represented by the ball $B(x, \varepsilon)$ . This illustrates a topological perturbation used to compute the topological derivative by examining the effect of the localized change. . . . .	41
1.4	Different representations of a mechanical piece $\Omega$ , where one can solve a partial differential equation using the finite element method once the domain has been discretized into a mesh. . . . .	44
1.5	The shape $\Omega$ is implicitly represented by a function $\phi(x)$ , where $\phi < 0$ inside the domain $\Omega$ , $\phi = 0$ on the boundary $\partial\Omega$ , and $\phi > 0$ outside. . . . .	45
1.6	The image shows a typical schematic from shape optimization of a cantilever beam. The beam is fixed on the left side $\Gamma_D$ and subjected to a vertical force on the right side $\Gamma_N$ . The optimized structure $\Omega$ minimizes compliance by efficiently distributing material, leaving voids to reduce weight while maintaining structural stiffness. The boundary $\Gamma$ is the “traction-free” part of $\partial\Omega$ , and the design region $D$ is highlighted in yellow, representing the area where the shape optimization occurs. . . . .	49
1.7	Initial design for the cantilever. . . . .	50
1.8	Snapshots of the shape optimization process described in Algorithm 1. . . . .	51
1.8	Snapshots of the shape optimization process described in Algorithm 1. . . . .	52
1.9	Two different local minimizers, with different topologies. Illustrations taken from §6.5.3 in [22]. . . . .	53
1.10	The image shows a branching structure (black regions) representing the conductive material, embedded in a lighter background (voids), characteristic of a homogenization phenomenon in shape optimization. The design features fine-scale microstructures that spread out like tree branches, aiming to minimize material usage while maintaining performance. This illustrates the emergence of complex geometries that are challenging to realize in practical applications, yet represent an optimal solution in the homogenized sense. . . . .	54
2.1	Illustration of the deformation of a regular domain $G$ in a possibly curved manifold $M$ . The deformation is governed by the map $F_\theta^M$ , which smoothly transforms points in $G$ to a deformed domain, tracking the path $\gamma_{\theta(p)}(t)$ for each point $p$ . . . . .	61
2.2	Comparison of volume distortion along a geodesic between the parallel orthonormal frame $E(t)$ and the Jacobi field $J(t)$ , both identified with matrix paths, coinciding at $t = 0$ but potentially diverging at later times, illustrating the impact of curvature on volume distortion. . . . .	65
2.3	The shaded region represents a tubular neighborhood $U \subset M$ around the boundary $\partial G$ of a regular domain $G$ . The solid black line arrow marks the projection of a point $F_\theta^M(p)$ onto the boundary $\partial G$ and the solid blue line corresponds to $\partial G$ . . . . .	67
2.4	The shaded region shows a tubular neighborhood $U \subset M$ around the boundary $\partial G$ of a regular domain $G$ . The solid blue line denotes $\partial G$ . . . . .	70

3.1	Example of a region $G(t)$ of a three-dimensional surface $S$ evolving according to a tangential velocity field $V(t, x)$ . . . . .	78
3.2	(a) Level set function $\phi : S \rightarrow \mathbb{R}$ for a region $G \subset S$ defined at the vertices of the triangulation of a surface $S \subset \mathbb{R}^3$ ; (b) High-quality mesh $\mathcal{T}$ of $S$ , enclosing a submesh $\mathcal{T}_{\text{int}}$ of $G$ (in dark blue). . . . .	81
3.3	(a) Invalid two-dimensional triangulation, presenting overlapping triangles (in red); (b) Valid, yet non conforming mesh (the red node lies strictly inside an edge); (c) Valid, conforming, but ill-shaped mesh (the red triangles are nearly degenerate); (d) High-quality computational mesh. . . . .	82
3.4	(a) Values of a level set function $\phi : S \rightarrow \mathbb{R}$ for a region $G \subset S$ ; (b) Ill-shaped mesh $\mathcal{T}_{\text{temp}}$ obtained after explicit discretization of $G$ into $\mathcal{T}$ ; (c) High-quality mesh $\tilde{\mathcal{T}}$ obtained after remeshing $\mathcal{T}_{\text{temp}}$ . . . . .	85
3.5	Illustration of the four operations involved in the remeshing strategy of Section 3.2.4; (a) Split of the “long” edge $pq$ into two edges $pm$ and $mp$ , where the new point $m$ is introduced on $S$ ; (b) Collapse of the “short” edge $pq$ ; (c) Swap of the edge $pq$ between the triangles $pqr$ and $pqs$ for the alternate configuration, featuring the edge $rs$ and the triangles $rsp$ and $rsq$ ; (d) Relocation of the vertex $p$ which slides along $S$ . . . . .	86
3.6	Setting of the motion of a region $G(t)$ within the unit sphere $\mathbb{S}^2$ along the conormal vector field $n_{\Sigma(t)}$ considered in Section 3.4 . . . . .	88
3.7	Evolution $G(t)$ of the spherical cap $\mathcal{C}_a$ under unit conormal velocity considered in Section 3.4; (a) Initial configuration $G(0) = \mathcal{C}_a$ ; (b) Intermediate configuration $G(t)$ at time $t = \pi/4$ ; (c) Final region $G(T)$ corresponding to the upper half-sphere. . . . .	89
3.8	History of the least-square error $\mathcal{E}$ in (3.17) for different values of the mesh size $h$ and “CFL number” $c$ in the conormal advection example of Section 3.4. . . . .	89
3.9	Illustration of the evolution of a fire on a topography. The slope vector $p$ indicates the direction of greatest ascent as visualized by the arrows pointing towards the peaks. The burning region $G$ is indicated by the yellow-red zone whose motion is prescribed by the arrows representing $V$ . . . . .	91
3.10	Illustration of the geometric relations involved in the computation of the rate of spread $R$ at the level of one triangle $T \in \mathcal{T}$ , in the fire propagation model of Section 3.5. . . . .	91
3.11	Landscape surface $S$ used in the first experiment of Section 3.5; the initially burnt region $G(0)$ , visualized in (b) and (d), is represented in red, and the color scale in (a) and (c) accounts for the height function $s$ . The lower left corner of $S$ is set to $(0, 0, 0) \in \mathbb{R}^3$ . . . . .	92
3.12	Snapshots of the first experiment of propagation of a fire front conducted in Section 3.5. . . . .	93
3.13	Snapshots of the second experiment of propagation of a fire front conducted in Section 3.5. . . . .	94
3.14	Snapshots of the third experiment of propagation of a fire front conducted in Section 3.5. . . . .	95
4.1	Optimization of the region $G$ of the boundary $\partial\Omega$ of a shape by the method of Hadamard. . . . .	99
4.2	Illustration of the mechanical part $\Omega$ where boundary conditions are optimized in Section 4.2.4. . . . .	105
4.3	Boundary optimization process of the mechanical device considered in Section 4.2.4 with penalization parameter $\ell = 0.01$ and mesh size $h_{\text{max}} = 0.08$ . The optimized region $G$ bearing homogeneous Dirichlet boundary conditions is represented in blue. . . . .	106
4.4	Optimized designs and convergence history. . . . .	107
4.5	<i>Variation <math>G_{x_0, \varepsilon}</math> in (4.18) of a surface region <math>G</math> obtained by addition of a “small” surfacic disk <math>\omega_{x_0, \varepsilon}</math>.</i> . . . . .	108
4.6	<i>The boundary <math>\partial\Omega</math> is assumed to be flat in a neighborhood of the considered point 0.</i> . . . .	109
4.7	Illustration of the unit triangle considered. . . . .	130
4.8	Numerical discretizations of $\mathbb{D}_1$ and numerical solution associated to the boundary integral equation problem ( $\mathcal{G}$ ). . . . .	132
4.9	Numerical plots of the function $\varphi$ and the regularized solution $\hat{\varphi}_\eta$ . Both plots are truncated to the same area to showcase their behavior on the interior, and not at the edge of the disk where both solutions blow up to infinity. Here $\eta = 1e - 5$ , $h = 0.1$ . . . . .	133
4.10	Plots of the numerical errors defined in Section 4.9.4. In each plot, the $h$ , $\eta$ and error values have been scaled logarithmically. . . . .	135
4.10 (cont.)	Plots of the numerical errors defined in Section 4.9.4. In each plot, the $h$ , $\eta$ and error values have been scaled logarithmically. . . . .	136
4.11	Plots of the resolution error $\mathcal{R}(h, \eta)$ for different right hand sides. In each plot, the $h$ , $\eta$ and error values have been scaled logarithmically. . . . .	137

4.11 (cont.) Plots of the resolution error $\mathcal{R}(h, \eta)$ for different right hand sides. In each plot, the $h$ , $\eta$ and error values have been scaled logarithmically. . . . .	138
4.12 Plots of the approximated solutions for different values of $f$ . . . . .	139
4.12 (cont.) Plots of the approximated solutions for different values of $f$ . . . . .	140
4.13 Illustration of an electrosmootic mixer. Two fluids (represented by blue/red arrows) go in via the inflow compartment, they get mixed in the mixing chamber which has a specific boundary repartition of anode and cathode regions. The final fluid (represented by purple arrows) goes out via the outflow compartment. . . . .	142
4.14 Different view angles of the tetrahedral mesh $\mathcal{T}$ employed in the simulations. The red and pink regions correspond to the anode and cathode regions, respectively, which are fixed and not subject to optimization. The green region represents the flow region, which is also fixed and not subject to optimization. The remaining region corresponds to the insulated region (homogeneous Neumann boundary), which will be subject to optimization. . . . .	147
4.15 Snapshots of the optimization process of the boundary repartition. Here $i$ indicates the number of iterations that have passed. The blue color corresponds to the cathode region and the orange color the anode region. . . . .	148
4.15 (cont.) Snapshots of the optimization process of the boundary repartition. In figures (g) and (h) the perspective of the design is shown in order to better appreciate the repartition of the boundary. In figure (i), the evolution of the criterion as a function of each iteration. . . . .	149
4.16 Results of the optimization process for the second experiment. . . . .	150
4.17 Results of the optimization process for the third experiment. . . . .	151
4.18 In the figure, the obstacle $\Omega$ is shown immersed within the medium $D$ . The fictional boundary $\Gamma$ is where the Sommerfeld radiation condition is imposed, and it depicts the incident wave $f$ and the scattered wave $u$ . . . . .	152
4.19 Illustration of the boundary normal setup for the scattering problem. . . . .	154
4.20 Convergence history of the criterion. . . . .	157
4.21 Different view angles of the tetrahedral mesh $\mathcal{T}$ which serves as the obstacle in the medium $D$ . The wingspan of the aircraft is 100 units wide, the rear wing is 21 units high, the cockpit is 8 units high and the aircraft itself is 110 units long. The boundary of the obstacle consists of only sound-hard material. . . . .	158
4.22 Different view angles of the tetrahedral mesh $\mathcal{T}$ which serves as the computational domain $D$ . The outermost layer $\partial D$ , is the boundary of the domain. The middle layer acts as the artificial boundary $\Gamma_E$ , where the Sommerfeld radiation condition is imposed. The innermost layer is just the boundary $\partial\Omega = \Gamma \cup \Gamma_R$ of the obstacle $\Omega$ . The outer box is 200 units high, 300 wide, and 500 units long. The inner box is 100 units high, 150 units wide, and 250 units long. . . . .	159
4.23 Snapshots of the optimization process of the sound-soft optimization process. Here $i$ indicates the number of iterations that have passed. The pink color corresponds to the sound-soft boundary, while the rest corresponds to the sound-hard boundary. . . . .	160
4.23 Snapshots of the optimization process of the sound-soft optimization process. Here $i$ indicates the number of iterations that have passed. The pink color corresponds to the sound-soft boundary, while the rest corresponds to the sound-hard boundary. . . . .	161
4.23 Snapshots of the optimization process of the sound-soft optimization process. Here $i$ indicates the number of iterations that have passed. The pink color corresponds to the sound-soft boundary, while the rest corresponds to the sound-hard boundary. . . . .	162
4.24 Top and bottom views of the final result ( $i = 142$ ) of the experiment in Section 4.12. . . . .	163
4.25 Side views of the final result ( $i = 142$ ) of the experiment in Section 4.12. . . . .	164
4.26 Perspective views of the final result ( $i = 142$ ) of the experiment in Section 4.12. . . . .	165
4.26 Magnitude of the scattered pressure field, with the repartition of sound-soft boundary found by the algorithm. . . . .	166
4.26 Bottom cross-section view of the magnitude of the scattered pressure field $u_{\Gamma_R}$ . . . . .	167
4.27 A supported water tank typically features a cylindrical body made of materials like steel, concrete, or polyethylene, with a flat or slightly curved bottom, and a domed or flat top equipped with an access hatch or manway. The tank is elevated on legs made of, typically made steel or reinforced concrete, whose number and thickness depend on the tank's size and load requirements. To enhance stability, cross braces may connect the legs, forming a lattice structure, and the legs are anchored to a solid base plate or foundation to distribute the load evenly and prevent sinking or tilting. . . . .	168

4.28	Different view angles of the tetrahedral mesh $\mathcal{T}$ employed in the simulations. The major radius and height of the water tank are both 15 units in length, while the minor radius is 11 units wide. The blue region in the picture corresponds to $\Gamma_F$ which is the region that will not be optimized and remains traction free. The small pink region on the bottom of the tank corresponds to the initial support region. . . . .	171
4.29	Snapshots of the optimization process of the boundary repartition for the first experiment. Here $i$ indicates the number of iterations that have passed. The blue color corresponds to the region not being optimized, the pink region to the colour of the support region on the tank. . . . .	172
4.29 (cont.)	Snapshots of the optimization process of the boundary repartition for the first experiment. Here $i$ indicates the number of iterations that have passed. The blue color corresponds to the region not being optimized, the pink region to the colour of the support region on the tank. . . . .	173
4.30	Final design of the supported region along with convergence history, for the first experiment. The bottom view of the tank is displayed to better illustrate the distribution of the supporting material in the areas prioritized by the algorithm. . . . .	174
4.31	Snapshots of the optimization process of the boundary repartition for the second experiment. Here $i$ indicates the number of iterations that have passed. The blue color corresponds to the cathode region and the orange color the anode region. . . . .	175
4.31	Snapshots of the optimization process of the boundary repartition for the second experiment. Here $i$ indicates the number of iterations that have passed. The blue color corresponds to the region not being optimized, the pink region to the colour of the support region on the tank. . . . .	176
4.31	Final design of the supported region along with convergence history, for the second experiment (with contour penalization). The bottom view of the tank is displayed to better illustrate the distribution of the supporting material in the areas prioritized by the algorithm. . . . .	177
4.32	A clamp-locator system is essential in manufacturing and machining operations, particularly in CNC machining, to hold and position workpieces accurately and securely. It comprises clamps to apply pressure and locators to ensure precise positioning, enhancing accuracy, repeatability, and safety. This system is vital for operations requiring high precision, such as drilling, milling, assembly, and inspection, ensuring uniformity and efficiency in mass production by reducing setup time and maintaining consistent positioning. . . . .	178
4.33	Different view angles of the tetrahedral mesh $\mathcal{T}$ employed in the clamp-locator simulations. The pink region on the bottom and top of the mechanical piece represents the region $\Gamma_F$ , which is traction free and not subject to optimization. The blue region corresponds to the initial locator region $\Gamma_D$ . The rest is the traction free boundary $\Gamma$ . . . . .	182
4.34	Angles of the front and back of the mechanical piece. The blue region corresponds to the initial guess of the locator region $\Gamma_D$ . It is 2 units wide in diameter. The orange circle contained within the pink region $\Gamma_F$ represents the region $\Gamma_T$ where the tools is applying the force. . . . .	183
4.35	Snapshots of the optimization process of the boundary repartition. Here $i$ indicates the number of iterations that have passed. The blue color corresponds to the locator region $\Gamma_D$ and the green color represents the clamp region $\Gamma_N$ . . . . .	184
4.35 (cont.)	Snapshots of the optimization process of the boundary repartition. Here $i$ indicates the number of iterations that have passed. The blue color corresponds to the locator region $\Gamma_D$ and the green color represents the clamp region $\Gamma_N$ . . . . .	185
4.35 (cont.)	Snapshots of the optimization process of the boundary repartition. Here $i$ indicates the number of iterations that have passed. The blue color corresponds to the locator region $\Gamma_D$ and the green color represents the clamp region $\Gamma_N$ . . . . .	186
4.36	Final design of the clamp-locator design for the mechanical piece along with the objective convergence history. . . . .	187
5.1	Mesh entities in a mesh of dimension $D = 2$ . The yellow triangle represents the polytope of highest dimension, the red lines are referred to as the edges (or faces) of the mesh, while the 0 dimensional points are the vertices. . . . .	192
5.2	Transformation from a reference element $\hat{\tau}$ to the physical element $\tau$ . . . . .	192
5.3	Visualization of the solution found by Listing 1. . . . .	198
5.4	Visualization of the solution outputted in the Helmholtz example . . . . .	209

5.5	Linear elasticity solution outputted by Listing 3 . . . . .	210
D.1	Example of a vector field $\theta$ on $M$ . The value of a vector field $\theta(p)$ can be regarded as a pair $(p, v)$ , which satisfies the relation $\pi(\theta(p)) = p$ . . . . .	243

# List of Algorithms

---

1	Body-fitted optimization of the compliance for a shape $\Omega$ . . . . .	50
2	Body-fitted tracking of the evolution of a region $G(t) \subset S$ . . . . .	82
3	Optimization of the region $G$ bearing homogeneous Dirichlet boundary conditions. . . . .	104
4	Shape and topology optimization of the region $G \subset \partial\Omega$ . . . . .	141

# Acknowledgments

---

This work is the product of many minds, hearts, hands, parties, beers, and workdays that seemed to never end. They say that the friends you meet along the way are the ones who make you appreciate the journey, and I can say that's absolutely true. Let me express my sincere gratitude to some of these incredible people.

Charles Dapogny, Eric Bonnetier, Rafael Estevez. First and foremost, to my advisor, Charles Dapogny—thank you for being a constant source of wisdom, guidance, and support. Your belief in me, even during the toughest moments, kept me going. You have not only shaped my research but also how I approach challenges, and for that, I will always be grateful. From the master's degree, you have always been there to support me and share your vast knowledge! Thank you for treating me as a colleague rather than just another student, and for all those cups of coffee. Secondly, to my other advisors, Eric Bonnetier and Rafael Estevez. Eric, thank you for always being calm, reassuring, and for making me feel part of your warm community. I could always count on you, even towards the end during the writing. Thank you for the mathematical discussions and your incredible patience. Rafa, thank you (very specifically!) for telling me where to take salsa lessons. Your advice and warm smiles were a constant reassurance to lighten up and have some fun, especially when I needed it the most.

Amine Abdellaziz, Dmitrii Zhemchuzhnikov. My hermanos, you know how much I love you both, so I won't say much. Thank you for embarking on this journey with me. You were there from the very start of the master's, and you stayed right to the end. You have my deepest respect and appreciation.

Margaux Leroy, Jean-Baptiste Keck, Manon Vialle. To the original party team—thank you for always giving me a reason to celebrate! Even when I didn't have an excuse to go out, you guys certainly found one. Beyond the fun, I deeply appreciate the role you played in helping me with my French and understanding your culture. Thank you!

Qiao Chen, Benjamin Zanger, Ignacia Cancino, Kris Linder, Claudia Fonte. AKA The Latino-Danish-German Society. Thank you for the best memories, the laughs, the fights, the discussions, the mountain trips, and those tower-coffee sessions. Qiao, thank you for being your adventurous self. Benji, you are hands down the kindest person I've met. Nachi, gracias por todo el chisme y las risas. Kris, thank you for carrying me down the stairs when I twisted my ankle. Claudia, thank you for being my confidante and your emotional support through it all.

Thank you to my rapporteurs, Ulisse Stefanelli and Marc Bonnet, for playing key roles during my thesis defense. A big thank you also to Fioralba Cakoni and Michael Vogelius for giving me the opportunity to collaborate at Rutgers. And Tonatiuh Sanchez, thank you for hosting me in Arizona!

To my friends and colleagues at the IMAG, thank you for your support and camaraderie. You've made this journey so much more enjoyable with your laughter, advice, and sense of community, brightening the highs and softening the lows. A few special mentions: Nils Laurent, Hubert Leterme, Anatole Gaoullet, Yu guan Hsieh, Valentina Schuller, Kajal Eybpoosh, Yunjiao Lu, Lucia Bautista, Alexandre Delporte, Laura Alonzo, Basile Dubois, Remi Vuillemot, Ieva Petro, Victor Merckle, and Manolis Perrot—thank you all.

To my family—thank you for your endless patience and love. You have been my anchor throughout this process, even when I was consumed by work. Your unwavering support has been my foundation, and I couldn't have done this without you.

Finally, a heartfelt thank you to everyone who crossed paths with me during this journey. Whether we worked together or shared a moment of understanding, you have left your mark on this thesis in more

ways than you know. My only regret is not having more time to spend with each of you.

# Résumé de la thèse

---

## Qu'est-ce que l'optimisation de forme et de topologie ?

Dans l'ingénierie moderne, la quête de conceptions optimales a conduit à des avancées significatives dans le domaine de l'optimisation de forme et de topologie. Ces méthodologies sont essentielles pour créer des conceptions qui répondent à des critères de performance spécifiques tout en minimisant l'utilisation de matériaux. Alors que des industries telles que l'aérospatiale, l'architecture et l'ingénierie biomédicale repoussent sans cesse les limites du possible, la demande pour des approches de conception innovantes atteint des niveaux sans précédent.

L'optimisation de forme se concentre sur le raffinement de la géométrie d'une structure, tandis que l'optimisation de topologie permet une réimagination radicale de la répartition des matériaux dans un espace de calcul fixe. Les deux approches ont une large applicabilité dans divers domaines, offrant des perspectives sur des phénomènes naturels et inspirant même la formulation de nouveaux problèmes mathématiques. Au-delà de la recherche théorique [186, 320, 70], ces techniques trouvent des applications pratiques dans des domaines aussi variés que l'architecture [327, 352, 48], la géologie [187], la science des matériaux [181, 72, 73], la mécanique des fluides [241, 242, 107, 144], et la vision par ordinateur ainsi que le traitement d'images [126, 125, 124].

Le processus d'optimisation d'une forme implique plusieurs étapes clés. Dans un premier temps, un objectif spécifique doit être défini — qu'il s'agisse de minimiser l'utilisation des matériaux, de réduire la traînée, de maximiser la résistance ou d'améliorer le transfert de chaleur. Ensuite, les paramètres physiques et les particularités influençant la forme, tels que les dimensions géométriques, les propriétés des matériaux ou les conditions aux limites, doivent être identifiés. Il est également crucial d'imposer des contraintes, qu'elles soient physiques, mécaniques ou géométriques, via des algorithmes d'optimisation appropriés. Les techniques couramment utilisées incluent les méthodes basées sur le gradient et les algorithmes évolutionnaires, chacun offrant des avantages uniques en fonction des exigences spécifiques du problème. Des exemples spécifiques obtenus grâce à ces techniques sont illustrés dans la Fig. 1.

Le développement continu et l'application de ces stratégies d'optimisation promettent d'améliorer considérablement notre capacité à concevoir et à fabriquer des structures à la fois efficaces, durables, et adaptées aux exigences évolutives du futur. Le besoin de répondre à ces défis constitue la principale motivation qui sous-tend notre travail.

## Pourquoi étendre ces techniques aux surfaces ?

Récemment, il y a eu un intérêt croissant pour l'extension des techniques d'optimisation, en se concentrant spécifiquement sur les domaines au sein des surfaces. Ce changement met l'accent sur l'optimisation des surfaces plutôt que sur l'ensemble du domaine, ce qui est particulièrement pertinent dans les applications où les propriétés de surface sont plus critiques que les caractéristiques globales. La plupart des techniques existantes d'optimisation de forme et de topologie pour les surfaces sont généralement appliquées aux coques et membranes [91]. En bref, les coques sont des structures très fines, telles que la coque d'un navire, les toits de bâtiments et les fuselages d'avions. Un travail fondamental a déjà été réalisé dans [3, 148], qui utilise les déformations de maillages simpliciaux, tandis que certains travaux récents [318, 338]

adoptent une approche basée sur une grille. Cependant, l'optimisation de surface ne doit pas se limiter aux structures en coque, car ses applications potentielles sont vastes et diverses :

- **Transfert de chaleur.** Dans les systèmes de refroidissement, la conception des surfaces des échangeurs de chaleur joue un rôle crucial dans l'efficacité du transfert thermique. En optimisant la forme et la topologie de ces surfaces, il est possible d'améliorer considérablement la performance thermique, de réduire l'utilisation des matériaux et d'améliorer la gestion globale de la chaleur dans les applications électroniques et industrielles. Ce sujet a suscité beaucoup d'intérêt dans des études récentes [203, 33].
- **Aérodynamique et hydrodynamique.** Les flux d'air et de fluide autour des surfaces telles que les ailes d'avion [232], les coques de bateaux [160], ou les pales d'éoliennes [334] peuvent être optimisés pour réduire la traînée et améliorer l'efficacité de la propulsion ou de la portance. L'optimisation topologique des surfaces permet de concevoir des structures plus légères et plus efficaces en prenant en compte le comportement du flux sur la surface.
- **Acoustique.** La performance acoustique est souvent influencée par la forme et les propriétés de la surface. L'optimisation des surfaces peut réduire les bruits indésirables ou améliorer les caractéristiques sonores souhaitées dans des applications telles que la conception de véhicules [224, 271], les salles de concert [272] et les structures anti-bruit [119, 129].

Malgré l'intérêt croissant pour ce domaine, il n'existe toujours pas de cadre complet pour aborder ce type de problèmes. Il reste beaucoup de travail à accomplir, notamment pour identifier des représentations de forme efficaces sur les surfaces et développer des techniques mathématiques robustes pour résoudre ces défis. C'est dans cette optique que la principale motivation de cette thèse est d'offrir des techniques générales, numériques et théoriques, permettant de conduire l'optimisation de forme et de topologie des domaines sur des surfaces plongées dans  $\mathbb{R}^d$ .

## Portée de la thèse

Cette thèse se concentre sur l'optimisation de régions plongées dans une hypersurface de  $\mathbb{R}^d$ . Du point de vue conceptuel, la recherche revisite des méthodes classiques de conception optimale, telles que la technique de variation de frontière d'Hadamard et le concept de dérivée topologique, que nous adaptons au contexte surfacique étudié. Parallèlement, nous proposons un cadre théorique plus général ancré dans la géométrie riemannienne, pour l'optimisation générale d'une région au sein d'une variété riemannienne. L'application numérique de ces bases théoriques nécessite une stratégie numérique conçue pour suivre les mouvements à grande échelle de régions sur des surfaces tridimensionnelles sous des champs de vitesse complexes. L'approche adoptée dans cette thèse intègre un maillage géométrique précis avec la méthode des ensembles de niveaux, permettant des calculs précis et capturant des déformations significatives, y compris des changements topologiques. La discussion se concentre sur l'optimisation de régions qui supportent des conditions aux limites spécifiques dans la formulation d'un problème aux valeurs aux limites physique. Cela implique l'analyse d'équations aux dérivées partielles avec des conditions aux limites de Dirichlet, Neumann et Robin pour optimiser la forme et la topologie de ces régions. Les applications pratiques de ces techniques sont démontrées dans des scénarios tels que les systèmes de refroidissement de composants mécaniques, les systèmes de positionnement avec pinces, les supports structurels et les dispositifs d'invisibilité acoustique. L'ossature de ces applications pratiques motive la mise en œuvre d'une bibliothèque open-source en C++20 spécifiquement développée pour l'optimisation de forme. Les paradigmes de conception de la bibliothèque sont discutés, avec des exemples d'implémentation axés sur l'optimisation de surfaces, ainsi qu'un aperçu de sa syntaxe et des suggestions pour des améliorations futures. En résumé, cette thèse étend le champ d'application des techniques d'optimisation de forme et de topologie au domaine des régions sur des surfaces dans  $\mathbb{R}^d$ , faisant progresser à la fois la compréhension théorique et pratique de la manière dont ces optimisations peuvent être appliquées à des géométries complexes dans divers contextes d'ingénierie et de conception.

## Résumé des chapitres

### Chapitre 1 : Une revue de l'optimisation de forme et de topologie

Ce chapitre offre une revue complète des méthodes classiques d'optimisation de forme et de topologie, retraçant leur développement historique, leurs applications et les avancées théoriques qui ont façonné ce domaine. Nous explorons l'évolution de ces techniques, devenues centrales dans les pratiques de conception modernes, en mettant en lumière leurs fondements mathématiques, tels que l'analyse par éléments finis, l'analyse de sensibilité et les algorithmes d'optimisation. Tout au long du chapitre, nous examinons les distinctions et les synergies entre l'optimisation de forme et de topologie, en offrant des perspectives sur la manière dont leur utilisation combinée peut conduire à des solutions de conception plus optimales.

Nous commençons par un aperçu des techniques classiques d'optimisation de forme et de topologie. Après une brève introduction historique dans la [Section 1.1](#), nous présentons la célèbre méthode de variation de frontière d'Hadamard dans la [Section 1.2](#), en expliquant le calcul des dérivées de forme ainsi que quelques résultats clés. Dans la [Section 1.3](#), nous détaillons comment les directions de descente lisse peuvent être déduites de ces données. Ensuite, nous introduisons le concept de dérivée topologique dans la [Section 1.4](#), suivi d'une discussion sur la méthode des ensembles de niveaux, largement utilisée, et son application dans une approche « adaptée au corps » dans la [Section 1.5](#). Pour illustrer ces méthodes en pratique, nous résolvons le problème classique du porte-à-faux dans la [Section 1.6](#). Dans la [Section 1.7](#), nous abordons les limites et les variantes de ces techniques, comme le phénomène d'homogénéisation, et enfin, dans la [Section 1.8](#), nous passons en revue et comparons les travaux existants qui ont étendu ces méthodes d'optimisation aux surfaces, en évaluant leurs contributions par rapport à cette thèse.

### Chapitre 2 : Vers un cadre pour l'optimisation des domaines sur les variétés

Dans ce chapitre, notre objectif est d'établir un cadre pour l'optimisation géométrique de régions sur des sous-variétés au sein d'une variété ambiante générale  $M$ . Nous adoptons le cadre de la géométrie différentielle riemannienne, ce qui nous permet de tirer parti de divers résultats classiques de ce domaine. Dans ce cadre, nous apportons des démonstrations de plusieurs résultats bien connus dans la littérature sur l'optimisation de forme. De plus, nous explorons différentes méthodes pour optimiser des géométries dans divers contextes.

Tout au long de ce chapitre, nous utilisons principalement la méthode de variation de frontière initialement introduite par Hadamard dans son article fondateur [177]. Cette méthode a été largement étudiée dans des travaux comme [22, 116, 185, 184, 301]. Dans un cadre riemannien, nous utilisons l'application exponentielle, qui fournit une représentation équivalente de l'approche dite de « perturbation de l'identité » [184, 301]. Cette approche est l'une des plus couramment utilisées dans la littérature. De plus, ce choix de représentation peut être vu comme un analogue de la « méthode des vitesses » [116, 354, 355], où un champ de vecteurs  $\theta$  sur  $\mathbb{R}^n$  induit un flot représentant la trajectoire d'une particule évoluant sur la frontière. Nous n'approfondirons pas les détails de ces méthodes classiques ici, car elles sont largement discutées dans les ouvrages cités.

Il convient de noter que des idées similaires à celles présentées dans ce chapitre ont été explorées dans [311], où le théorème de structure d'Hadamard est démontré sur des sous-variétés, et où les surfaces fissurées sont étudiées dans ce modèle. Plus récemment, [285] a utilisé des idées équivalentes à celles présentées dans ce chapitre pour le calcul de dérivées de forme directionnelles d'ordre supérieur sur des sous-variétés. Cependant, ces travaux diffèrent légèrement de notre cadre, car ils considèrent spécifiquement des sous-variétés de l'espace ambiant  $\mathbb{R}^n$ , tandis que nous considérons l'espace ambiant comme une variété riemannienne complète  $M$ , ce qui conduit naturellement à l'utilisation de l'application exponentielle comme forme équivalente de perturbation de l'identité ou de la méthode des vitesses. Ce choix nous permet de tirer parti de la théorie de la géométrie différentielle riemannienne, qui est bien développée et largement étudiée.

Notre sélection de cette théorie n'est pas inédite. Par exemple, l'étude [188], qui se concentre sur les dérivées de forme via les formes différentielles, présente plusieurs approches élégantes et avantages, notamment la description indépendante des coordonnées des modèles et la séparation claire des invariants sous les transformations homéomorphes. Dans notre étude, nous avons constaté que l'adoption d'une perspective riemannienne conduit généralement à des démonstrations plus élégantes et offre une compréhension plus intrinsèque des aspects géométriques de l'optimisation de forme. Cependant, cela se fait au prix d'une rigueur mathématique accrue en raison du langage de la géométrie différentielle riemannienne.

Enfin, il convient de noter qu’une perspective riemannienne, abordée différemment, a déjà été explorée dans [287], où l’ensemble des formes est considéré comme une variété riemannienne, et  $W^{1,\infty}(\mathbb{R}^n, \mathbb{R}^n)$  est regardé comme l’espace tangent. En général, l’objectif de cette approche est de considérer l’ensemble de toutes les formes comme une variété riemannienne abstraite de dimension infinie. Cela se distingue de notre cadre, qui voit les formes comme des sous-variétés d’une variété ambiante et cherche à les déformer en utilisant la méthode de variation de frontière.

Le chapitre est organisé comme suit : Dans Section 2.1, nous étendons la méthode de variation de frontière d’Hadamard au cas général d’une variété riemannienne  $M$ , en utilisant l’application exponentielle  $\exp : TM \rightarrow M$  pour relier le fibré tangent aux points de la variété. Ensuite, dans Section 2.2, nous discutons de l’utilisation des fonctions distance signées pour représenter les domaines plongés dans la variété riemannienne, et nous étendons plusieurs résultats classiques de l’optimisation de forme à ce cadre plus général. Enfin, dans Section 2.3, nous présentons des résultats généraux liés au calcul des dérivées de forme et à leurs formes (c’est-à-dire le théorème de structure d’Hadamard).

### Chapitre 3 : Évolution adaptée au corps sur une surface via une méthode d’évolution de maillage basée sur l’ensemble de niveaux

La vaste tâche de représenter l’évolution d’un domaine  $G(t) \subset \mathbb{R}^d$  (avec  $d = 2$  ou  $3$  en pratique) a suscité de nombreuses recherches mathématiques et numériques. Ce sujet joue un rôle central dans diverses disciplines appliquées, allant des graphismes informatiques et de la vision par ordinateur [79, 350] à la simulation numérique de phénomènes physiques tels que la propagation des fractures [60] ou la dynamique des interfaces fluides [90, 96], en passant par les problèmes inverses et l’optimisation de forme [22, 29, 69].

Plusieurs stratégies numériques ont été proposées pour aborder cette tâche, avec des avantages et des inconvénients concurrentiels. Cependant, toutes les implémentations font face à un défi majeur : dans des situations complexes et réalistes, le champ de vitesse  $V(t, x)$  qui pilote le mouvement de  $G(t)$  a une origine physique et dépend de caractéristiques géométriques de haut niveau de cette région (notamment la courbure de  $\partial G(t)$ ), ou de la solution d’un problème aux limites posé sur celle-ci. Il est donc notoirement difficile de trouver un cadre permettant une description précise de  $G(t)$  pour effectuer des calculs précis de ces quantités à tout moment  $t$ , ce qui exige idéalement un maillage exact et de haute qualité, tout en traitant de manière robuste son évolution.

À cet égard, les stratégies lagrangiennes, qui suivent l’évolution d’un maillage exact de  $G(t)$  en déplaçant ses sommets selon  $V(t, x)$  entre les itérations successives du processus, sont souvent compromises par une dégradation sévère de la qualité du maillage, qui devient rapidement invalide et incompatible avec les calculs, voir par exemple [132, 133]. Certes, plusieurs heuristiques améliorent la robustesse de cette pratique. Par exemple, on peut alterner les déformations du maillage avec des étapes de remaillage occasionnelles visant à améliorer sa qualité. De plus, on peut détecter et supprimer les éléments mal formés avant qu’ils ne conduisent à une dégénérescence complète, ou même modifier la vitesse des sommets internes du maillage pour réduire l’apparition de motifs se chevauchant, voir par exemple [37, 46, 75, 123, 341].

Dans cet esprit, la technique récente des complexes simpliciaux déformables a démontré la capacité de gérer des mouvements impressionnants en utilisant la formation d’éléments de maillage presque dégénérés près de la frontière du domaine comme déclencheur de changements topologiques, voir [89, 88, 239]. Plus récemment encore, la méthode X-mesh [240] procède en déplaçant les sommets du maillage de  $G(t)$  selon  $V(t, x)$  jusqu’à ce que la mesure de certains éléments atteigne zéro. Le mouvement est alors relayé entre les nœuds voisins tout en préservant la connectivité du maillage. Cette stratégie repose sur l’hypothèse que les problèmes aux limites peuvent être résolus efficacement sur des maillages comportant des éléments dégénérés, sous certaines hypothèses sur leurs rapports de forme. Malgré ces avancées remarquables, il est important de noter que de telles stratégies de déformation de maillage lagrangiennes sont généralement réservées à la description de mouvements « relativement petits » de l’ensemble  $G(t)$ .

Pour surmonter les faiblesses des stratégies lagrangiennes, les techniques eulériennes de capture d’interface reposent sur une description implicite de la forme en mouvement  $G(t)$ . Parmi celles-ci, la méthode des ensembles de niveaux, introduite dans [258], représente  $G(t)$  comme le sous-domaine négatif d’une « fonction d’ensemble de niveaux » auxiliaire  $\phi(t, \cdot) : \mathbb{R}^d \rightarrow \mathbb{R}$  définie dans tout l’espace ambiant  $\mathbb{R}^d$ , voir [257, 293]. Le domaine  $G(t)$  n’est jamais maillé explicitement, étant redécouvert à chaque itération du processus à partir des valeurs de  $\phi(t, \cdot)$ . Bien que cela permette de décrire des mouvements arbitrairement grands, une telle représentation implicite est malheureusement moins propice à la résolution précise d’équations aux dérivées partielles définies sur  $G(t)$ .

Ces questions centrales et populaires dans l’analyse numérique du mouvement d’un domaine  $G(t)$  ont

reçu étonnamment peu d'attention lorsque le milieu ambiant est une variété, notamment une surface  $S$  dans  $\mathbb{R}^3$ ; pourtant, ce contexte alternatif englobe de nombreuses applications intéressantes :

- Les flots géométriques, tels que le flot de courbure moyenne ou le flot de Willmore, où le champ de vitesse  $V(t, x)$  dépend de caractéristiques géométriques de haut niveau de  $G(t)$ , peuvent être adaptés au cas d'une région au sein d'une surface [316].
- Diverses opérations d'intérêt dans le domaine des graphismes informatiques sont commodément formulées en termes de l'évolution d'une région au sein d'une surface. Par exemple, un modèle populaire pour la génération de textures sur une surface repose sur la résolution d'une équation de réaction-diffusion, voir par exemple [319, 328]; de même, la segmentation d'images sur des surfaces peut être abordée grâce à une adaptation appropriée de l'algorithme de Chan-Vese [316].
- Divers problèmes d'évolution physique se produisent au sein d'une surface, voir par exemple [248] à propos de la solidification d'un film fluide mince sur un substrat de surface, et [315] sur la dynamique des changements de phase sur des surfaces en science des matériaux.
- Le souhait d'optimiser la forme de régions au sein d'une surface ambiante donnée se pose par exemple dans la conception optimale de coques [318], de dispositifs électroniques curvilignes [208], dans le contexte des écoulements de surface [117], ou dans l'identification de systèmes de fixation optimaux pour des structures mécaniques, voir par exemple [108, 333].

À notre connaissance, les premières simulations numériques de l'évolution d'une région  $G(t)$  au sein d'une surface  $S$ , proposées dans [87] et [204], portaient sur le flot de courbure géodésique. Celles-ci utilisent une version de la méthode des ensembles de niveaux adaptée à la donnée de  $S$  en tant que patch paramétré, ou en tant que graphe d'une fonction définie sur l'espace 2D, respectivement. Un cadre plus général est proposé dans [54] et [84], où la surface fermée  $S := \{x \in \mathbb{R}^d, \psi(x) = 0\}$  est représentée comme l'ensemble de niveau 0 d'une fonction fixe  $\psi : \mathbb{R}^d \rightarrow \mathbb{R}$ , et  $G(t) = \{x \in S, \phi(t, x) < 0\}$  est le sous-domaine négatif de  $S$  induit par une autre fonction d'ensemble de niveaux (dépendant du temps)  $\phi(t, \cdot) : \mathbb{R}^d \rightarrow \mathbb{R}$ . Ici et dans les investigations ultérieures dans ce cadre [53, 169, 281], l'équation régissant l'évolution de la fonction d'ensemble de niveaux  $\phi(t, \cdot)$  est formulée dans tout l'espace  $\mathbb{R}^d$  à l'aide d'opérateurs de projection. Une autre série de contributions [231, 278] exploite la méthode des points les plus proches de [230], dédiée à la résolution d'équations aux dérivées partielles sur des surfaces. Comme cette dernière ne nécessite que la donnée d'une application associant à tout point  $x \in \mathbb{R}^d$  un point le plus proche (en termes de distance euclidienne) de la surface  $S$ , ce cadre permet à  $S$  d'être ouvert.

Le présent chapitre vise à introduire une méthodologie numérique robuste pour suivre des mouvements arbitrairement grands d'une région  $G(t)$  au sein d'une surface ambiante  $S \subset \mathbb{R}^3$  – y compris des changements de topologie – tout en maintenant une représentation maillée exacte de cette région tout au long du processus. La stratégie proposée est une extension naturelle des travaux antérieurs [12, 13, 14] – qui étaient consacrés à l'évolution des domaines de l'espace euclidien  $\mathbb{R}^d$  – au contexte actuel où le milieu ambiant est une surface  $S$  dans  $\mathbb{R}^d$ . Elle combine deux représentations complémentaires de  $G(t)$  à chaque étape de l'évolution : d'une part,  $G(t)$  est discrétisé explicitement, sous la forme d'un sous-maillage d'une triangulation de surface de haute qualité  $\mathcal{T}$  de la surface ambiante  $S$ , ce qui permet de calculer précisément ses caractéristiques géométriques ou de résoudre les problèmes aux limites associés via la méthode des éléments finis – et ainsi d'évaluer avec précision le champ de vitesse  $V(t, x)$ . D'autre part,  $G(t)$  est décrit implicitement via la méthode des ensembles de niveaux, comme le sous-domaine négatif d'une fonction scalaire  $\phi(t, \cdot) : S \rightarrow \mathbb{R}$ , de sorte que des mouvements arbitrairement grands de  $G(t)$  peuvent être réalisés. Le point central de cette stratégie est un ensemble d'algorithmes de maillage efficaces et de schémas numériques permettant de passer d'une représentation à l'autre.

Ce chapitre est organisé comme suit. Dans la [Section 3.1](#), nous présentons plus en détail la question du suivi de l'évolution d'une région au sein d'une surface. Ensuite, dans la [Section 3.2](#), nous décrivons la stratégie numérique proposée pour réaliser cette tâche et nous en détaillons les principaux ingrédients ; nous discutons notamment du calcul de la fonction distance signée à une région sur une surface, de la résolution de l'équation d'évolution des ensembles de niveaux dans ce contexte, et de nos principales opérations de remaillage dédiées aux triangulations de surface. Fait intéressant, ces méthodes numériques sont implémentées dans des codes open-source qui peuvent être utilisés facilement de manière « boîte noire ». Quelques applications numériques de notre cadre sont présentées dans [Sections 3.3 to 3.5](#) : après avoir évalué son efficacité sur un cas de test analytique, nous considérons le mouvement d'une interface complexe représentant un front de feu, dont l'expansion au sein d'un paysage fixe est pilotée par des quantités géométriques attachées au feu et à celles du paysage.

## Chapitre 4 : Optimisation de forme et de topologie des régions supportant des conditions aux limites

Les préoccupations croissantes causées par la rareté des ressources matérielles et la nécessité flagrante de réaliser des économies d'énergie ont rendu les techniques d'optimisation de forme et de topologie plus actuelles que jamais au sein des communautés académiques et industrielles, où elles trouvent des applications dans des domaines aussi variés que la mécanique des structures [52, 298], le génie civil et l'architecture [3, 48], la mécanique des fluides [6, 58], l'électromagnétisme [158, 196, 218, 251], et le génie biomédical [191, 270, 351].

Dans les cas classiques de tels problèmes, la conception étudiée est un domaine « massif »  $\Omega$  dans  $\mathbb{R}^d$  ( $d = 2, 3$ ), qui est optimisé par rapport à un critère de performance  $J(\Omega)$ , sous des contraintes telles que son volume ou son périmètre. En pratique,  $J(\Omega)$  dépend généralement d'une fonction d'« état » physique  $u$ , caractérisée comme la solution d'une équation aux dérivées partielles posée sur  $\Omega$ . La plupart du temps, les régions de  $\partial\Omega$  supportant des conditions aux limites spécifiques associées à cette équation sont imposées par le contexte et ne font pas l'objet d'une optimisation.

Le présent chapitre étudie des problèmes de conception optimale où la variable est précisément l'une de ces régions de  $\partial\Omega$  supportant un type particulier de conditions aux limites dans la formulation du problème physique. Parmi les diverses instances de ces questions, citons les suivantes :

- En électrostatique,  $\Omega$  représente un conducteur, et le potentiel de tension  $u : \Omega \rightarrow \mathbb{R}$  est la solution de l'équation de conductivité. Il est mis à la terre sur un sous-ensemble  $\Gamma_D$  de  $\partial\Omega$ , et un flux  $g : \Gamma_N \rightarrow \mathbb{R}$  est imposé sur une région disjointe  $\Gamma_N \subset \partial\Omega$  : ces effets sont modélisés par une condition de Dirichlet homogène sur  $\Gamma_D$  et une condition de Neumann inhomogène sur  $\Gamma_N$ . La partie restante  $\Gamma$  de  $\partial\Omega$ , qui est isolée de l'extérieur, est soumise à une condition aux limites de Neumann homogène. Bien que  $\Gamma_D$  et  $\Gamma_N$  soient généralement fixes, on peut souhaiter minimiser l'amplitude du champ électrique dans  $\Omega$  en fonction de leur emplacement sur  $\partial\Omega$ .
- En acoustique,  $u : \Omega \rightarrow \mathbb{R}$  est la pression acoustique dans une salle  $\Omega$ , solution de l'équation de Helmholtz. La frontière  $\partial\Omega$  est décomposée en deux régions  $\Gamma_N$  et  $\Gamma_R$  : des conditions aux limites de Neumann sont imposées sur  $\Gamma_N$ , où une onde incidente subit une réflexion parfaite, tandis que  $\Gamma_R$  porte des conditions aux limites de Robin, tenant compte d'une absorption partielle de cette onde. On peut alors se demander comment disposer  $\Gamma_N$  et  $\Gamma_R$  sur  $\partial\Omega$  pour minimiser la pression acoustique dans  $\Omega$ .
- En mécanique des structures,  $\Omega$  est une pièce mécanique, attachée sur un sous-ensemble  $\Gamma_D$  de sa frontière  $\partial\Omega$ , et soumise à des charges de surface  $g : \Gamma_N \rightarrow \mathbb{R}^d$ , appliquées sur une région disjointe  $\Gamma_N \subset \partial\Omega$  ; le champ vectoriel  $u : \Omega \rightarrow \mathbb{R}^d$ , représentant le déplacement de la structure, est la solution du système d'élasticité linéaire. Habituellement,  $\Gamma_D$  et  $\Gamma_N$  sont fixés par le contexte, et seule la partie restante, sans traction,  $\Gamma$ , est optimisée. Cependant, il peut être pertinent d'optimiser l'emplacement de la région de fixation  $\Gamma_D$  afin de minimiser le déplacement de la structure.

Ces questions s'inscrivent dans le cadre général de l'optimisation de forme d'un sous-ensemble  $G$  d'une surface ambiante fixe  $S \subset \mathbb{R}^d$ . Les premières études dans ce contexte sont consacrées à la simulation des flots géométriques sur  $S$ , notamment le flot de courbure moyenne.

## Chapitre 5 : Rodin : Une bibliothèque numérique en C++20 pour l'optimisation de forme et de topologie

Ce chapitre présente la conception et l'implémentation de **Rodin**, un cadre éléments finis léger et modulaire, développé pour supporter les algorithmes numériques utilisés dans cette thèse. La bibliothèque est conçue pour l'optimisation de forme et de topologie, offrant des outils pour le raffinement et le remaillage des formes, ainsi que des mécanismes pour spécifier et résoudre des problèmes variationnels. En employant des paradigmes de programmation intuitifs, Rodin permet des flux de travail d'optimisation efficaces et flexibles, tout en gardant la structure du code claire et compréhensible.

Rodin est disponible sous la licence **Boost Software License 1.0** à l'adresse suivante :

<https://github.com/cbritopacheco/rodin>

et la documentation est disponible à :

<https://cbritopacheco.github.io/rodin>.

Tous les résultats numériques présentés dans cette thèse ont été produits à l'aide de Rodin. La bibliothèque a été créée pour répondre à la difficulté de traduire des équations aux dérivées partielles (EDP) complexes en code informatique, un processus souvent sujet aux erreurs et aux inefficacités. Le cadre suit une philosophie d'alignement de sa structure sur la notation mathématique des EDP, facilitant ainsi une transition fluide entre la théorie et l'implémentation. Cet alignement offre plusieurs avantages :

1. **Facilité d'utilisation et compréhension intuitive** : La conception de Rodin reflète la structure mathématique des EDP, permettant aux utilisateurs de penser en termes familiers et de traduire facilement leurs modèles théoriques en code, réduisant ainsi la charge cognitive liée à l'abstraction.
2. **Réduction du taux d'erreur** : En maintenant une ressemblance étroite avec les formulations mathématiques originales, la probabilité d'erreurs lors de la transcription est minimisée, ce qui conduit à des simulations plus fiables et plus précises.
3. **Amélioration de la collaboration** : La syntaxe intuitive et basée sur les mathématiques permet à des experts de différents domaines (mathématiques, ingénierie, informatique) de collaborer, de réviser et de contribuer au code sans avoir besoin de connaissances approfondies en programmation.
4. **Valeur éducative** : Pour les étudiants et les nouveaux venus, Rodin sert de passerelle éducative entre les concepts théoriques des EDP et leur implémentation informatique, facilitant ainsi la compréhension des deux aspects.
5. **Correspondance directe entre théorie et code** : Rodin permet une implémentation directe des modèles théoriques, facilitant le développement et les tests dans des domaines complexes tels que la dynamique des fluides, l'électromagnétisme et la mécanique des structures.
6. **Flexibilité pour les modèles complexes** : L'alignement étroit du cadre avec la notation mathématique simplifie l'expérimentation avec des modèles non standard, des conditions aux limites et des propriétés des matériaux, offrant ainsi une plus grande flexibilité dans la recherche et le développement.
7. **Intégration avec les avancées théoriques** : À mesure que les méthodes numériques et l'analyse des EDP évoluent, l'adhésion de Rodin à la notation mathématique lui permet d'intégrer rapidement de nouvelles techniques, garantissant que les utilisateurs restent à jour avec les dernières méthodologies.

Tout au long de ce chapitre, des exemples d'application de Rodin dans l'optimisation de forme et de topologie sont présentés. La modularité et la flexibilité du cadre en font un outil essentiel pour résoudre efficacement des problèmes d'optimisation dans divers domaines d'ingénierie. En résumé, Rodin simplifie non seulement l'implémentation des algorithmes d'optimisation de forme et de topologie, mais il comble également le fossé entre la théorie mathématique et le calcul pratique, améliorant ainsi à la fois la précision et la convivialité dans les simulations numériques.

## État de l'art pour l'optimisation de formes et de topologie des surfaces

Malgré des avancées significatives dans le domaine de l'optimisation de formes et de topologie, la majorité des recherches se sont concentrées sur les domaines volumiques, avec une attention limitée accordée à l'optimisation d'une région sur une surface. Les méthodes existantes, examinées dans ce chapitre, sont principalement conçues pour optimiser des domaines  $\Omega \subset \mathbb{R}^d$ . Cependant, les problèmes basés sur les surfaces présentent des complexités supplémentaires en raison de la géométrie complexe des surfaces, nécessitant des techniques computationnelles spécialisées.

Pour aggraver la situation, et comme souligné dans l'introduction, l'optimisation de surfaces a des applications dans plusieurs domaines. En transfert de chaleur, l'optimisation des surfaces d'échangeurs thermiques améliore l'efficacité thermique, réduit l'utilisation des matériaux et renforce la gestion thermique globale [203, 33]. En aérodynamique et hydrodynamique, l'optimisation des surfaces, telles que les ailes d'avion, les coques de bateaux et les pales d'éoliennes, réduit la traînée et améliore l'efficacité de portance ou de propulsion [232, 160, 334]. En acoustique, la forme de la surface est cruciale pour le contrôle du son, qu'il s'agisse de réduire le bruit des véhicules ou d'améliorer l'acoustique des salles de concert [224, 272]. De plus, l'optimisation des structures à parois minces, telles que les coques et les

membranes utilisées dans les coques de navire, les toits et les fuselages d’avion, a progressé grâce aux méthodes basées sur les maillages simpliciaux et les grilles [91, 318, 338]. La diversité de ces applications souligne le besoin urgent d’un cadre global pour relever ces défis.

Cette thèse comble le fossé dans l’optimisation de formes et de topologie basée sur les surfaces à travers quatre contributions clés :

1. Développements théoriques et généralisations aux variétés, étendant les techniques d’optimisation classiques aux contextes des variétés riemanniennes.
2. Évolution numérique des domaines sur des hypersurfaces immergées dans  $\mathbb{R}^d$ , avec des stratégies computationnelles adaptées aux défis de l’optimisation des surfaces.
3. Optimisation des conditions aux limites pour diverses EDPs, introduisant de nouvelles méthodes pour optimiser les régions qui supportent des conditions aux limites dans différents contextes physiques.
4. Développement d’outils de programmation open-source facilitant la mise en œuvre des méthodes introduites dans cette thèse.

Maintenant, nous visons à détailler les travaux déjà existants dans la littérature et à préciser nos contributions dans ces domaines clés.

## Développements théoriques et généralisations aux variétés

---

*La principale contribution liée à ce sujet est contenue dans [Chapter 2](#).*

---

Nous introduisons un cadre novateur pour l’optimisation géométrique de régions sur des sous-variétés au sein d’une variété riemannienne ambiante générale  $M$ . Alors que la méthode classique de variation de frontière, introduite à l’origine par Hadamard [177], a été largement explorée dans les espaces euclidiens [22, 116, 185, 301], notre contribution étend cette approche au cadre riemannien, un contexte qui n’a pas été abordé de manière approfondie dans les travaux antérieurs. En employant l’application exponentielle dans une variété riemannienne, nous fournissons une représentation équivalente à la méthode de « perturbation d’identité », traditionnellement utilisée dans les espaces euclidiens, et l’adaptions aux sous-variétés dans un cadre géométrique plus général.

Bien que certains travaux antérieurs, tels que [311] et [285], aient exploré des idées similaires—comme l’application du théorème de Hadamard aux sous-variétés et le calcul des dérivées de forme directionnelles de second ordre—ces études se limitent aux sous-variétés de l’espace ambiant  $\mathbb{R}^n$ . En revanche, notre cadre est situé dans un cadre plus général de variété riemannienne complète, où l’utilisation de l’application exponentielle nous permet de prendre en compte la courbure intrinsèque de la variété, offrant une base plus polyvalente et rigoureuse pour l’optimisation de formes.

Notre approche s’appuie sur les avancées précédentes, telles que les travaux de [188], qui utilisent les formes différentielles pour les dérivées de forme, et offre un modèle indépendant des coordonnées et préservant les invariants. Bien qu’adopter une perspective riemannienne augmente la rigueur mathématique, nous soutenons que cela conduit à des démonstrations plus élégantes et à une compréhension plus profonde des aspects géométriques de l’optimisation de formes. De plus, ce cadre se distingue d’autres perspectives riemanniennes, telles que celle de [287], qui traite l’ensemble de toutes les formes comme une variété abstraite de dimension infinie. Au contraire, notre focus reste sur les sous-variétés d’une variété ambiante, en utilisant la méthode de variation de frontière pour les déformer.

En résumé, la principale contribution est le développement d’un cadre complet qui généralise les techniques classiques d’optimisation de formes au cadre riemannien, offrant de nouvelles perspectives et outils pour l’optimisation de régions sur des sous-variétés courbées, un sujet qui a été relativement peu exploré dans la littérature précédente.

## Évolution numérique des domaines sur des hypersurfaces immergées dans $\mathbb{R}^d$

---

*La principale contribution liée à ce sujet est contenue dans [Chapter 3](#).*

---

Le problème de la représentation et du suivi de l'évolution d'un domaine  $G(t) \subset \mathbb{R}^d$  (où  $d = 2$  ou  $3$ ) a été un sujet central dans plusieurs disciplines appliquées, telles que les graphiques informatiques, la dynamique des fluides, la propagation des fractures et l'optimisation de formes. De nombreuses méthodes numériques ont été développées pour relever ce défi, mais un problème commun demeure : équilibrer la représentation précise de  $G(t)$  avec la gestion robuste de son évolution, en particulier lorsque le champ de vitesse  $V(t, x)$  dépend de caractéristiques géométriques complexes comme la courbure ou la solution de problèmes aux limites.

Les méthodes lagrangiennes, qui suivent directement le mouvement du maillage, souffrent souvent d'une dégradation du maillage à mesure que le mouvement progresse, ce qui conduit à des géométries invalides et à des défis computationnels. Bien que des améliorations aient été apportées, comme les techniques de remaillage ou la modification des vitesses internes du maillage [37, 75], ces méthodes sont généralement limitées à de petites déformations. Certaines approches plus récentes, telles que le Complexe Simplicial Déformable [89] et la méthode X-mesh [240], ont repoussé les limites des méthodes lagrangiennes en abordant les grandes déformations et les changements topologiques. Cependant, ces techniques rencontrent encore des difficultés pour résoudre avec précision des problèmes aux limites sur des maillages dégénérés.

D'autre part, les méthodes eulériennes, en particulier la méthode des ensembles de niveaux [258], offrent un cadre plus flexible pour représenter les grandes déformations de  $G(t)$  sans avoir besoin de maillage explicite. Bien qu'elles puissent gérer des mouvements complexes, ces méthodes sont moins adaptées à la résolution des EDP sur le domaine évolutif en raison de la nature implicite de la représentation.

Alors que l'évolution des domaines dans l'espace euclidien a été largement étudiée, relativement peu de travaux ont été réalisés lorsque le milieu ambiant est une variété, telle qu'une surface  $S \subset \mathbb{R}^3$ . Cela est surprenant compte tenu du nombre d'applications importantes, incluant les flux géométriques, la génération de textures basées sur les surfaces en infographie, et les processus physiques se produisant sur les surfaces (par exemple, les changements de phase dans les matériaux). Bien que des efforts précurseurs comme [87] et [204] aient appliqué des méthodes d'ensembles de niveaux pour suivre les flux de courbure géodésique sur des surfaces, ces approches étaient limitées à des paramétrisations spécifiques ou des graphes de surfaces. Des cadres plus généraux ont été introduits plus tard, tels que ceux utilisant une fonction d'ensemble de niveaux fixe pour décrire la surface  $S$  et le domaine  $G(t)$  évoluant en son sein [54, 84], ou la méthode du point le plus proche pour résoudre des EDP sur des surfaces [230].

Notre principale contribution est une nouvelle méthodologie numérique qui fait progresser l'état de l'art en permettant de suivre des mouvements arbitrairement grands de  $G(t)$  sur une surface ambiante  $S \subset \mathbb{R}^3$ , tout en préservant à la fois la topologie et une représentation maillée de haute qualité tout au long du processus. En nous appuyant sur les travaux précédents sur l'évolution des domaines dans l'espace euclidien [12, 13, 14], nous étendons ces techniques au cas où le milieu ambiant est une surface. Notre approche combine deux représentations complémentaires :

1. Une discrétisation explicite de  $G(t)$  comme un sous-maillage d'une triangulation de haute qualité de  $S$ , permettant des calculs géométriques précis et la résolution de problèmes aux limites.
2. Une représentation implicite par ensemble de niveaux de  $G(t)$ , permettant des déformations et des changements topologiques arbitrairement grands.

L'innovation clé de notre travail réside dans les algorithmes de maillage efficaces et les schémas numériques qui permettent des transitions fluides entre ces deux représentations, surmontant ainsi les limitations des méthodes lagrangiennes et eulériennes des études précédentes.

## Optimisation des régions supportant les conditions aux limites pour différentes EDP

---

*La principale contribution liée à ce sujet est contenue dans [Chapter 4](#).*

---

L'optimisation des régions supportant les conditions aux limites pour diverses équations aux dérivées partielles (EDP) est un domaine en pleine expansion, avec de nombreuses applications, mais qui reste peu exploré. En élasticité, par exemple, l'optimisation des régions où des conditions aux limites de Dirichlet et de Neumann sont imposées (comme dans les systèmes de fixation et de localisateurs) peut améliorer la conception des points de fixation ou d'application de charges. Des études antérieures ont

utilisé des méthodes d'optimisation topologique basées sur la densité, des algorithmes génétiques, des réseaux neuronaux et des méthodes d'ensembles de niveaux [228, 201, 289, 332, 333, 345]. En acoustique, des recherches ont porté sur l'optimisation de la distribution de matériaux absorbants (conditions aux limites de Robin) afin de minimiser la pression sonore, comme démontré dans [119].

Des travaux récents, comme [318], utilisent couramment l'optimisation basée sur la densité sur des maillages fixes, tandis que les méthodes d'évolution de maillage adaptées au corps, dérivées de travaux tels que [14], ont été appliquées pour suivre de grandes déformations, comme l'illustre [234], qui a optimisé les valeurs propres de Neumann sur une sphère 3D. Cependant, ces méthodes sont souvent spécifiques à des cas particuliers et manquent d'un cadre unifié pour traiter l'optimisation de formes et de topologie des régions supportant des conditions aux limites. Ce manque laisse les cadres théoriques et les stratégies pour aborder les grandes déformations ou les changements topologiques relativement sous-développés, malgré l'importance de ces problèmes dans de nombreux domaines.

En s'appuyant sur les contributions théoriques de [108] et [56], notre travail propose un cadre complet pour l'optimisation de formes et de topologie des régions  $G \subset \partial\Omega$  supportant des conditions aux limites pour des EDP sur un domaine fixe  $\Omega$ . Le travail de [108] offre une étude rare sur l'optimisation de formes, se concentrant sur la sensibilité d'une fonctionnelle  $J(G)$  à de petites perturbations diffeomorphiques de  $G$ , tandis que [56] étend cette compréhension aux perturbations singulières où les conditions aux limites basculent entre Dirichlet et Neumann.

Notre contribution combine et étend ces approches pour développer un flux de travail robuste. Nous intégrons des dérivées de forme et topologiques pour évaluer la sensibilité de  $J(G)$  à la fois aux petites déformations de  $\partial G$  et aux perturbations singulières en introduisant de petites régions de surface. Bien que les calculs formels soient présentés dans le contexte de l'équation de conductivité, nos méthodes sont adaptables à des applications plus complexes, telles que l'acoustique et la mécanique des structures.

Une innovation clé de notre approche est l'application duale de l'analyse asymptotique : d'abord, pour lisser les transitions singulières entre les régions avec des conditions aux limites différentes, simplifiant ainsi le calcul des dérivées de forme ; et ensuite, pour explorer les perturbations singulières en introduisant de petites zones avec des conditions aux limites modifiées afin de quantifier la sensibilité topologique. Cette stratégie nous permet de proposer un cadre novateur, flexible et généralisable qui n'était pas encore disponible dans la littérature.

D'un point de vue théorique, nous proposons des méthodes formelles pour calculer à la fois les dérivées de forme et les dérivées topologiques. Ces méthodes, détaillées dans le contexte plus simple de l'équation de conductivité, peuvent être étendues à des scénarios plus complexes tels que l'acoustique et la mécanique des structures. Bien que les dérivées de forme soient relativement simples pour une gamme de problèmes, les dérivées topologiques nécessitent des adaptations plus complexes, que nous abordons en détail. Notre travail démontre finalement comment l'analyse asymptotique peut simplifier la répartition des conditions aux limites et permettre l'étude des perturbations singulières, élargissant ainsi l'applicabilité des techniques d'optimisation de formes et de topologie.

## Outils de programmation open-source pour le développement d'algorithmes d'optimisation de formes et de topologie

*La principale contribution liée à ce sujet est contenue dans [Chapter 5](#).*

La publication du code MATLAB d'optimisation topologique en 99 lignes, connu sous le nom de top99 [297], par Ole Sigmund en 2001 a marqué un moment décisif dans le domaine de l'optimisation structurelle et multidisciplinaire. Ce code concis et accessible est rapidement devenu un outil éducatif largement utilisé, inspirant de nombreuses contributions qui ont simplifié des algorithmes d'optimisation complexes en formats accessibles. L'impact de top99 a dépassé son utilité immédiate, favorisant une culture d'apprentissage, d'expérimentation et d'innovation au sein de la communauté de l'optimisation. Ces dernières années, le nombre d'outils éducatifs et d'articles dans ce domaine a augmenté régulièrement [324], avec de nombreux scripts ou outils simplifiés conçus pour des tâches spécifiques.

Bien que ces outils soient précieux pour illustrer des concepts de base et effectuer des optimisations simples, ils s'avèrent souvent insuffisants pour traiter les complexités des problèmes réels. De nombreuses tâches d'optimisation de forme impliquent des problèmes non linéaires [310], l'intégration avec des logiciels de simulation [4], la génération de maillages et des solveurs mathématiques variés [202], des défis que ces outils simplifiés ne peuvent pleinement prendre en charge. Des travaux récents, comme [104], ont introduit des bibliothèques et des dépôts utiles, mais ils viennent souvent avec de multiples dépendances

et nécessitent une expertise technique considérable. Des tendances similaires se retrouvent dans d'autres outils scientifiques comme [227, 107, 18], qui, bien qu'enrichis techniquement, sont mieux adaptés au prototypage académique qu'aux applications robustes et évolutives requises dans des scénarios plus complexes. Pour véritablement répondre aux besoins des ingénieurs et des chercheurs, les logiciels d'optimisation de formes doivent prioriser l'évolutivité, l'intégration et les performances. Sans une bibliothèque robuste et bien conçue, les utilisateurs sont contraints de développer des solutions ad hoc, entraînant des flux de travail inefficaces, un temps de développement accru et des résultats incohérents.

En réponse à ces limitations, nous introduisons Rodin, une bibliothèque C++ conçue pour fournir une solution complète pour l'optimisation de formes et de topologie. Contrairement aux outils existants, Rodin est un cadre modulaire et léger d'éléments finis qui intègre des algorithmes d'optimisation de formes et de topologie dans une seule interface de programmation (API). Rodin est conçu pour être évolutif, permettant aux utilisateurs de passer sans difficulté du prototypage à des problèmes plus complexes sans nécessiter une expertise technique approfondie ni de multiples dépendances.

Rodin offre une suite complète de fonctionnalités essentielles pour l'optimisation avancée de formes et de topologie, toutes intégrées en interne. Celles-ci incluent :

- Le raffinement de formes et le remaillage via MMG [103].
- Le calcul des distances des domaines à l'aide de MSHDIST [105].
- L'advection des fonctions d'ensembles de niveaux via une intégration avec la boîte à outils computationnelle ISCD [66, 106].

Ces outils ont été testés et éprouvés pendant des années dans le contexte de l'optimisation de formes et sont directement intégrés dans Rodin, étant donné qu'ils sont codés en C. En tant que bibliothèque C++, Rodin peut interagir nativement avec le C, assurant une intégration fluide et des performances optimales. Rodin introduit également une approche inspirée de FreeFem++ pour définir des problèmes variationnels, combinant prototypage rapide et convivialité sans sacrifier l'interopérabilité. Il utilise la méthode des éléments finis pour assembler les opérateurs de rigidité et les vecteurs de masse associés, souvent employés pour simuler des problèmes physiques. Cette approche améliore la facilité d'utilisation tout en maintenant la flexibilité nécessaire pour traiter des problèmes complexes. La bibliothèque fournit également un accès à une gamme de solveurs linéaires, y compris des solveurs de Krylov comme GMRES [279] et des solveurs directs comme UMFPACK [112], tous entièrement intégrés dans l'écosystème C++. Cela élimine le besoin de logiciels externes, avec des bibliothèques tierces directement incorporées dans le système de compilation de la bibliothèque.

À notre connaissance, Rodin est une contribution novatrice dans le domaine, aucune autre bibliothèque C++ n'offrant ce niveau d'intégration, de flexibilité et d'évolutivité pour l'optimisation de formes et de topologie.

## Contributions

Cette recherche a donné lieu à deux prépublications. La première est basée sur le Chapitre 3 et la seconde sur le Chapitre 4.

- C. Brito-Pacheco, C. Dapogny. *Suivi adapté au corps sur une surface via une méthode d'évolution de maillage basée sur l'ensemble de niveaux*, soumis (2023).
- E. Bonnetier, C. Brito-Pacheco, C. Dapogny. *Optimisation numérique de forme et de topologie des régions supportant des conditions aux limites*, en préparation (2024).

Ces travaux ont été présentés lors des événements suivants :

- CANUM 2024. Été 2024. *Optimisation de forme et de topologie des régions supportant des conditions aux limites*.
- WCSMO 2023. Été 2023. *Optimisation de forme et de topologie des régions supportant des conditions aux limites*.
- MFEM Community Workshop 2022. *Rodin : Un cadre moderne et léger en C++17 pour l'optimisation de forme, de densité et de topologie*.

# Notation

---

- $\mathbb{N}$  denotes the set of natural numbers.
- $\mathbb{R}$  denotes the real number line.
- $A^n$  denotes the  $n$ -product  $\prod_{i=1}^n A$  of the set  $A$
- Given a map  $f : A \rightarrow B$ , its image and domain are denoted by  $\text{Im}(f)$  and  $\text{Dom}(f)$ , respectively.
- $a \ll b$  is read as “ $a$  is much less than  $b$ ”.
- If  $\theta : \mathbb{R}^d \rightarrow \mathbb{R}^d$  is a vector field we may write  $\theta = (\theta_1, \dots, \theta_d)$ . Additionally, the  $i$ -th derivative of the  $j$ -th component is denoted  $\frac{\partial \theta_j}{\partial x_i}$  or in a shorthand manner  $\partial_{x_i} \theta_j$ .
- Measure of a set  $\Omega$  is denoted by  $|\Omega|$
- If a set  $\Omega$  is compactly contained in  $D$ , we denote it by  $\Omega \subset\subset D$ .
- $\mathcal{GL}_d(\mathbb{R})$  is the set of invertible square matrices in  $\mathbb{R}^{d \times d}$ .
- $A^{-1}$  is the inverse of  $A \in \mathcal{GL}_d(\mathbb{R})$ .
- $A^t$  is the adjoint or transpose of a real matrix  $A$ .
- $A^{-t} := (A^{-1})^t$
- The Fréchet differential of a mapping  $f$  at the point  $u$  in the direction  $\hat{u}$  is denoted by  $Df(u)(\hat{u})$ .
- The Sobolev space  $W^{1,\infty}(\mathbb{R}^d; \mathbb{R}^d)$  is known as the space of Lipschitz functions, endowed with norm

$$\|u\|_{W^{1,\infty}} := \sup_{x \in \mathbb{R}^n} (|u(x)| + |\nabla u(x)|) \quad (0.1)$$

- $H_{\Gamma}^p(\Omega) := \{u \in H^p(\Omega) \mid u = 0 \text{ on } \Gamma\}$ .
- $C_0^\infty(\Omega)$  denotes the space of smooth, compactly supported functions in  $\Omega$ .

# Introduction

---

## What is shape and topology optimization?

In modern engineering, the quest for optimal designs has driven significant advances in the field of shape and topology optimization. These methodologies are essential for creating designs that meet specific performance criteria while minimizing material usage. As industries such as aerospace, architecture, and biomedical engineering continually push the boundaries of possibility, the demand for innovative design approaches has reached unprecedented levels.

Shape optimization focuses on refining the geometry of a structure, while topology optimization allows for a radical reimagining of the material distribution within a fixed computational space. Both approaches have broad applicability across various domains, offering insights into natural phenomena and even inspiring the formulation of new mathematical problems. Beyond theoretical investigation [186, 320, 70], some examples of these diverse practical domains include architecture [327, 352, 48], geology [187], material science [181, 72, 73], fluid mechanics [241, 242, 107, 144], computer vision and image processing [126, 125, 124].

The process of optimizing a shape involves several key steps. Initially, a specific goal must be defined—whether minimizing material usage, reducing drag, maximizing strength, or enhancing heat transfer. Next, the physical parameters and particularities that influence the shape, such as geometric dimensions, material properties, or boundary conditions, must be identified. It is also crucial to enforce constraints, which may be physical, mechanical, or geometrical via appropriate optimization algorithms. Common techniques include gradient-based methods, evolutionary algorithms, each offering unique strengths depending on the specific requirements of the problem at hand. Specific examples obtained via these techniques are illustrated in Fig. 1.

The continuous development and application of these optimization strategies hold the promise of significantly improving our capacity to design and manufacture structures that are not only efficient but also sustainable and adapted to the evolving demands of the future. The need to provide answers to these challenges contribute the primary motivation that drives our work.

## Why extend these techniques to the case of surfaces ?

Recently, there has been growing interest in extending optimization techniques to focus specifically on domains within surfaces. This shift emphasizes surface optimization over the entire domain, which is particularly relevant in applications where surface properties are more critical than bulk characteristics. Most existing techniques in shape and topology optimization for surfaces are typically applied to shells and membranes [91]. In a nutshell, shells are very thin structures, such as the hull of a ship, building roofs, and aircraft fuselages. A lot of fundamental work has already been done in [3, 148] which employs simplicial mesh deformations, while some recent work [318, 338] utilizes a grid based approach. However, surface optimization need not be limited to shell structures, as its potential applications are broad and diverse:

- **Heat transfer.** In cooling systems, the design of heat exchanger surfaces plays a critical role in determining their heat transfer efficiency. By optimizing the shape and topology of these surfaces,

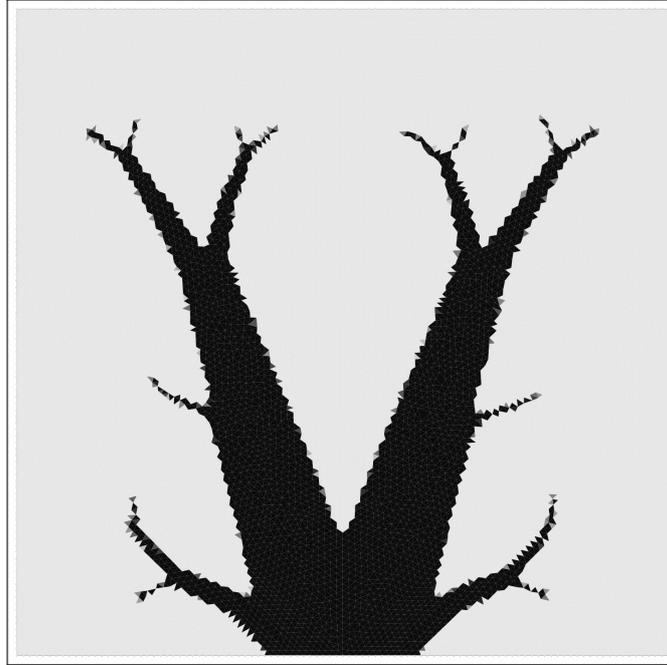
it is possible to significantly improve thermal performance, reduce material usage, and enhance overall heat management in both electronic and industrial applications. This topic has garnered considerable attention in recent studies [203, 33].

- **Aerodynamics and hydrodynamics.** Air and fluid flows around surfaces such as airplane wings [232], boat hulls [160], or wind turbine blades [334] can be optimized to reduce drag and improve lift or propulsion efficiency. Surface topology optimization enables the design of lighter, more efficient structures by accounting for flow behavior over the surface.
- **Acoustics.** Acoustic performance is often influenced by surface shape and properties. Optimizing surfaces can reduce unwanted noise or enhance desired sound characteristics in applications such as vehicle design [224, 271], concert halls [272], and noise-reducing structures [119, 129].

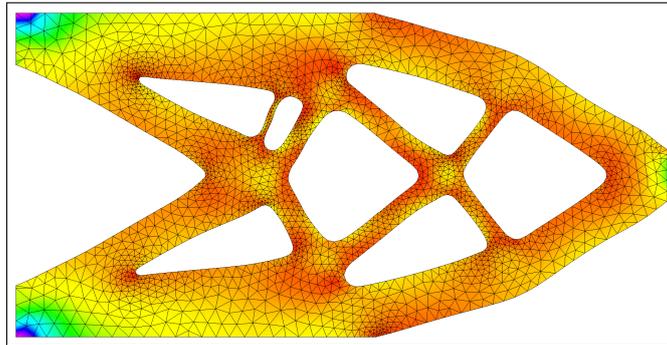
Despite the increasing interest in this area, there is still no comprehensive framework to address these types of problems. Significant work remains, particularly in identifying efficient shape representations on surfaces and developing robust mathematical techniques to solve these challenges. In light of this, the main motivation behind this thesis is to provide general, numerical and theoretical techniques to conduct the shape and topology optimization of domains in surfaces embedded in  $\mathbb{R}^d$ .

## Scope of the thesis

This thesis focuses on the optimization of regions embedded in a hypersurface of  $\mathbb{R}^d$ . From the conceptual viewpoint, the research revisits classical optimal design methods, such as Hadamard’s boundary variation technique and the concept of the topological derivative, which we adapt to the surfacic context at hand. Meanwhile we propose a more general theoretical framework framed within Riemannian geometry, for the general optimization of a region within a Riemannian manifold. The numerical application of this theoretical groundwork calls for a numerical strategy designed to track large-scale motion of regions on three-dimensional surfaces under complex velocity fields. The approach matched in this thesis integrates precise geometric meshing with the level set method, enabling accurate computations and capturing significant deformations, including topological changes. The discussion focuses on optimizing regions that support specific boundary conditions in the formulation of a physical boundary value problem. This involves analyzing partial differential equations with Dirichlet, Neumann, and Robin boundary conditions to optimize the shape and topology of these regions. Practical applications of these techniques are demonstrated in scenarios such as cooling systems of mechanical components, clamp-locator systems, in structural supports, and acoustic cloaks. The backbone of these practical applications, motivates the implementation of an open-source C++20 library specifically developed for shape optimization. The library’s design paradigms are discussed, with implementation examples focused on surface optimization, along with an overview of its syntax and suggestions for future enhancements. In summary, this thesis extends the range of applications of shape and topology optimization techniques to the realm of region within surfaces in  $\mathbb{R}^d$ , advancing both the theoretical and practical understanding of how these optimizations can be applied to complex geometries within a variety of engineering and design contexts.



(a) A tree-like structure minimizes material and cooling by efficiently distributing heat through branching pathways, which increases surface area for cooling and uses less material to achieve effective heat dissipation compared to a uniformly thick plaque. The design has been obtained via the SIMP method (see [51, 275]).



(b) A cantilever minimizes compliance (which is a measure of flexibility or deformation under load) by efficiently distributing the material to resist bending. The cantilever design typically maximizes stiffness at the fixed end, where the bending moment is greatest, and tapers toward the free end, where the moment is less. This design has been obtained via the level-set method in conjunction with shape derivatives (see §2 of [15] for the overall methodology).

Figure 1: Examples of optimal designs obtained via different shape and topology optimization techniques.

---



---

## Summary of chapters

### Chapter 1: A review of shape and topology optimization

In this preliminary chapter, we provide a general overview of classical techniques in shape and topology optimization. After some light historical context in [Section 1.1](#), we introduce Hadamard’s renowned method of boundary variation in [Section 1.2](#). There we explain how to compute shape derivatives and some of the principal results related to them, while in [Section 1.3](#) we explain how to perform the inference of smooth descent directions from this datum. In [Section 1.4](#), we explain the concept of the topological derivative. In [Section 1.5](#), we revisit the widely known level-set method and discuss its application in a “body-fitted” approach. To demonstrate these techniques in practice, we present the classical cantilever problem in [Section 1.6](#). Additionally, in [Section 1.7](#), we address the limitations and variations of these methods, providing a broader perspective on the field, including the phenomenon known as homogenization. Lastly, in [Section 1.8](#), we review and compare the existing techniques and works that have extended these optimization methods to surfaces, and we evaluate these contributions in relation to those made in this thesis.

### Chapter 2: Towards a framework for the optimization of domains on manifolds

In this chapter, we establish a framework for the geometric optimization of regions on submanifolds within a general ambient Riemannian manifold  $M$ . By utilizing classical results from Riemannian differential geometry, we rigorously prove well-known results in shape optimization and explore various methods for optimizing geometries in different contexts. The primary method used is the boundary variation technique, originally introduced by Hadamard, with perturbations represented via the exponential map, following the approach often referred to as the “velocity method.” This widely used method allows us to delve into the geometric intricacies of shape optimization within the Riemannian framework, providing a solid foundation for future research on shape optimization on curved manifolds.

The chapter is organized as follows: In [Section 2.1](#), we extend Hadamard’s boundary variation method to the general case of a Riemannian manifold  $M$ , using the exponential map  $\exp : TM \rightarrow M$  to relate the tangent bundle to points on the manifold. Next, in [Section 2.2](#), we discuss the use of signed distance functions to represent domains embedded in the Riemannian manifold, and we extend several classical shape optimization results to this more general setting. Finally, in [Section 2.3](#), we present general results related to the computation of shape derivatives and their forms (ie. Hadamard’s structure theorem).

### Chapter 3: Body-fitted tracking within a surface via a level set based mesh evolution method

This chapter introduces a numerical strategy for tracking the large-scale motions of a region  $G(t)$  on a three-dimensional surface  $S$ , driven by a complex velocity field  $V(t, x)$ . Extending previous work on evolving domains in Euclidean space  $\mathbb{R}^d$  [[12](#), [13](#), [14](#)], the approach is adapted to surface-based problems. Two complementary representations of the region  $G(t)$  are utilized: an exact mesh representation for precise geometric and finite element computations, and an implicit level set representation to handle large deformations and topological changes. Efficient numerical algorithms enable seamless transitions between these representations as required by the computational task.

The chapter proceeds by first outlining the problem of tracking region evolution on a surface in [Section 3.1](#), followed by a detailed explanation of the numerical strategy in [Section 3.2](#). This includes key components such as signed distance function calculations, level set evolution, and surface-specific remeshing operations. The methods are implemented in open-source software, facilitating their use in various applications. Finally, in [Sections 3.3](#) to [3.5](#), we present numerical applications that demonstrate the effectiveness of the framework, including the motion by the conormal field, and simulations of fire front propagation, showcasing the robustness and versatility of the proposed approach.

### Chapter 4: Shape and topology optimization of the regions supporting boundary conditions

This long chapter deals with the main motivation of thesis. It focuses on a specific class of shape and topology optimization problems, where the goal is to optimize a region  $G$  on the boundary  $\partial\Omega$  of a

---

domain  $\Omega$ , which supports particular boundary conditions in the formulation of a physical boundary value problem.

The chapter opens with [Section 4.1](#), where we present the mathematical framework for optimizing regions that support boundary conditions. These techniques are categorized into two main approaches: geometric optimization and topological optimization. Geometric optimization is covered in [Section 4.2](#), where we describe the instance of Hadamard’s method in the context of surfaces, drawing on the work of [\[108\]](#). Topological optimization, introduced in [Section 4.3](#), is a key contribution of this thesis. It introduces the concept of topological sensitivity for creating new regions with boundary conditions. This contribution builds on prior work [\[56\]](#), which explores the asymptotic expansion between the “background” potential and its perturbed version by a slight change in the definition of the boundary conditions in the conductivity equation. To provide the necessary foundation for this work, we revisit essential concepts related to Sobolev spaces and the single layer potential in [Sections 4.4](#) and [4.5](#).

In [Section 4.6](#), we establish a baseline analysis by examining the sensitivity of introducing a homogeneous Dirichlet boundary within a Neumann boundary in the context of the conductivity equation. This analysis is then extended to other physical models, including the Helmholtz equation in [Section 4.7](#) and linear elasticity in [Section 4.8](#). Building on these analyses, we propose a general method for shape and topology optimization of regions supporting boundary conditions in [Section 4.10](#), leveraging the evolution strategy outlined in [Chapter 3](#). The chapter prepares several real-world applications of these methods in [Sections 4.7](#), [4.11](#), [4.13](#) and [4.14](#).

## Chapter 5: Rodin: A numerical C++20 library for shape and topology optimization

In this chapter, we discuss the implementation and design of our C++ library, which we developed to implement the numerical algorithms used in this thesis. Here we explain the programming paradigms utilized in the design of the library which allow the easy implementation of various shape optimization methods, and showcase some of the examples we have obtained. We draw attention to the particular syntax of the library which permits the easy comprehension of the resulting programming code.

## Contributions

This research has given rise to two preprints. The first one based on Chapter 3 and the second one on Chapter 4.

- C. Brito-Pacheco, C. Dapogny. *Body-fitted tracking within a surface via a level set based mesh evolution method*, submitted (2023).
- E. Bonnetier, C. Brito-Pacheco, C. Dapogny. *Numerical shape and topology optimization of the regions supporting boundary conditions*, in preparation (2024).

It has been presented at the following occasions:

- CANUM 2024. Summer 2024. *Shape and topology optimization of the regions supporting boundary conditions*.
- WCSMO 2023. Summer 2023. *Shape and topology optimization of the regions supporting boundary conditions*.
- MFEM Community Workshop 2022. *Rodin: Lightweight and Modern C++17 Shape, Density and Topology Optimization Framework*.

# Chapter 1

## A review of shape and topology optimization

---

This chapter aims to offer a comprehensive review of “classical” shape and topology optimization methods, tracing their development, applications, and the theoretical advancements that have shaped the field. We explore the historical context and key developments that have propelled shape and topology optimization to the forefront of modern design practices. We review the mathematical formulations that underpin these methods, including the use of finite element analysis, sensitivity analysis, and optimization algorithms. The chapter highlights the differences and synergies between shape and topology optimization, providing insights into how they can be used in tandem to achieve superior design outcomes.

### 1.1 Historical context

The origins of shape optimization can be traced back to the calculus of variations, a branch of mathematical analysis that emerged in the 18th century, pioneered by mathematicians such as Leonhard Euler (15 April 1707 – 18 September 1783) and Joseph-Louis Lagrange (25 January 1736 – 10 April 1813). Their early work [138, 216] laid the groundwork for optimizing functionals, which are mathematical expressions involving integrals over a domain. One of the earliest and most significant problems connected to shape optimization is the isoperimetric problem, which seeks to determine the shape that maximizes the area enclosed by a given perimeter. This problem intrigued ancient Greek mathematicians, such as Zenodorus (circa 200 BC), who studied various polygonal figures and their area maximizing properties [180]. Zenodorus demonstrated that among polygons with the same perimeter, the regular polygon encloses the largest area, and that the circle encloses a larger area than any polygon with the same perimeter [206]. The study of the isoperimetric problem was so appealing, that it later became a fundamental question within the calculus of variations. Mathematically, the problem can be formulated as the following optimization problem:

$$\max_{\Omega \subset \mathbb{R}^2} \text{Area}(\Omega) \quad \text{s.t.} \quad \text{Per}(\Omega) = P, \quad (\mathcal{A})$$

for some given  $P > 0$ , with  $\text{Area}(\Omega) = \int_{\Omega} dx$  and  $\text{Per}(\Omega) = \int_{\partial\Omega} ds$ . The solution to  $(\mathcal{A})$  is well-known:

*The shape that maximizes the enclosed area for a given perimeter is a circle.*

This solution has by now been extensively studied, proved, and re-proved within various fields and through different techniques, many of which are closely related to shape optimization. For detailed discussions on its solution and the field of isoperimetric inequalities, we refer the reader to [259, 262], as well as the classical texts on the calculus of variations [161, 99].

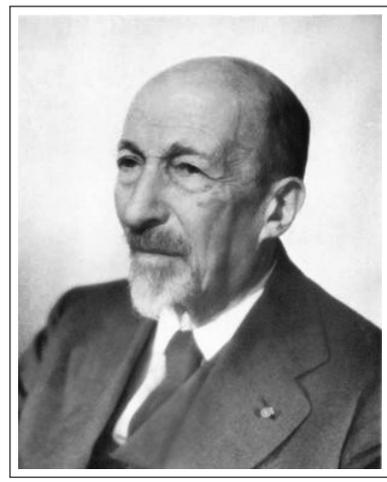
In the early 20th century, shape optimization began to mould as a distinct discipline, largely due to the contributions of Jacques Hadamard (8 December 1865 – 17 October 1963). Hadamard introduced



(a) Leonhard Euler



(b) Joseph-Louis Lagrange



(c) Jacques Hadamard

the concept of boundary variations (the subject of [Section 1.2](#)), which is nowadays a cornerstone in the field. His work provided essential insights into how small changes in the shape of a domain affect solutions to boundary value problems, a key consideration in shape optimization.

The latter half of the 20th century saw substantial advancements in shape optimization, driven by the development of numerical methods and computational tools. Pioneers such as John von Neumann and Stanislaw Ulam contributed to the development of computational techniques that could be applied to shape optimization. The emergence of modern computers enabled the optimization of increasingly complex shapes and domains, which in turn led to practical applications in various engineering and physical sciences.

The introduction of finite element methods [\[92\]](#) and other numerical techniques (e.g. finite differences) has also been pivotal in solving shape optimization problems in real-world scenarios by leveraging the power of discretizing continuous shapes that appear in the real world, and simulating the physics behind them.

In recent decades, shape optimization has further evolved with the integration of advanced techniques such as level-set methods [\[67, 325, 21\]](#), topology optimization [\[52, 330, 298\]](#), genetic algorithms [\[299\]](#), and most recently it has seen utilization with the virtual element method (see [\[49\]](#) for the basics) in the semi-discrete optimal transport setting [\[109\]](#). These modern approaches facilitate the optimization of highly intricate shapes and structures and allow for the formulation of general optimization problems of the form:

$$\min_{\Omega \subset \mathbb{R}} J(\Omega) \quad \text{s.t.} \quad C(\Omega) \leq 0, \quad (\mathcal{P})$$

of which [\(A\)](#) is a special case. Here  $J(\Omega)$  is the criterion to be minimized constrained by  $C(\Omega)$ .

Today, shape optimization remains a vibrant area of research, with significant progress being made in multi-disciplinary optimization, where shape optimization is integrated with other design variables, such as material properties [\[50\]](#). The integration of machine learning and artificial intelligence into shape optimization processes is further expanding the possibilities for automated and intelligent design (see for example [\[225, 335, 86\]](#)).

Overall, shape optimization has evolved from its classical origins in the calculus of variations to a modern, computationally driven field with extensive applications. The milestones in its development have solidified its status as an essential tool in engineering, physics, and beyond, with ongoing research continuing to push the limits of what can be achieved.

## 1.2 Hadamard's boundary variation method

Hadamard's boundary variation method, a cornerstone in the field of shape optimization, provides a systematic approach for analyzing how small perturbations in the boundary of a domain affect a given functional. This method, initially introduced by French mathematician Jacques Hadamard in his seminal article [\[177\]](#), is particularly significant in the study of boundary value problems, where the shape or configuration of the domain plays a crucial role in determining the behavior of the solution.

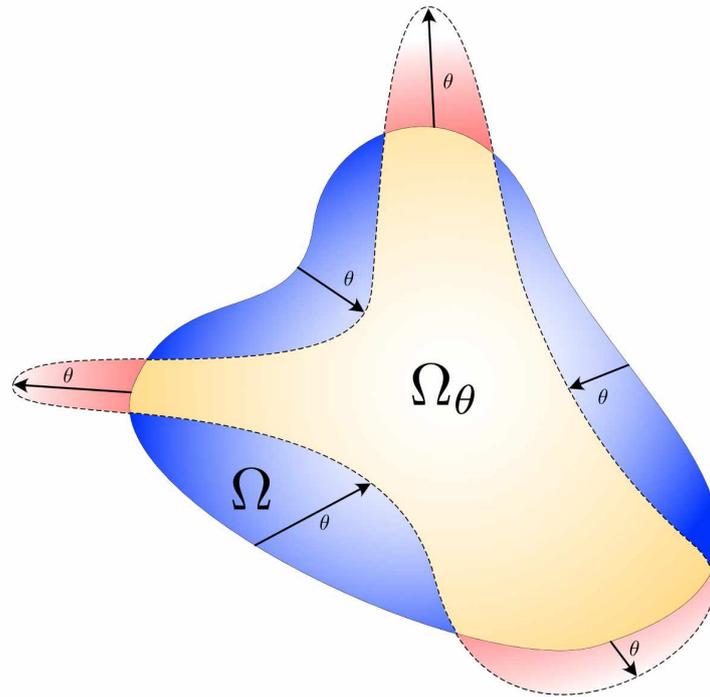


Figure 1.2: Domain  $\Omega$  undergoing boundary variations characterized by the deformation vector field  $\theta$ . The perturbed domain  $\Omega_\theta$  shows smooth modifications of the boundary.

At its core, Hadamard's method involves the perturbation of the boundary to induce a notion of shape derivative, which quantifies the sensitivity of a functional  $J(\Omega)$  to infinitesimal geometrical deformations of the domain. This derivative is instrumental in determining how slight modifications to the shape can improve or deteriorate the performance of a system, making it a useful tool to solve a problem like  $(\mathcal{P})$ . Numerous authors have revisited this topic, discussing it in standard texts on shape optimization such as [22, 185, 116, 183, 301], and the reader is referred to those texts for a comprehensive presentation. This section serves as the basis of Chapter 2 and the geometrical optimization section of Chapter 4.

### 1.2.1 Definitions and generalities

The method involves considering a Lipschitz domain, denoted as  $\Omega \subset \mathbb{R}^d$ , and examining the deformation induced by the mapping:

$$x \mapsto (\text{Id} + \theta)(x), \quad \theta \in W^{1,\infty}(\mathbb{R}^d, \mathbb{R}^d).$$

This deformation leads to a deformed domain, denoted as  $\Omega_\theta$ , which is the image of  $\Omega$  under this mapping:

$$\Omega_\theta := (\text{Id} + \theta)(\Omega) = \{x + \theta(x) : x \in \Omega\}.$$

A visual representation of this deformation is depicted in Fig. 1.2. Importantly, one may prove that for sufficiently "small" values of  $\theta$ , the mapping  $x \mapsto x + \theta(x)$  becomes a Lipschitz diffeomorphism. Consequently, any deformation  $\Omega_\theta$  retains the same topology as  $\Omega$ . More specifically, the following holds.

**Theorem 1.1.** *Let  $\theta \in W^{1,\infty}(\mathbb{R}^d, \mathbb{R}^d)$  be such that  $\|\theta\|_{W^{1,\infty}(\mathbb{R}^d, \mathbb{R}^d)} < 1$ . Then the mapping*

$$(\text{Id} + \theta) : \mathbb{R}^d \rightarrow \mathbb{R}^d$$

*is a Lipschitz diffeomorphism.*

*Idea of proof.* The proof relies on the fact that since the norm is strictly lower than 1, then its Neumann series  $\sum_{k=0}^{\infty} \theta^k$  is convergent, whence the invertibility [22, 140].

The concept of varying the boundary in shape optimization calls for the notion of differentiability with respect to the shape. This arises from the ability to shift the requirement for differentiability to the underlying mapping responsible for the deformation of the boundary.

**Definition 1.1 (Shape differentiability).** *The mapping  $\Omega \mapsto J(\Omega)$  is said to be **shape differentiable** at  $\Omega$  if the mapping*

$$\begin{aligned} W^{1,\infty}(\mathbb{R}^d; \mathbb{R}^d) &\rightarrow \mathbb{R} \\ \theta &\mapsto J(\Omega_\theta) = J((\text{Id} + \theta)(\Omega)) \end{aligned}$$

*is Fréchet differentiable at  $\theta = 0$ . On the other hand, there exists a continuous linear mapping  $L : W^{1,\infty}(\mathbb{R}^d; \mathbb{R}^d) \rightarrow \mathbb{R}$  such that*

$$J(\Omega_\theta) = J(\Omega) + L(\theta) + o(\theta), \quad \text{with} \quad \lim_{\theta \rightarrow 0} \frac{|o(\theta)|}{\|\theta\|_{W^{1,\infty}(\mathbb{R}^d; \mathbb{R}^d)}} = 0$$

*We call  $J'(\Omega) := L$ , the **shape derivative** of  $J$  at  $\Omega$ .*

Let us denote by  $n_{\partial\Omega}(x)$  the unique outward-pointing unit normal vector field along the boundary  $\partial\Omega$ . A significant theorem concerning the nature of shape derivatives states that, under certain smoothness conditions for both the considered functional and the domains involved, the shape derivatives predominantly depend on the normal trace component  $\theta \cdot n_{\partial\Omega}$  of the deformation  $\theta$  applied to the boundary  $\partial\Omega$ . This theorem, widely known as Hadamard's structure theorem, carries substantial importance in the field.

**Theorem 1.2 (Hadamard's structure theorem).** *Let  $\Omega \subset \mathbb{R}^d$  be a smooth bounded open subset and  $J(\Omega)$  be a shape differentiable functional. Let  $\theta_1, \theta_2 \in W^{1,\infty}(\mathbb{R}^d; \mathbb{R}^d)$  be such that  $\theta_2 - \theta_1 \in C^1(\mathbb{R}^d; \mathbb{R}^d)$  and  $\theta_1 \cdot n_{\partial\Omega} = \theta_2 \cdot n_{\partial\Omega}$  on  $\partial\Omega$ . Then,*

$$J'(\Omega)(\theta_1) = J'(\Omega)(\theta_2).$$

*Proof.* For the proof we refer to [184]. ■

One can extract more structure out of the shape derivative  $J'(\Omega)(\theta)$ . The following remark is of significant importance.

**Remark 1.1 (Structure of integral criterions).** *In many practical applications, the objective function  $J(\Omega)$  is often expressed as an integral involving the solution  $u_\Omega$  of a boundary value problem. In such cases, the shape derivative  $J'(\Omega)(\theta)$  can be expressed more precisely as:*

$$J'(\Omega)(\theta) = \int_{\partial\Omega} f_\Omega \theta \cdot n_{\partial\Omega} \, ds, \tag{1.1}$$

*where  $f_\Omega : \partial\Omega \rightarrow \mathbb{R}$  is a scalar field dependent on  $\Omega$ . Already, this formulation simplifies the process of finding a descent direction, which can be achieved by selecting  $\theta = -f_\Omega n_{\partial\Omega}$ , so that:*

$$J'(\Omega)(\theta) = - \int_{\partial\Omega} f_\Omega^2 \, ds < 0.$$

*The fact that this quantity is negative proves that it is indeed a descent direction. Clearly, utilizing [Definition 1.1](#), we have:*

$$J(\Omega_{t\theta}) = J(\Omega) - tJ'(\Omega)(\theta) + o(t) < J(\Omega),$$

*for small  $t > 0$ .*

## 1.2.2 Techniques and approaches to compute shape derivatives

The logical next step in investigating shape derivatives is to compute them. In this section, we provide a brief review and presentation of shape derivatives, focusing on the methods used for their calculation. By exploring these fundamental concepts, we lay the groundwork for the more in-depth discussions that will follow in this thesis, setting the stage for a deeper exploration of shape optimization.

**Theorem 1.3.** *Let  $f \in W^{1,1}(\mathbb{R}^d)$  and define the shape functional  $J$  by*

$$J(\Omega) := \int_{\Omega} f(x) \, dx.$$

*Then  $J$  is shape differentiable at any  $\Omega$  and*

$$J'(\Omega)(\theta) = \int_{\Omega} \nabla \cdot (f\theta) \, dx, \quad (1.2)$$

*for any  $\theta \in W^{1,\infty}(\mathbb{R}^d; \mathbb{R}^d)$ . Furthermore, we have:*

$$J'(\Omega)(\theta) = \int_{\partial\Omega} \theta \cdot n_{\partial\Omega} \, ds. \quad (1.3)$$

*Sketch of proof.* We must establish the Fréchet differentiability of  $\theta \mapsto J(\Omega_{\theta})$  at  $\theta = 0$ . To this end, we perform a change of variables to obtain:

$$J(\Omega_{\theta}) = \int_{\Omega} f \circ (\text{Id} + \theta) |\det(I + \nabla\theta)| \, dx.$$

To see that the integrand is differentiable at  $\theta = 0$  we must compute the differentials of the mappings  $f \circ (\text{Id} + \theta)$  and  $\theta \mapsto \det(I + \nabla\theta)$ . For the proof of the former we refer to [185] in which case the differential is given by:

$$\theta \mapsto \nabla f \cdot \theta.$$

For the latter we prove differentiability of the mapping  $A \in \mathcal{GL}(\mathbb{R}) \mapsto \det(A) \in \mathbb{R}$  at  $I$ , in which case the differential is given by:

$$A \mapsto \text{tr}(A),$$

which instantly yields the formula for the differential of the original mapping:

$$\theta \mapsto \nabla \cdot \theta.$$

Applying the product rule yields (1.2). Moreover if  $\Omega$  is Lipschitz then an integration by parts yields (1.3). ■

A related result is the one dealing with the shape derivative of an integral over the boundary  $\partial\Omega$ . The proof follows a similar line of reasoning as that of the previous theorem, involving a change of variables and the computation of the differential of the integrand. We omit the proof in this context but refer interested readers to [185] for a comprehensive proof.

**Theorem 1.4.** *Let  $g \in W^{1,2}(\mathbb{R}^d)$  and define the shape functional*

$$J(\Omega) := \int_{\partial\Omega} g \, ds.$$

*Then,  $J$  is differentiable at any (bounded domain)  $\Omega$  of class  $C^2$  and its shape differential (restricted to smooth deformations  $W^{1,\infty}(\mathbb{R}^d; \mathbb{R}^d)$ ) reads:*

$$J'(\Omega)(\theta) = \int_{\partial\Omega} \left( \frac{\partial g}{\partial n_{\partial\Omega}} + \kappa g \right) \theta \cdot n_{\partial\Omega} \, ds,$$

*where  $\kappa = \nabla \cdot (n_{\partial\Omega})$  denotes the mean curvature of  $\partial\Omega$ .*

The subsequent corollaries present the application of the two preceding theorems to two widely studied quantities of interest: the volume and perimeter of a shape  $\Omega$ .

**Corollary 1.1 (Derivative of the volume).** *Let  $\Omega$  be Lipschitz. Define the **volume of a shape  $\Omega$**  by*

$$\text{Vol}(\Omega) := \int_{\Omega} dx.$$

*Then, the volume is shape differentiable at  $\Omega$  and its shape derivative is given by:*

$$\text{Vol}'(\Omega)(\theta) = \int_{\Omega} \nabla \cdot \theta \, dx = \int_{\partial\Omega} \theta \cdot n_{\partial\Omega} \, ds.$$

**Corollary 1.2 (Derivative of the perimeter).** *Let  $\Omega$  be of class  $C^2$ . Define the **perimeter of a domain  $\Omega$**  by*

$$\text{Per}(\Omega) := \int_{\partial\Omega} ds.$$

*Then, the perimeter is shape differentiable at  $\Omega$  and its shape derivative is given by:*

$$\text{Per}'(\Omega)(\theta) = \int_{\partial\Omega} \kappa \theta \cdot n_{\partial\Omega} \, ds$$

These corollaries for the shape derivatives of volume and perimeter are widely utilized in practical applications to impose constraints or apply penalties associated with these geometric properties. For instance, consider a multi-criteria functional:

$$J(\Omega) := \int_{\Omega} j(u_{\Omega}) \, dx + \ell \text{Vol}(\Omega),$$

where the goal is to find the optimal shape  $\Omega$  that minimizes a quantity dependent on the solution  $u_{\Omega}$  to a boundary value problem, while simultaneously minimizing the volume of the shape. In this context, the parameter  $\ell > 0$  serves as a penalization factor.

### 1.2.3 Shape derivatives of criteria depending on the solution to a boundary value problem

In the majority of shape and topology optimization applications, our focus lies in functionals that rely on the solution to a boundary value problem, which is intrinsically linked to the desired shape. This problem, commonly known as the “state equation”, captures and describes the behavior of the underlying physical system at play. To set ideas, we can consider the functional:

$$\min_{\Omega \subset \mathbb{R}^d} J(\Omega) = \int_{\Omega} j(u_{\Omega}) \, dx,$$

where  $j \in C^2(\mathbb{R})$  and satisfies the growth conditions

$$\forall u \in \mathbb{R}, |j(u)| \leq C(1 + |u|^2), |j'(u)| \leq C(1 + |u|), |j''(u)| \leq C, \quad (1.4)$$

and where  $u_{\Omega} \in H_0^1(\Omega)$  is the solution to the Laplace equation (1.5) with homogeneous Dirichlet boundary conditions on the whole boundary  $\partial\Omega$ .

$$\begin{cases} -\Delta u_{\Omega} = f & \text{in } \Omega \\ u_{\Omega} = 0 & \text{on } \partial\Omega. \end{cases} \quad (1.5)$$

The computation of the shape derivative  $J'(\Omega)(\theta)$ , relies on the calculation of the “derivative” of the map  $\Omega \mapsto u_{\Omega}$ , which requires a precise definition. To address this, we introduce the following concepts.

**Definition 1.2 (Transport mapping).** *The **transported mapping** of  $u_{\Omega}$  is defined by:*

$$\begin{aligned} W^{1,\infty}(\mathbb{R}^d, \mathbb{R}^d) &\rightarrow H_0^1(\Omega) \\ \theta &\mapsto \overline{u_{\Omega}}(\theta) := u_{\Omega_{\theta}} \circ (\text{Id} + \theta) \end{aligned}$$

With this in mind, we can consider the Fréchet derivative of this mapping and subsequently define the notion of Lagrangian derivative.

**Definition 1.3 (Lagrangian derivative).** *Suppose that  $\theta \mapsto \overline{u_\Omega}(\theta)$  is Fréchet differentiable at  $\theta = 0$ . We then say that the map  $\Omega \mapsto u_\Omega$  has a **Lagrangian derivative** at  $\Omega$ , and we denote its Fréchet differential by  $\dot{u}_\Omega(\theta)$ .*

In contrast to the Lagrangian derivative, we can also consider the ‘‘Eulerian’’ derivative. For any fixed point  $x \in \Omega$ , we examine the derivative  $u'_\Omega(\theta)(x)$  of the mapping:

$$\theta \longmapsto u_{\Omega_\theta}(x),$$

as a mapping from  $W^{1,\infty}(\mathbb{R}^d, \mathbb{R}^d)$  to  $H_0^1(\Omega)$ . For fixed  $x$ , this derivative is well-defined within a neighborhood of 0 in  $W^{1,\infty}(\mathbb{R}^d, \mathbb{R}^d)$  mapping into  $\mathbb{R}$  because, for sufficiently small  $\theta \in W^{1,\infty}(\mathbb{R}^d, \mathbb{R}^d)$ , the point  $x$  remains within the deformed shape  $\Omega_\theta$ , where  $u_{\Omega_\theta}$  is evaluated. However, for a boundary point  $x \in \partial\Omega$ , this variation can be problematic if the vector field  $\theta(x)$  points inward, causing the point  $x$  to no longer belong to  $\Omega_\theta$  or its boundary.

Unlike the Lagrangian derivative, which is defined not just pointwise but also as a mapping from  $W^{1,\infty}(\mathbb{R}^d, \mathbb{R}^d)$  into  $H_0^1(\Omega)$ , allowing us to consider the Fréchet derivative of  $\theta \mapsto \overline{u_\Omega}(\theta)$ , the Eulerian approach provides a different perspective. If both the Eulerian and Lagrangian derivatives exist, the chain rule implies that for any point  $x \in \Omega$ ,

$$\dot{u}_\Omega(\theta)(x) = \left. \frac{d}{d\theta} (u_{\Omega_\theta}(x + \theta(x))) \right|_{\theta=0} = u'_\Omega(\theta)(x) + \nabla u_\Omega(x) \cdot \theta(x).$$

Thus, the Eulerian derivative can be expressed in terms of the Lagrangian derivative, leading to the following definition.

**Definition 1.4 (Eulerian derivative).** *The mapping  $\Omega \mapsto u_\Omega$  has an **Eulerian derivative**  $u'_\Omega(\theta)$  at  $\Omega$  if it has a Lagrangian derivative and if, in addition,  $\nabla u_\Omega \in H^1(\Omega)$ . The Eulerian derivative  $u'_\Omega(\theta) \in H^1(\Omega)$  is then defined as:*

$$u'_\Omega(\theta) = \dot{u}_\Omega(\theta) - \nabla u_\Omega \cdot \theta.$$

We shall not use the Eulerian derivative here, instead preferring the Lagrangian approach.

**Proposition 1.1.** *Let  $\Omega$  be a bounded Lipschitz domain and suppose that  $u_\Omega$  has a Lagrangian derivative  $\dot{u}_\Omega(\theta) \in H^1(\mathbb{R})$ . Then,  $J(\Omega) := \int_\Omega j(u_\Omega) dx$  is shape differentiable at  $\Omega$  and*

$$J'(\Omega)(\theta) = \int_\Omega (\nabla \cdot \theta) j(u_\Omega) + j'(u_\Omega) \dot{u}_\Omega(\theta) dx. \quad (1.6)$$

The proof of this proposition consists in a simple application of the product and chain rule for Fréchet derivatives. In this manner, we have reduced our problem of computing the shape derivative of the functional  $J(\Omega)$ , to that of computing the Lagrangian derivative, which is where the actual complexity lies. This computation is fairly standard (see for example [15]), and one can prove the differentiability of the mapping  $\theta \mapsto \overline{u_\Omega}(\theta)$  via an application of the implicit function theorem ([Theorem B.3](#)).

**Theorem 1.5.** *The Lagrangian derivative  $\dot{u}_\Omega(\theta)$  of the solution  $u_\Omega$  is characterized by the weak formulation:*

$$\forall v \in H_0^1(\Omega_\theta), \quad \int_\Omega \nabla \dot{u}_\Omega(\theta) \cdot \nabla v dx + \int_\Omega ((\nabla \cdot \theta) \mathbf{I} - \nabla \theta - \nabla \theta^t) \nabla u_\Omega \cdot \nabla v dx = \int_\Omega v \nabla \cdot (f \theta) dx. \quad (1.7)$$

*Proof.* The differentiability of the transported mapping  $\theta \mapsto \overline{u_\Omega}(\theta)$  is admitted and we simply show the calculation leading to the variational formulation (1.7). With this in mind we simply proceed with the computation. We wish to characterize  $\overline{u_\Omega}(\theta)$  by its weak formulation. To this end, we consider the variational formulation for  $u_{\Omega_\theta}$ :

$$\forall v \in H_0^1(\Omega), \quad \int_\Omega \nabla u_{\Omega_\theta} \cdot \nabla v dx = \int_\Omega f v dx$$

Performing a change of variables yields

$$\forall v \in H_0^1(\Omega), \quad \int_\Omega |\det(\mathbf{I} + \nabla \theta)| (\nabla u_{\Omega_\theta} \cdot \nabla v) \circ (\text{Id} + \theta) dx = \int_\Omega |\det(\mathbf{I} + \nabla \theta)| (f v) \circ (\text{Id} + \theta) dx$$

Now note that,

$$\forall v \in H_0^1(\Omega), \quad (\nabla v) \circ (\text{Id} + \theta) = (I + \nabla\theta)^{-1} \nabla (v \circ (\text{Id} + \theta))$$

Thus, after choosing  $v = w \circ (\text{Id} + \theta)^{-1}$  as a test function for  $w \in H_0^1(\Omega)$ , we may simplify the expression to

$$\forall w \in H_0^1(\Omega), \quad \int_{\Omega} A(\theta) \nabla \overline{u_{\Omega}}(\theta) \cdot \nabla w \, dx = \int_{\Omega} |\det(I + \nabla\theta)| f \circ (\text{Id} + \theta) w \, dx$$

where  $A(\theta) := |\det(I + \nabla\theta)| (I + \nabla\theta)^{-1} (I + \nabla\theta)^{-T}$ . Hence we have characterized  $\overline{u_{\Omega}}(\theta)$  by a weak formulation. Thus, we may now compute the Fréchet differential at  $\theta = 0$  utilizing the product rule, and apply the definition of  $\dot{u}_{\Omega}(\theta)$  to obtain the result. ■

In order to remove the implicit dependence on the Lagrangian derivative.

**Theorem 1.6 (Volume form).** *The derivative  $J'(\Omega)(\theta)$  rewrites:*

$$J'(\Omega)(\theta) = \int_{\Omega} (\nabla \cdot \theta) j'(u_{\Omega}) - \int_{\Omega} \nabla \cdot (f\theta) p_{\Omega} \, dx + \int_{\Omega} ((\nabla \cdot \theta)I - \nabla\theta - \nabla\theta^t) \nabla u_{\Omega} \cdot \nabla p_{\Omega} \, dx, \quad (1.8)$$

where  $p_{\Omega} \in H_0^1(\Omega)$  is solution the boundary value problem:

$$\begin{cases} -\Delta p_{\Omega} = -j'(u_{\Omega}) & \text{in } \Omega \\ p_{\Omega} = 0 & \text{on } \partial\Omega. \end{cases} \quad (1.9)$$

*Proof.* Note that by the Lax-Milgram theory, the weak formulation for  $p_{\Omega} \in H_0^1(\Omega)$  solution to the adjoint equation (1.9):

$$\forall v \in H_0^1(\Omega), \quad \int_{\Omega} \nabla p_{\Omega} \cdot \nabla v \, dx = - \int_{\Omega} j'(u_{\Omega}) v \, dx,$$

is well-posed. Thus, choosing  $v = \dot{u}_{\Omega}(\theta)$ , we get:

$$\int_{\Omega} \nabla p_{\Omega} \cdot \nabla \dot{u}_{\Omega}(\theta) \, dx = - \int_{\Omega} j'(u_{\Omega}) \dot{u}_{\Omega}(\theta) \, dx.$$

From another side, recall the weak formulation for  $\dot{u}_{\Omega}(\theta)$  and choose  $p_{\Omega}$  as a test function, so that in particular:

$$\int_{\Omega} \nabla \dot{u}_{\Omega}(\theta) \cdot \nabla p_{\Omega} \, dx = \int_{\Omega} \nabla \cdot (f\theta) p_{\Omega} \, dx - \int_{\Omega} ((\nabla \cdot \theta)I - \nabla\theta - \nabla\theta^t) \nabla u_{\Omega} \cdot \nabla p_{\Omega} \, dx.$$

Finally, we may recognize terms and substitute in (1.6) to obtain the result. ■

Formula (1.8) is commonly known as the ‘‘volumetric expression’’ for the shape derivative because it does not follow the structure of (1.1), since it involves integrals over the entire domain  $\Omega$  rather than just its boundary. While it is possible to extract descent directions from this expression, the process is somewhat complex; see Section 1.3 for further details. By assuming higher regularity for  $u_{\Omega}$  and  $p_{\Omega}$ , this task can be simplified by transforming the expression into a surface integral that aligns with the structure of (1.1). This transformation, however, demands greater regularity for  $u_{\Omega}$  and  $p_{\Omega}$ , typically ensured by elliptic regularity theory. In this case, we can integrate by parts towards the

**Proposition 1.2 (Surface form).** *Assume that  $u_{\Omega}$  and  $p_{\Omega}$  are in  $H^2(\Omega)$ . Then,*

$$\forall \theta \in W^{1,\infty}(\mathbb{R}^d, \mathbb{R}^d), \quad J'(\Omega)(\theta) = \int_{\partial\Omega} j(u_{\Omega}) \theta \cdot n_{\partial\Omega} \, ds - \int_{\partial\Omega} \frac{\partial u_{\Omega}}{\partial n_{\partial\Omega}} \frac{\partial p_{\Omega}}{\partial n_{\partial\Omega}} \theta \cdot n_{\partial\Omega} \, ds - \int_{\partial\Omega} f p_{\Omega} \theta \cdot n_{\partial\Omega} \, ds. \quad (1.10)$$

*Proof.* We can integrate by parts, via Green’s identities (Proposition C.1), to obtain:

$$\begin{aligned}
 J'(\Omega)(\theta) &= \int_{\partial\Omega} j(u_\Omega)\theta \cdot n_{\partial\Omega} \, ds - \int_{\Omega} j'(u_\Omega)\nabla u_\Omega \cdot \theta \, dx \\
 &\quad + \int_{\partial\Omega} \left( (\nabla u_\Omega \cdot \nabla p_\Omega)\theta \cdot n_{\partial\Omega} - (\nabla u_\Omega \cdot \theta) \frac{\partial p_\Omega}{\partial n_{\partial\Omega}} - (\nabla p_\Omega \cdot \theta) \frac{\partial u_\Omega}{\partial n_{\partial\Omega}} \right) ds \\
 &\quad - \int_{\Omega} (\nabla(\nabla u_\Omega \cdot \nabla p_\Omega) \cdot \theta - \nabla^2 p_\Omega \nabla u_\Omega \cdot \theta - \nabla^2 u_\Omega \nabla p_\Omega \cdot \theta) \, dx \\
 &\quad + \int_{\Omega} (\Delta u_\Omega(\theta \cdot \nabla p_\Omega) + \Delta p_\Omega(\theta \cdot \nabla u_\Omega)) \, dx \\
 &\quad - \int_{\partial\Omega} f p_\Omega \theta \cdot n_{\partial\Omega} \, ds + \int_{\Omega} f \nabla p_\Omega \cdot \theta \, dx.
 \end{aligned} \tag{4.34}$$

Recalling the interior conditions for  $u_\Omega$  and  $p_\Omega$ :

$$-\Delta u_\Omega = f, \quad -\Delta p_\Omega = -j'(u_\Omega),$$

and the identity:

$$\nabla(\nabla u_\Omega \cdot \nabla p_\Omega) = \nabla^2 u_\Omega \nabla p_\Omega + \nabla^2 p_\Omega \nabla u_\Omega,$$

we can verify that all the integrals supported on  $\Omega$  cancel out:

$$\begin{aligned}
 & - \int_{\Omega} j'(u_\Omega)\nabla u_\Omega \cdot \theta \, dx - \int_{\Omega} (\nabla(\nabla u_\Omega \cdot \nabla p_\Omega) \cdot \theta - \nabla^2 p_\Omega \nabla u_\Omega \cdot \theta - \nabla^2 u_\Omega \nabla p_\Omega \cdot \theta) \, dx \\
 & + \int_{\Omega} (\Delta u_\Omega(\theta \cdot \nabla p_\Omega) + \Delta p_\Omega(\theta \cdot \nabla u_\Omega)) \, dx + \int_{\Omega} f \nabla p_\Omega \cdot \theta \, dx = 0.
 \end{aligned}$$

Additionally, since  $u_\Omega$  and  $p_\Omega$  vanish on  $\partial\Omega$ , along with their tangential derivatives, we have the following identity:

$$\int_{\partial\Omega} \left( (\nabla u_\Omega \cdot \nabla p_\Omega)\theta \cdot n_{\partial\Omega} - (\nabla u_\Omega \cdot \theta) \frac{\partial p_\Omega}{\partial n_{\partial\Omega}} - (\nabla p_\Omega \cdot \theta) \frac{\partial u_\Omega}{\partial n_{\partial\Omega}} \right) ds = - \int_{\partial\Omega} \frac{\partial u_\Omega}{\partial n_{\partial\Omega}} \frac{\partial p_\Omega}{\partial n_{\partial\Omega}} \theta \cdot n_{\partial\Omega} \, ds,$$

so that we end up with the desired expression (1.10). ■

The main theoretical distinction between the surfacic and volumetric forms of the shape derivative  $J'(\Omega)(\theta)$  lies in their regularity requirements. The choice of which form is more practical in applications is more nuanced. The surfacic form offers the advantage of providing a descent direction immediately, facilitating their implementation. Specifically, this form only needs to be evaluated at boundary points, which are generally fewer in number than the interior points of the domain. On the other hand, the volume form is typically more stable.

### 1.3 The Hilbertian extension-regularization procedure

This section outlines an effective and adaptable framework for deriving optimal descent directions, based on the shape derivative  $J'(\Omega)(\theta)$ . While widely recognized in shape optimization, the method's origins are somewhat obscure. It is explored in sources such as [67, 114, 267], particularly with regard to shape optimization and inverse problems. In particular we refer the reader to §5.2 of [15] which presents the procedure in straightforward manner, and is what most of this section is based on.

The approach centers on utilizing the inner product to compute a gradient  $\theta$  from the differential  $J'(\Omega)(\theta)$ . As discussed in Section 1.2.1, shape derivatives for most of the functionals considered in this chapter take the form

$$\forall \theta \in W^{1,\infty}(\mathbb{R}^d, \mathbb{R}^d), \quad J'(\Omega)(\theta) = \int_{\partial\Omega} f_\Omega \theta \cdot n \, ds = (f_\Omega n_{\partial\Omega}, \theta)_{L^2(\partial\Omega)^d},$$

where  $f_\Omega : \partial\Omega \rightarrow \mathbb{R}$  is a scalar field dependent on the solution  $u_\Omega$  to the state equation and the adjoint state  $p_\Omega$ . Here,  $(\cdot, \cdot)_{L^2(\partial\Omega)^d}$  denotes the usual inner product on  $L^2(\partial\Omega)^d$ . It is natural to propose the descent direction:

$$\theta = -f_\Omega n_{\partial\Omega} \quad \text{on } \partial\Omega. \tag{1.11}$$

In this case, for a small enough step  $\tau > 0$ , the descent of  $J(\Omega)$  is expressed as:

$$J(\Omega_{\tau\theta}) = J(\Omega) - \tau \int_{\partial\Omega} f_{\Omega}^2 ds < J(\Omega),$$

unless  $f_{\Omega}$  vanishes on  $\partial\Omega$ , indicating that  $\Omega$  is stationary with respect to  $J(\Omega)$ . Ignoring for a moment that  $W^{1,\infty}(\mathbb{R}^d, \mathbb{R}^d)$  is not a subset of  $L^2(\partial\Omega)^d$ , the form in (1.11) corresponds to the gradient of the Fréchet derivative  $\theta \mapsto J'(\Omega)(\theta)$ , with respect to the  $L^2(\partial\Omega)^d$  inner product. This choice is generally unsuitable for two reasons:

1. The deformation  $\theta$  in (1.11) may be irregular, causing numerical artifacts in the computations of  $u_{\Omega}$  or  $p_{\Omega}$ , and potentially introducing unwanted oscillations in  $\Omega_{\tau\theta}$ . A smoother deformation field is preferable.
2. The definition in (1.11) applies only on the boundary  $\partial\Omega$ . Despite the fact that  $f_{\Omega}$  (and  $n_{\partial\Omega}$ ) can often be extended to  $\mathbb{R}^d$ , such extensions may be awkward, requiring a more appropriate extension based on the objective.

Based on these considerations, the idea of velocity extension and regularization of the  $L^2(\partial\Omega)^d$  gradient of  $J(\Omega)$  allows to overcome both issues. Let  $H \subset W^{1,\infty}(\mathbb{R}^d, \mathbb{R}^d)$  be a Hilbert space, with inner product  $a(\cdot, \cdot)$ . We find  $g_{\Omega} \in H$  such that:

$$\forall w \in H, a(g_{\Omega}, w) = J'(\Omega)(w).$$

Obviously,  $-g_{\Omega}$  is again a descent direction for  $J(\Omega)$  if  $g_{\Omega} \neq 0$ , since for  $\tau > 0$  small enough, the definition of shape derivative yields:

$$\begin{aligned} J(\Omega_{-\tau g_{\Omega}}) &= J(\Omega) - \tau J'(\Omega)(g_{\Omega}) + o(\tau) \\ &= J(\Omega) - \tau a(g_{\Omega}, g_{\Omega}) + o(\tau) \\ &< J(\Omega). \end{aligned}$$

Moreover,  $g_{\Omega}$  is a “better” descent direction than  $-f_{\Omega}n$ , insofar as:

- The Hilbert space  $H$  is made of “regular” vector fields (they are at least in  $W^{1,\infty}(\mathbb{R}^d, \mathbb{R}^d)$ ); therefore, the gradient  $g_{\Omega} \in H$  is naturally “regular”.
- Since  $H$  contains vector fields defined on the whole ambient space  $\mathbb{R}^d$  (or, in practice, on the whole computational domain  $D$ ), then  $g_{\Omega}$  is naturally defined on  $\mathbb{R}^d$  (respectively, on  $D$ ).

The perhaps most natural choice is  $H = H^m(\mathbb{R}^d)^d$ , with the standard inner product:

$$a(u, v) = \sum_{|\alpha| \leq m} \int_{\mathbb{R}^d} \frac{\partial^{\alpha} u}{\partial x^{\alpha}} \cdot \frac{\partial^{\alpha} v}{\partial x^{\alpha}} dx. \quad (1.12)$$

The Sobolev embeddings indeed guarantee that, if  $m > \frac{d}{2} + 1$ ,  $H$  is continuously embedded in  $W^{1,\infty}(\mathbb{R}^d, \mathbb{R}^d)$  (see e.g. [139]). Functions in  $H$  are “regular” insofar as their derivatives up to order  $m$  are square integrable functions. Other choices are possible and we refer to [15] for further reading.

**Remark 1.2.** *A common choice in practice is  $H = H^1(\mathbb{R}^d)^d$ , even though it is not a subspace of  $W^{1,\infty}(\mathbb{R}^d, \mathbb{R}^d)$ . While this approach is practical and easy to implement, it may lack full mathematical rigor. Additionally, an alternative to the inner product defined in (1.12) is:*

$$a(u, v) = \alpha^2 \int_{\mathbb{R}^d} \nabla u : \nabla v dx + \int_{\mathbb{R}^d} u \cdot v dx,$$

*where the constant  $\alpha$  reflects a regularization length-scale, roughly indicating the extent to which the gradient is smoothed near the boundary  $\partial\Omega$ .*

An important aspect of this procedure is worth emphasizing. Although the surface shape derivative  $J'(\Omega)(\theta)$  can be expressed in a volume form (see (1.8)), this form, while explicit in terms of  $\theta$ , does not naturally provide a descent direction. By using this framework and choosing a Hilbert space like  $H = H^1(\mathbb{R}^d)^d$  with the inner product  $a(\cdot, \cdot)$ , we can derive a descent direction from this complex expression. Specifically, solving

$$\forall v \in H, a(g_{\Omega}, v) = J'(\Omega)(v)$$

yields a vector field  $-g_{\Omega}$  that acts as a descent direction for  $J(\Omega)$ . This method, based on the distributed form of the shape derivative, is highly practical and often more effective than the surface form, which can involve challenging numerical computations, such as high-order derivatives or surface curvatures.

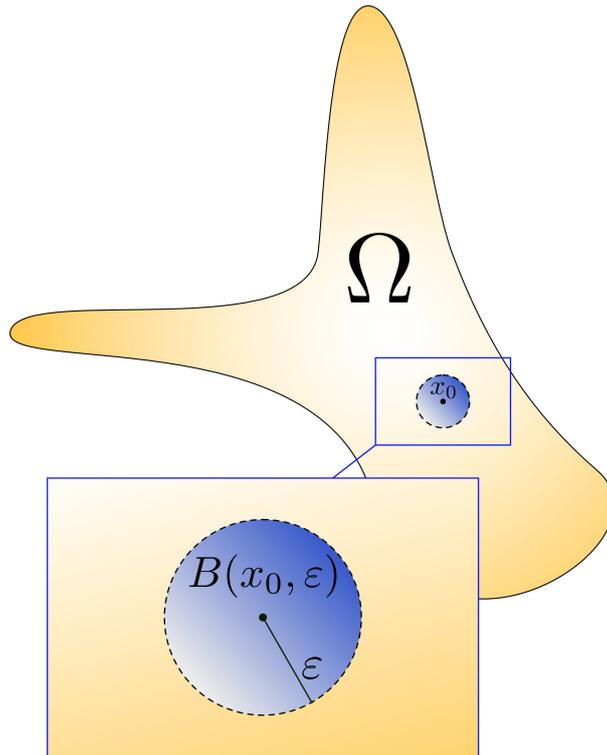


Figure 1.3: Domain  $\Omega$  with a small hole nucleated at point  $x$ , represented by the ball  $B(x, \varepsilon)$ . This illustrates a topological perturbation used to compute the topological derivative by examining the effect of the localized change.

## 1.4 Topological derivatives

This section provides a concise overview of topological derivatives, serving as a foundation for the work in [Chapter 4](#), where the topological derivative is extended to the surfacic case of optimizing regions bearing boundary conditions. The study of topological derivatives is essential for addressing complex problems where traditional shape optimization techniques are insufficient. Unlike shape derivatives, which are limited to smooth boundary deformations, topological derivatives enable more drastic changes in the topology of the domain, such as merging, splitting, or removing parts. This flexibility is particularly valuable in problems involving multiple phases or materials, where the optimal solution may require significant structural modifications. Particular examples to where the topological derivative has been applied include [\[159\]](#), where they optimize an electric machine to maximize its torque so it can deliver more mechanical power with less electrical input, enhancing its overall energy efficiency and operational stability. Another example, this time in structural optimization, optimizes self-loading structures (systems that experience forces due to their own weight) [\[253\]](#). This type of loading is crucial in the design and analysis of buildings, bridges, towers, and other large-scale constructions, where the structure's weight can contribute significantly to the overall stresses and deformations. A particular variation of the topological derivative, where one usually considers the nucleation of an open ball in the domain, is the work [\[100\]](#), where the author consider the addition of a thin tubular ligament (e.g. a material bar in a truss structure) between two distant regions of the boundary of the considered domain, usually to reinforce an already existing structure. Most of these applications usually couple the topological derivatives with the level set method (see for instance [\[68, 20, 340, 17\]](#)), which will be briefly explained in [Section 1.5](#).

The concept of topological derivative is based on variations of  $\Omega \subset \mathbb{R}^d$  ( $d = 2, 3$ ) of the form

$$\Omega_{x_0, \varepsilon} := \Omega \setminus \overline{B(x_0, \varepsilon)}$$

where  $B(x_0, \varepsilon)$  is the open ball with center  $x_0$  and radius  $\varepsilon$ . In other terms,  $\Omega_{x_0, \varepsilon}$  is obtained from  $\Omega$  by nucleation of a hole centered at  $x_0 \in \Omega$  with small radius  $\varepsilon > 0$ ; see [Fig. 1.3](#). The topological derivative  $d_T J(\Omega)(\varepsilon_0)$  of  $J$  at  $\Omega$  is the first non-trivial term in the asymptotic expansion of  $J(\Omega_{x_0, \varepsilon})$  as  $\varepsilon \rightarrow 0$ ; typically:

$$J(\Omega_{x_0, \varepsilon}) = J(\Omega) + \rho(\varepsilon) d_T J(\Omega)(x_0) + o(\rho(\varepsilon)).$$

for some positive function  $\rho : \mathbb{R}^+ \rightarrow \mathbb{R}^+$ . For criterion dependent on the solution  $u_{\Omega, \varepsilon}$  of a boundary value problem, we use the asymptotic expansion of the perturbed solution  $u_\varepsilon := u_{\Omega_{x_0, \varepsilon}}$ . Since this process is intricate and technical, we will just present the example in §1.2.2 of [254] which is very simple. We consider the criterion for shapes  $\Omega \subset \mathbb{R}^2$ :

$$J(\Omega) = \int_{\Omega} j(u_{\Omega}) \, dx$$

where  $u_{\Omega}$  is the solution to the boundary value problem (1.5) and  $j$  satisfies the conditions (1.4). We seek to introduce a topological perturbation of the source term of the form  $f_{x_0, \varepsilon} = \chi_{\Omega} - (1 - \eta)\chi_{B(x_0, \varepsilon)}$  around the fixed point  $x_0 \in \Omega$ , for  $\varepsilon > 0$  small enough that the ball  $B(x_0, \varepsilon)$  is compactly contain in  $\Omega$ . Here  $\chi_A$  represents the indicator function of the set  $A$ . In this case, we have:

$$f_{x_0, \varepsilon}(x) = \begin{cases} f(x) & \text{if } x \in \Omega \setminus \overline{B(x_0, \varepsilon)}, \\ \eta f(x) & \text{if } x \in B(x_0, \varepsilon), \end{cases}$$

with  $\eta > 0$  a constant that represents the contrast on the source term. Furthermore, a simplifying assumption that is typically used when conducting these types of analyses is that the source  $f(x)$  is constant in this neighborhood of  $x_0$ . With these considerations, we can write  $u_\varepsilon$  as the solution to the perturbed problem as:

$$J(\Omega_{x_0, \varepsilon}) = \int_{\Omega \setminus \overline{B(x_0, \varepsilon)}} j(u_\varepsilon) \, dx,$$

where  $u_{\Omega, \varepsilon}$  is solution in  $H_0^1(\Omega)$  such that:

$$\forall v \in H_0^1(\Omega), \quad \int_{\Omega \setminus \overline{B(x_0, \varepsilon)}} \nabla u_\varepsilon \cdot \nabla v \, dx = \int_{\Omega \setminus \overline{B(x_0, \varepsilon)}} f_{x_0, \varepsilon} v \, dx.$$

With these equations in mind, it is possible to prove that the perturbed criterion has the following asymptotic expansion in terms of  $\varepsilon$ ,

$$J(\Omega_{x_0, \varepsilon}) = J(\Omega) + \pi \varepsilon^2 (1 - \eta) f(x_0) p(x_0) + o(\varepsilon)$$

where  $p \in H_0^1(\Omega)$  is the solution to the adjoint equation to the unperturbed problem

$$\begin{cases} -\Delta p_{\Omega} = -j'(u_{\Omega}) & \text{in } \Omega \\ p_{\Omega} = 0 & \text{on } \partial\Omega. \end{cases}$$

This yields that the topological derivative is the function  $d_T J : \Omega \rightarrow \mathbb{R}$

$$d_T J(x_0) = (1 - \eta) f(x_0) p(x_0).$$

In general topological and shape derivatives will share very similar structures. Indeed, the following proposition from [254] expresses this relationship.

**Proposition 1.3.** *If the criterion  $J(\Omega_{x_0, \varepsilon})$  for  $\varepsilon$  small enough, it admits the topological asymptotic expansion of the form:*

$$J(\Omega_{x_0, \varepsilon}) = J(\Omega) + \rho(\varepsilon) d_T J(\Omega)(x_0) + o(\rho(\varepsilon)), \quad \frac{o(\rho(\varepsilon))}{|\rho(\varepsilon)|} \xrightarrow{\varepsilon \rightarrow 0} 0,$$

where  $J(\Omega)$  is the shape functional associated to the original (unperturbed) domain and the function  $d_T J(\Omega)(x_0)$  is the topological derivative of the shape functional  $J$ . Then, for every fixed vector field such that  $\theta|_{\partial B(x_0, \varepsilon)} = -n_{\partial B(x_0, \varepsilon)}$ , the topological derivative can be written by:

$$d_T J(\Omega)(x_0) = \lim_{\varepsilon \rightarrow 0} \left( \frac{1}{\rho'(\varepsilon)} J'(\Omega_{x_0, \varepsilon})(\theta) \right),$$

where  $J'(\Omega_{x_0, \varepsilon})(\theta)$  is the shape derivative of  $J$  at  $\Omega_{x_0, \varepsilon}$  in the direction  $\theta$ .

This result will become clear when we present our work in Chapter 4. For more detailed information on topological derivatives, we refer to [31, 300].

## 1.5 Body-fitted shape optimization

The representation of shapes poses a significant challenge, specially when it comes to tracking their evolution. Classical fields such as computer graphics [155, 209], computer vision [261], robotics [329], and image processing [343]. This challenge arises from the need to capture a shape’s essential features while maintaining computational efficiency and adaptability across different applications. Additionally, the dimensionality of shapes introduces complexity, as 2D shapes, typically represented as closed curves, differ fundamentally from 3D shapes, which involve surfaces and volumes. An example of such a representation arises in CAD software (see Fig. 1.4).

This section introduces the “body-fitted” approach, which allows for alternating between implicit and explicit representations of a shape  $\Omega(t)$  to track large deformations induced by a velocity field  $V(t, x)$ , within a computational domain  $D \subset \mathbb{R}^d$ , including potential topological changes, while maintaining an accurate meshed representation throughout the process. This approach, based on [12, 13, 14], combines the two complementary representations of  $\Omega(t)$  during its evolution.

On one hand,  $\Omega(t)$  is explicitly discretized as a submesh of a high-quality surface triangulation  $\mathcal{T}$  of  $D$ , enabling precise geometric computations and solving boundary value problems using the finite element method, thus allowing accurate evaluation of the velocity field  $V(t, x)$  (which corresponds to the shape gradient, in the present application). On the other hand,  $\Omega(t)$  is implicitly described using the level set method, represented as the negative subdomain of a scalar function  $\phi(t, \cdot) : D \rightarrow \mathbb{R}$ , which accommodates arbitrarily large motions of  $\Omega(t)$ . This strategy will serve as the basis of all the examples generated in this thesis, and also will serve as the precursor to the strategy described in Chapter 3, where it will be extended to the case where  $D$  is replaced by a surface in  $\mathbb{R}^d$ .

### 1.5.1 Shape representation via triangulations

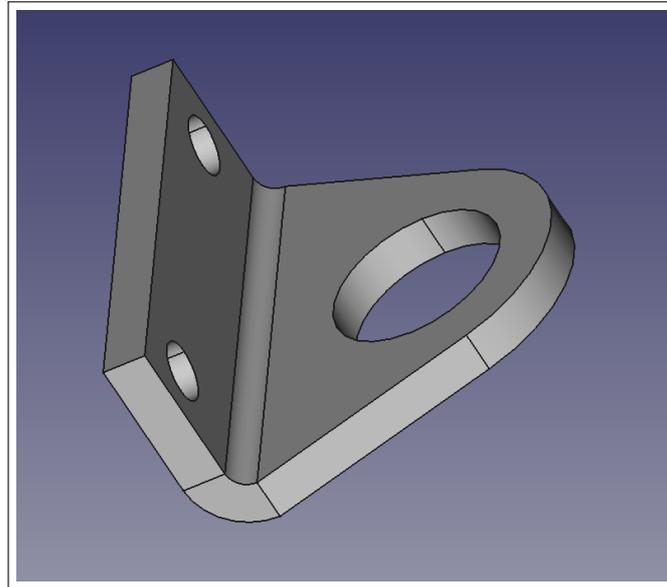
In realistic applications of shape optimization, the challenge involves modeling a physical problem where the shape, denoted by  $\Omega$ , is treated as a variable. This problem is typically described by a system of partial differential equations, with  $\Omega$  serving as the domain. To discretize the problem, a mesh  $\mathcal{T}$  of  $\Omega$  is often employed, following well-established numerical techniques such as the finite element method [92, 136]. The finite element method operates on  $\mathcal{T}$  by constructing finite element spaces that are used for the discretization of the PDEs, allowing for the approximation of solutions in a computationally feasible way. The mesh  $\mathcal{T}$  is defined as a collection of open simplices (triangles in 2D, tetrahedra in 3D), denoted as  $\mathcal{T} = \{\tau_k\}_{k=1, \dots, N}$ , which cover the domain  $\Omega$  such that:

$$\bar{\Omega} = \bigcup_{k=1}^N \bar{\tau}_k.$$

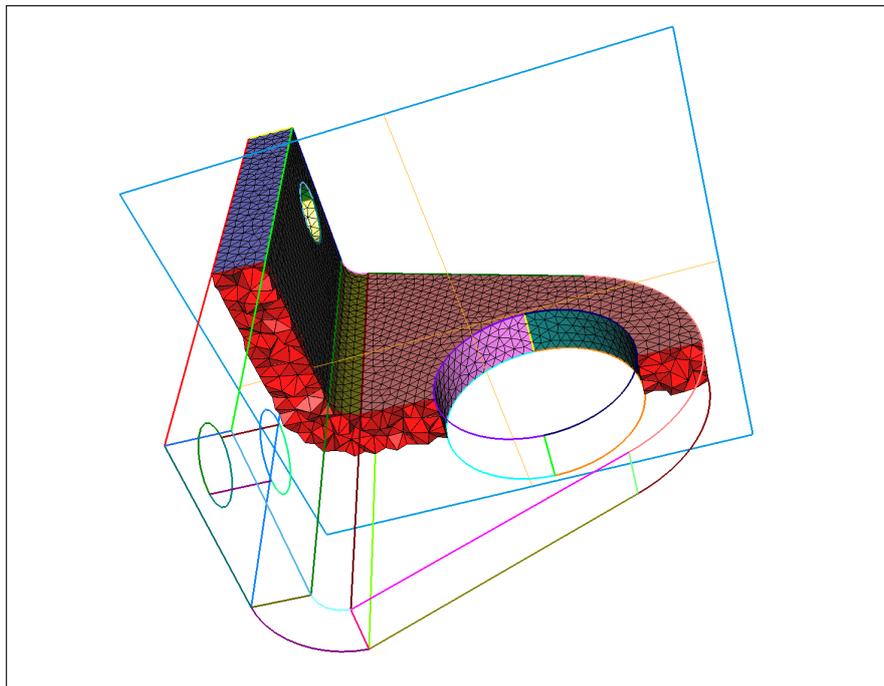
This means that the union of the closures of all the simplices forms the closure of the domain  $\Omega$ . Additionally, the triangulation must satisfy two important conditions:

1. **Non-overlapping simplices.** The simplices  $\tau_k$  must not overlap, i.e.,  $\tau_k \cap \tau_l = \emptyset$  whenever  $k \neq l$ .
2. **Conformity of the mesh.** The mesh  $\mathcal{T}$  must be conforming, meaning that for any two simplices  $\tau_k$  and  $\tau_l$ , their intersection  $\bar{\tau}_k \cap \bar{\tau}_l$  is either a shared vertex, a shared edge, or, in the case of three-dimensional shapes, a shared face of the triangulation  $\mathcal{T}$ .

Mesh generation is a complex process, and commonly used methods such as Delaunay triangulation do not always guarantee success, particularly in producing high-quality meshes, as certain shapes cannot be subdivided into tetrahedra without introducing additional vertices (see [85]). Mesh quality refers to the arrangement and shape of elements in representing the domain accurately while ensuring efficient numerical computation. While various metrics assess mesh quality, it is well established that mesh quality significantly impacts the accuracy of the discretized PDE solution (see [92, 136]). Given the extensive existing literature on mesh generation, we do not explore this topic further; a comprehensive explanation of mesh generation techniques, particularly in the finite element context, can be found in [152]. High-quality mesh generation remains an open problem with ongoing research and improvements. Several mesh generators exist, and some have been used to generate the meshes in this thesis. For more details, see [162] for GMSH, [182] for FreeFem++’s mesh generator, and [179] for TetGen; see Fig. 1.4 for an example of such a mesh.



(a) CAD representation of the domain  $\Omega$ .



(b) Mesh discretization of the domain  $\Omega$  splitted down the middle to show the 3D elements. The figure shows that the mesh has been discretized into 1D (edges), 2D (triangles), and 3D (tetrahedral) elements. The colors corresponds to different features of the mesh which have been identified by the mesh generator. In this case, the mesh was generated from the CAD design by GMSH.

Figure 1.4: Different representations of a mechanical piece  $\Omega$ , where one can solve a partial differential equation using the finite element method once the domain has been discretized into a mesh.

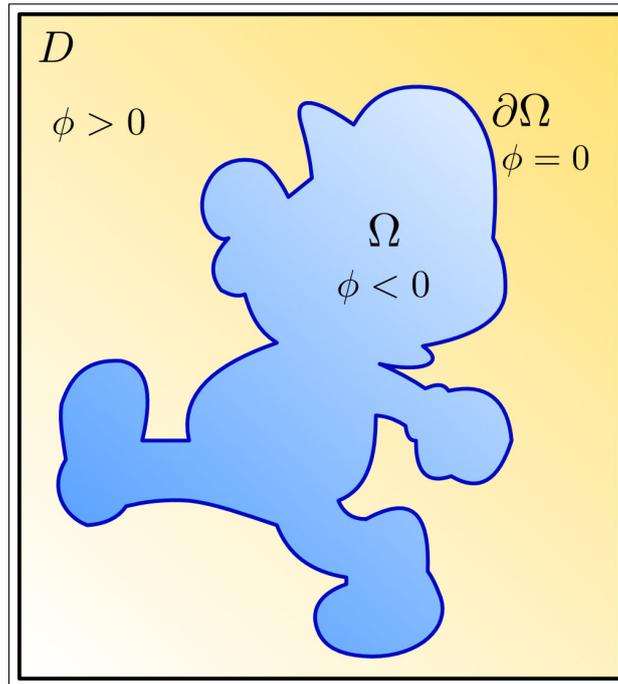


Figure 1.5: The shape  $\Omega$  is implicitly represented by a function  $\phi(x)$ , where  $\phi < 0$  inside the domain  $\Omega$ ,  $\phi = 0$  on the boundary  $\partial\Omega$ , and  $\phi > 0$  outside.

### 1.5.2 The level set method for tracking the motion of a domain

The level set method, introduced by Stanley Osher and James Sethian in the late 1980s [258], is a powerful numerical technique for tracking interfaces and shapes, particularly in problems involving curvature flow and fluid dynamics. Its versatility and robustness have led to widespread adoption across fields such as computer vision, image processing, computational geometry, and control theory, where the ability to track evolving shapes accurately is essential. Among these applications, shape optimization stands out as a key area where the level set method has had a profound impact. Many works in the shape optimization literature utilize this framework [67, 325, 17, 326] and in today's world, it has been widely accepted as a robust and efficient way of representing shapes.

The success of the level set method can be attributed to its flexibility; in particular it can represent complex and evolving shapes without the need for explicit parameterization and permits topological changes like merging or breaking apart of interfaces. The core concept involves representing a moving interface as the zero level set of a function  $\phi : [0, T] \times \mathbb{R}^d \rightarrow \mathbb{R}$ , where  $T > 0$  is the final time. For a domain  $\Omega(t) \subset \mathbb{R}^d$  with an evolving boundary  $\partial\Omega(t)$ , the interface is implicitly given by:

$$\partial\Omega(t) = \{x \in \mathbb{R}^d \mid \phi(t, x) = 0\},$$

where  $\phi(t, x) < 0$  inside the domain,  $\phi(t, x) = 0$  on the boundary, and  $\phi(t, x) > 0$  outside the domain; see Fig. 1.5.

**Remark 1.3.** In general, the confines of utilizing a discretization means that we usually consider a hold-all computational domain  $D \subset \mathbb{R}^d$  which serves as an approximation of the entire free space. In this case the level-set function can be defined as:

$$\forall x \in D, \quad \begin{cases} \phi(t, x) < 0 & \text{if } x \in \Omega(t), \\ \phi(t, x) = 0 & \text{if } x \in \partial\Omega(t), \\ \phi(t, x) > 0 & \text{if } x \in D \setminus \Omega(t), \end{cases}$$

for every time  $t > 0$ .

The evolution of the interface, under a velocity field  $V(t, x)$ , is governed by the level set function

$\phi(t, x)$  through a Hamilton-Jacobi-type equation:

$$\frac{\partial \phi}{\partial t}(t, x) + V(t, x) \cdot \nabla \phi(t, x) = 0. \quad (1.13)$$

When the velocity is normal to the interface, this simplifies to:

$$\frac{\partial \phi}{\partial t}(t, x) + v(t, x) |\nabla \phi(t, x)| = 0,$$

with the scalar field  $v(t, x)$  representing the normal speed. In the case of shape optimization, the time variable  $t$  is an artificial parameter. It represents the artificial descent parameter (ie. the iterations) as the shape optimization process advances, while the velocity field  $V(t, x)$  is a descent direction  $\theta$  inferred from the shape derivative  $J(\Omega(t))(\theta)$  at the shape  $\Omega(t)$ .

Aside from providing an efficient and simple representation of the underlying shape  $\Omega$ , the level-set function  $\phi$  also provides easy expressions for different geometric objects of related to the domain or its boundary. Let  $\phi$  be a level-set function for  $\Omega$  (where time dependence has been ignored).

- The normal vector  $n_{\partial\Omega}(x)$  to  $\partial\Omega$ , pointing outward  $\Omega$  reads:

$$n_{\partial\Omega}(x) = \frac{\nabla \phi(x)}{|\nabla \phi(x)|}, \quad x \in D.$$

The above formula actually accounts for an extension of  $n_{\partial\Omega}(x)$  from  $\partial\Omega$  to the computational domain  $D$  as a whole.

- The second fundamental form  $\mathbb{I}(x)$  and the mean curvature  $\kappa(x)$  at the point  $x \in \partial\Omega$  are the  $d \times d$  matrix and real value defined by:

$$\forall \xi \in \mathbb{R}^d, \quad \mathbb{I}(x)\xi \cdot \xi = \nabla \cdot \left( \frac{\nabla \phi}{|\nabla \phi|} \right) (x) \xi \cdot \xi \quad \text{and} \quad \kappa(x) = \nabla \cdot \left( \frac{\nabla \phi}{|\nabla \phi|} \right) (x).$$

Furthermore, if  $\phi_A$  and  $\phi_B$  are two different level-set functions for shapes  $A$  and  $B$ , then we have the following operations.

- The union  $A \cup B$  of the two shapes is computed as:

$$\phi_{A \cup B}(x) = \min(\phi_A(x), \phi_B(x)).$$

- The intersection  $A \cap B$  is given by the maximum of their level set functions:

$$\phi_{A \cap B}(x) = \max(\phi_A(x), \phi_B(x)).$$

- To compute the difference  $A \setminus B$ , the level set function is defined as:

$$\phi_{A \setminus B}(x) = \min(\phi_A(x), -\phi_B(x)).$$

### 1.5.3 Numerical discretization of the advection equations

The practical solution of (1.13) for shape optimization involves discretizing the time interval  $[0, T]$  into subintervals  $(t_n, t_{n+1})$  where  $n \in \mathbb{N}$ . In this setting, the sequence of shapes  $\Omega^n = \Omega(t_n)$  is used to derive the descent direction  $V^n(x) := V(t_n, x) = \theta^n(x)$ , obtained from the shape derivative  $J(\Omega^n)(\theta)$ . The velocity field is assumed constant over each time step  $(t_n, t_{n+1})$ , i.e.,

$$\forall t \in (t_n, t_{n+1}), \quad V(t, x) \approx V(t_n, x). \quad (1.14)$$

This time discretization results in a sequence of linear advection equations:

$$\frac{\partial \phi^n}{\partial t}(t, x) + V^n(x) \cdot \nabla \phi^n(t, x) = 0, \quad \text{for } t \in (t_n, t_{n+1}) \times D, \quad (1.15)$$

with the initial condition  $\phi^n(0, x)$  representing the level-set function corresponding to the shape at time  $\Omega(t_n)$ .

The resolution of (1.15) is well-established in the literature when the mesh  $\mathcal{T}$  is a Cartesian grid. Many finite difference schemes have been developed [236, 223] and are implemented in numerical libraries. However, for unstructured simplicial meshes, these techniques are not applicable, requiring more advanced methods such as discontinuous Galerkin schemes [120].

A more general approach to solving the advection equation (1.15), independent of mesh structure, is the method of characteristics. Under appropriate regularity assumptions on the velocity field  $V^n(x)$ , the exact solution  $\phi^n(t, x)$  is:

$$\phi^n(t, x) = \phi_0^n(X(0, t, x)), \quad t \in (t_n, t_{n+1}), \quad x \in D, \quad (1.16)$$

where  $s \mapsto X(s, t, x)$  is the characteristic curve of  $V^n$ , which solves the ordinary differential equation:

$$\dot{X}(s, t, x) = V^n(X(s, t, x)) \quad \text{for } s \in (t_n, t_{n+1}), \quad X(t, t, x) = x. \quad (1.17)$$

In simple terms,  $s \mapsto X(s, t, x)$  describes the trajectory of a particle driven by the velocity field  $V^n$ , starting at  $x$  at time  $t$ . Using (1.16), the value of  $\phi^n(t_{n+1}, x)$  at a given vertex  $x$  of the computational mesh  $\mathcal{T}$  can be computed by solving the backward-in-time equation (1.17). Implementations of this technique already exist as open-source libraries in [106], which utilizes a 4th-order Runge-Kutta scheme, with the associated theoretical background developed in [66].

#### 1.5.4 The signed distance function

No assumptions have been made so far regarding the specific nature of the level set function  $\phi$  used to track the domain  $\Omega(t)$ . Historically, the application of distance functions to geometric problems can be traced back to classical differential geometry. However, their significant utility in optimization and computational methods only became apparent with the development of the level set method; the signed distance function naturally emerged as an ideal candidate for a level set function, owing to its advantageous mathematical properties, simplicity, and flexibility in describing evolving boundaries. The signed distance function is based on the distance  $d(x, \Omega) := \min_{y \in \partial\Omega} |x - y|$ , which measures the shortest distance from a point  $x$  to the boundary  $\partial\Omega$  of the domain  $\Omega$ . Formally, the signed distance function is defined as:

$$\forall x \in D, \quad d_\Omega(x) = \begin{cases} -d(x, \Omega) & \text{if } x \in \Omega, \\ 0 & \text{if } x \in \partial\Omega, \\ d(x, \Omega) & \text{if } x \in D \setminus \Omega, \end{cases} \quad (1.18)$$

where  $D$  represents the computational domain, and in a more general setting,  $D = \mathbb{R}^d$ .

One of the most compelling reasons why  $d_\Omega$  is particularly well-suited for use as a level set function lies in its “signed distance property”. The signed distance function  $d_\Omega$  has a smooth gradient  $\nabla d_\Omega$  near the boundary, with the magnitude of the gradient being exactly 1. On  $\partial\Omega$ , the gradient is precisely the normal vector field (see [165]). More formally:

**Proposition 1.4.** *Let  $\Omega$  be of class  $C^k$ , where  $k \geq 2$ . Then  $d_\Omega$  is of class  $C^k$  for points  $x \in D$  that are sufficiently close to  $\partial\Omega$ , and*

$$|\nabla d_\Omega(x)| = 1. \quad (1.19)$$

*Additionally, if  $x \in \partial\Omega$ , then  $\nabla d_\Omega(x) = n_{\partial\Omega}(x)$ .*

This regularity near the zero level set simplifies the computation of key geometric quantities such as normals and curvatures, which are essential for shape optimization. Indeed, if one is able to compute the signed distance function for  $\Omega$  in  $D$ , then one only need to compute its gradient to have a natural extension of the normal vector field  $n_{\partial\Omega}$  to all of  $D$ . Additionally, geometric quantities, such as those in Section 1.5.2, can be expressed in simpler terms:

$$\mathbb{I}(x) = \nabla^2 d_\Omega(x) \quad \text{and} \quad \kappa(x) = \Delta d_\Omega(x),$$

where  $\nabla^2$  represents the Hessian matrix, and  $\Delta$  the Laplacian operator.

In light of this, the numerical computation of the signed distance function is crucial to our study. Its computation is based on the well-known “fast marching method” on Cartesian grids, introduced by James

---

Sethian in [291] and further developed in [292]. The central idea behind this efficient numerical method is to solve the Eikonal equation:

$$\begin{cases} |\nabla u(x)| = 1, & \text{for } x \in \Omega, \\ u(x) = 0, & \text{for } x \in \partial\Omega, \end{cases}$$

by leveraging the fact that information propagates outward in a monotonic manner. Specifically, once a wavefront passes through a point, the value of the solution at that point becomes fixed and does not change, allowing for efficient computation. In this manner, the algorithm has a complexity of  $O(n \log n)$  where  $n$  is the number of grid points. Extensions to the case of an unstructured simplicial mesh have already been proposed in the works [205, 268], and open source implementations are freely available [106, 105], which we employ throughout this work.

### 1.5.5 Discretization of the negative sub-domain of the level-set function

The next step in the body-fitted approach is transitioning from a level-set representation to a meshed representation of the shape  $\Omega$ . An exact meshed representation enables the use of finite element discretization techniques, facilitating the modeling of the underlying physics.

Let  $\mathcal{T}$  be a triangulation of a computational domain  $D \subset \mathbb{R}^d$ , and let  $\phi : D \rightarrow \mathbb{R}$  be a level set function for a region  $\Omega \subset D$ , given by its values at the vertices of  $\mathcal{T}$ . Our goal is to construct a new, high-quality computational mesh  $\tilde{\mathcal{T}}$  of  $D$ , consisting of two submeshes,  $\tilde{\mathcal{T}}_{\text{int}}$  and  $\tilde{\mathcal{T}}_{\text{ext}}$ , for the regions  $\Omega$  and  $D \setminus \bar{\Omega}$ , respectively. This can be achieved through various strategies (see [153, 154]); here, we adopt the approach from [102]. This method involves two main steps:

1. Triangles  $\tau \in \mathcal{T}$  that intersect the zero level set  $\Gamma = \partial\Omega$  of  $\phi$  are identified based on the function values at the vertices, and  $\Gamma$  is explicitly incorporated into  $\mathcal{T}$ . This is done using the marching tetrahedra algorithm [127], a variant of the marching cubes method [226], which splits each intersecting triangle into a valid, conforming configuration with explicit representation of the segment  $\Gamma \cap \tau$ . The resulting triangulation  $\mathcal{T}_{\text{temp}}$  of  $D$  contains submeshes  $\mathcal{T}_{\text{temp,int}}$  and  $\mathcal{T}_{\text{temp,ext}}$  for  $\Omega$  and  $D \setminus \bar{\Omega}$ . However,  $\mathcal{T}_{\text{temp}}$  contains poorly shaped, nearly flat elements, which are unsuitable for accurate geometric and finite element computations (see [92]).
2. The intermediate mesh  $\mathcal{T}_{\text{temp}}$  is iteratively refined to improve element quality, aiming to make them as equilateral as possible. This process results in a high-quality mesh  $\tilde{\mathcal{T}}$  of  $D$ , with explicit discretizations of both  $\Omega$  and  $D \setminus \bar{\Omega}$ .

The second step is the most complex, involving geometric calculations such as determining the normal vector  $n_{\partial\Omega}$  to  $\partial\Omega$ , and its variation at neighboring vertices of  $\mathcal{T}$ . These computations help determine the optimal element sizes for accurately approximating  $D$  and  $\Omega$ .

These steps are detailed in Section 3.2.4, where we extend the method to the case of surfaces, employing the same techniques. For now, we omit the exact operational details and note that most of these methods have already been implemented in the general-purpose open-source library MMG, which is dedicated to simplicial remeshing. For a more comprehensive presentation, refer to [102], and for recent developments on the subject, see [38].

## 1.6 The classical cantilever example

In this section, we illustrate the previous techniques with one of the most classical examples in the literature – the so called cantilever example [15, 22, 52, 104]. The objective is to minimize the compliance of a beam, which represents the total strain energy of a mechanical structure  $\Omega$  under applied load – so that minimizing compliance amounts to maximizing the structural stiffness. The beam, fixed at one end and subject to external forces at specified points, undergoes shape or topology changes satisfying constraints such as a fixed material volume.

### 1.6.1 The optimization problem

Let  $\Omega$  denote the shape, whose boundary  $\partial\Omega$  is decomposed into three disjoint pieces:

$$\partial\Omega = \bar{\Gamma}_D \cup \bar{\Gamma}_N \cup \bar{\Gamma},$$

where:

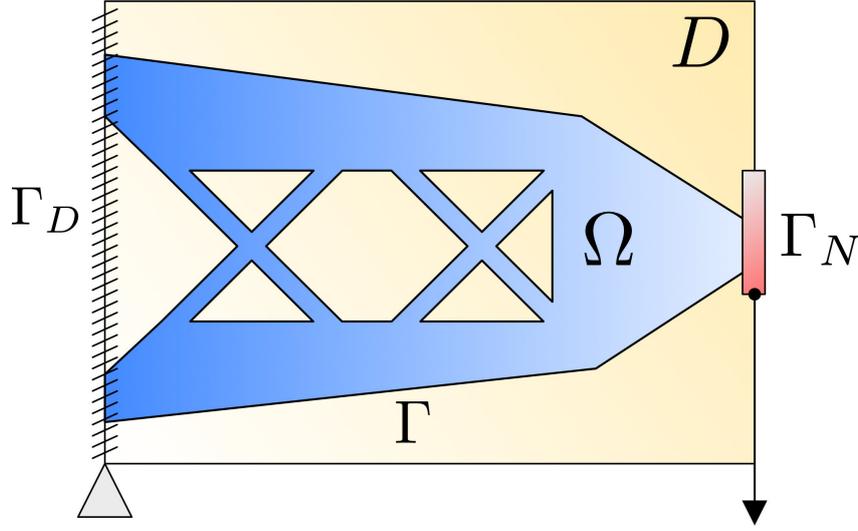


Figure 1.6: The image shows a typical schematic from shape optimization of a cantilever beam. The beam is fixed on the left side  $\Gamma_D$  and subjected to a vertical force on the right side  $\Gamma_N$ . The optimized structure  $\Omega$  minimizes compliance by efficiently distributing material, leaving voids to reduce weight while maintaining structural stiffness. The boundary  $\Gamma$  is the “traction-free” part of  $\partial\Omega$ , and the design region  $D$  is highlighted in yellow, representing the area where the shape optimization occurs.

- The displacement of the shape  $\Omega$  is prevented on the region  $\Gamma_D$ ;
- The region  $\Gamma_N$  is subjected to surface loads  $g \in L^2(\Gamma_N)^d$ ;
- The remaining part  $\Gamma$  is traction-free.

We seek to minimize the compliance of the shape  $\Omega \subset \mathbb{R}^d$ , while utilizing the least amount of material possible:

$$\min_{\Omega \subset \mathbb{R}^d} \int_{\Omega} Ae(u_{\Omega}) : e(u_{\Omega}) \, dx + \ell \text{Vol}(\Omega), \quad (1.20)$$

where  $u_{\Omega}$  is the unique  $H_{\Gamma_D}^1(\Omega)$  solution to the boundary value problem:

$$\begin{cases} -\nabla \cdot (Ae(u_{\Omega})) = 0 & \text{in } \Omega, \\ Ae(u_{\Omega})n_{\partial\Omega} = g & \text{on } \Gamma_N, \\ Ae(u_{\Omega})n_{\partial\Omega} = 0 & \text{on } \Gamma, \\ u_{\Omega} = 0 & \text{on } \Gamma_D. \end{cases} \quad (1.21)$$

Here,  $e(u) := \frac{1}{2}(\nabla u + \nabla u^T)$  is the strain tensor associated with a displacement field, and  $A$  is the Hooke’s tensor, defined by:

$$Ae = 2\mu e + \lambda \text{tr}(e)\mathbf{I},$$

where  $\mathbf{I}$  is the identity  $d \times d$  matrix, and  $\lambda, \mu$  are the Lamé parameters of the constituent material, satisfying  $\mu > 0$  and  $\lambda + \frac{2\mu}{d} > 0$ . These parameters are often expressed in terms of more physically relevant quantities:

$$\mu = \frac{E}{2(1+\nu)}, \quad \lambda = \frac{E\nu}{(1+\nu)(1+\nu(1-d))},$$

where  $E$  is the Young’s modulus, representing the material’s resistance to traction, and  $\nu$  is the Poisson’s ratio, measuring the transverse deflection of the material when subjected to traction loads. For a more comprehensive introduction to linearized elasticity, see [93].

In this context, the shape derivative of the functional in (1.20), is given by [15]:

$$J'(\Omega)(\theta) = - \int_{\partial\Omega} (Ae(u_{\Omega}) : e(u_{\Omega}) - \ell) \theta \cdot n_{\partial\Omega} \, ds. \quad (1.22)$$

Utilizing this information, we propose the following algorithm to find the optimal shape for (1.20).

**Algorithm 1:** Body-fitted optimization of the compliance for a shape  $\Omega$ .

**Input:** Mesh  $\mathcal{T}^0$  of  $\Omega$ , whose interior contains an explicit discretization of the initial region  $\Omega^0$ .

**for**  $n = 0, \dots, N - 1$  **do**

1. Compute the signed distance function  $d_{\Omega^n}$  to  $\Omega^n$  at the vertices of the mesh  $\mathcal{T}$ .
2. Solve the equation (1.21) utilizing the interior domain  $\mathcal{T}_{\text{int}}^n$ .
3. Infer a descent direction from  $\theta^n$  using the shape derivative expression  $J'(\Omega^n)(\theta)$  in (1.22).
4. Solve the advection equation:

$$\begin{cases} \frac{\partial \phi}{\partial t}(t, x) + \theta^n(x) \cdot \nabla_{\partial \Omega} \phi(t, x) = 0 & \text{for } (t, x) \in (0, \Delta t) \times \partial \Omega, \\ \phi(0, x) = d_{\Omega^n}(x) & \text{for } x \in \partial \Omega, \end{cases}$$

on the total mesh  $\mathcal{T}^n$  of  $\partial \Omega$ . A new level set function  $\phi^{n+1} = \phi(\Delta t, \cdot)$  is obtained for

$$\Omega^{n+1} = \{x \in \partial \Omega, \phi^{n+1}(x) < 0\}.$$

5. From the datum of  $\phi^{n+1}$  at the vertices of  $\mathcal{T}^n$ , create a new, high-quality mesh  $\mathcal{T}^{n+1}$  of  $\Omega^{n+1}$  made of two submeshes  $\mathcal{T}_{\text{int}}^{n+1}$  and  $\mathcal{T}_{\text{ext}}^{n+1}$  for  $\Omega^{n+1}$  and  $D \setminus \overline{\Omega^{n+1}}$ , respectively.

**end**

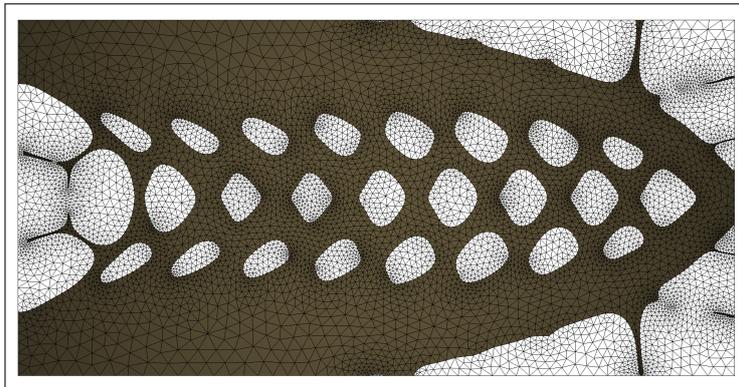
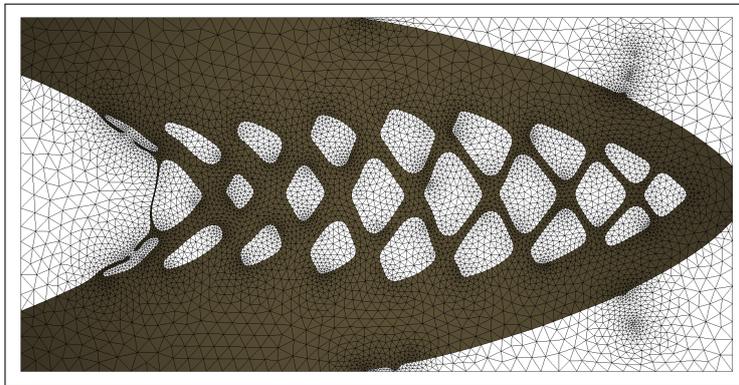
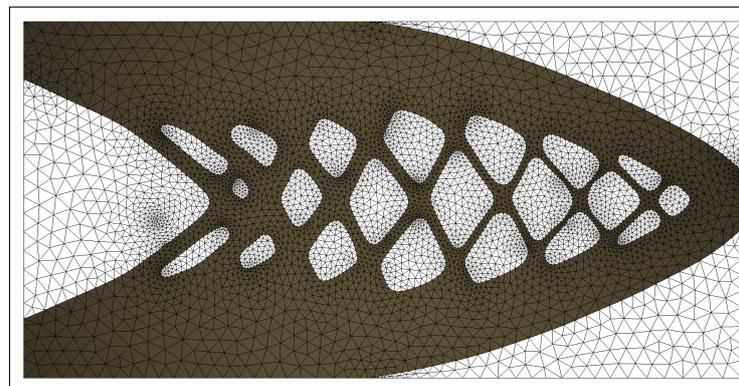
**Output:** Mesh  $\mathcal{T}^N$  containing an explicit discretization  $\mathcal{T}_{\text{int}}^N$  of  $\Omega^N$ .

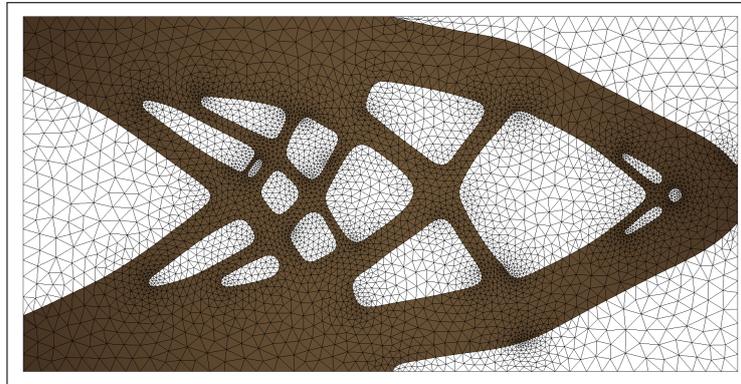
## 1.6.2 A numerical experiment

For our experiment, we choose  $\mu = 0.3846$ ,  $\lambda = 0.5769$ ,  $\ell = 0.4$ , with the initial design pictured in Fig. 1.7. In the picture one can see how the shape evolves from the initial topology up to the classical cantilever shape often seen in the literature. Here, the mesh lines are visible to illustrate that at any moment of the process, we can track the exact mesh representation of the domain  $\Omega$  (in brown) and how the level-set method permits various topological changes due to the merging of boundaries. The objective is clearly seen to be minimized as the iterations pass.

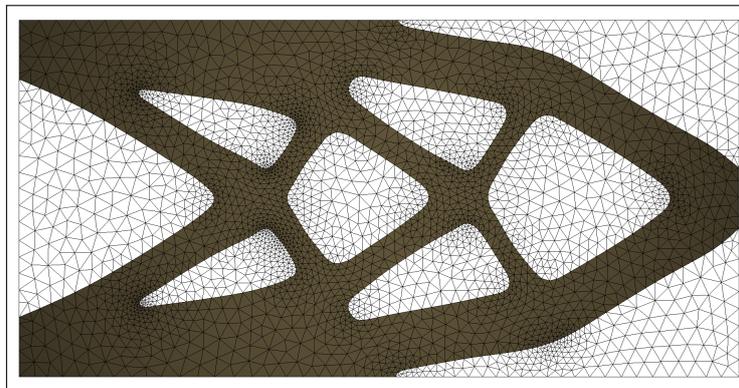


Figure 1.7: Initial design for the cantilever.

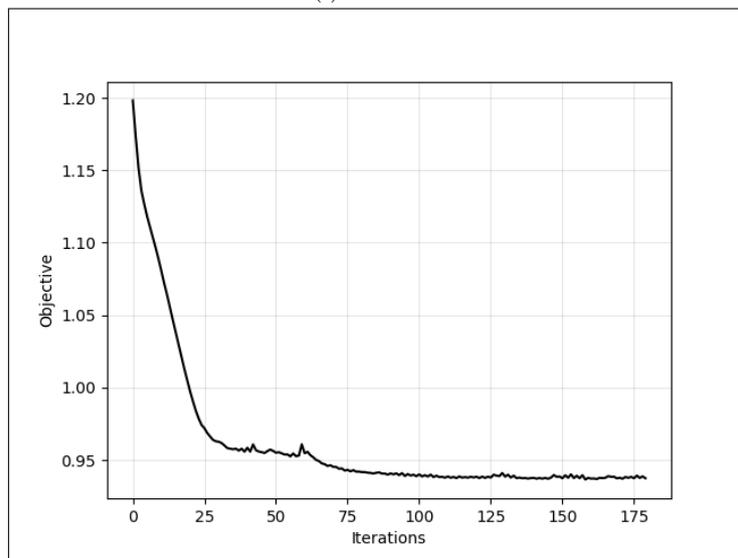
(a)  $i = 10$ (b)  $i = 20$ (c)  $i = 30$ (d)  $i = 40$ Figure 1.8: Snapshots of the shape optimization process described in [Algorithm 1](#).



(e)  $i = 80$



(f)  $i = 180$

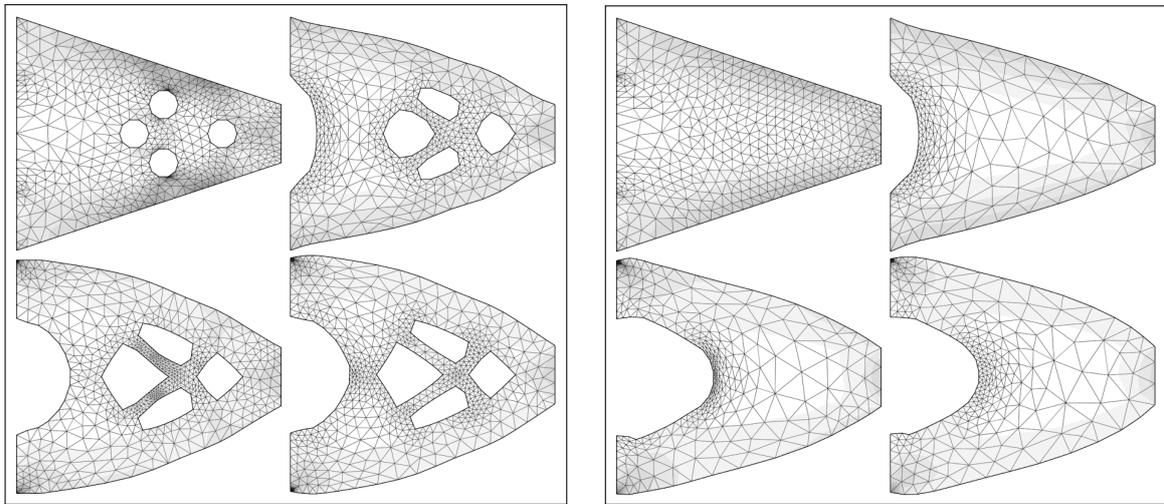


(g) Convergence history for the objective functional of problem (1.20).

Figure 1.8: Snapshots of the shape optimization process described in Algorithm 1.

## 1.7 Limitations, variants and the homogenization phenomenon

Two significant challenges frequently arise when solving problems like (1.20): local minima and the non-existence of classical solutions. The cost function is typically non-linear and non-convex, which leads to multiple valleys and peaks. As a result, local minima often emerge, where a shape has a lower cost than nearby configurations but is not globally optimal. Gradient-based optimization algorithms, which adjust the shape iteratively based on local information (such as shape derivatives), are especially prone to converging to these local minima [15]. Consequently, the final shape may be suboptimal and highly sensitive to the initial guess. For example, as shown in Fig. 1.9, using a mesh deformation technique (displacing mesh vertices in the direction of the descent vector  $\theta$ ) can lead to different local minima. One mesh, starting with holes, converges to a minimizer with the same topology, while another starts without holes and similarly converges to a local minimizer with the same initial topology. These issues motivate the development of methods that can optimize shapes while allowing for topological changes.



(a) Evolution process of an initial mesh topology with holes.

(b) Evolution process of an initial mesh topology without holes.

Figure 1.9: Two different local minimizers, with different topologies. Illustrations taken from §6.5.3 in [22].

An even more fundamental issue is the generic non-existence of classical solutions, where optimal shapes may not exist. Though at first, this might seem like a problem limited to the theoretical realm, it has serious practical implications. Algorithms often fail to converge under mesh refinement and are highly sensitive to initial conditions, a direct consequence of this non-existence. This lack of convergence means that the computed solution cannot be guaranteed to be optimal, even approximately. To address these challenges, one approach is to impose additional constraints that restrict the class of admissible designs [19], ensuring the existence of an optimal solution and improving the reliability of the optimization process. For instance, let us mention the addition of a perimeter constraint to guarantee the compactness of minimizing sequences and so, existence of optimal shapes [26].

This non-existence of solutions is largely attributed to the homogenization phenomenon, which manifests both theoretically and at the numerical level. From a numerical perspective, optimization algorithms tend to exploit combinations of different material phases or voids to improve the design's performance at fine scales. For example, consider minimizing a functional like the mean temperature  $J(\Omega) = \frac{1}{\text{Area}(\Omega)} \int_{\Omega} u_{\Omega} dx + \text{Vol}(\Omega)$ , where  $u_{\Omega}$  is the solution to (1.5), and material usage is also penalized. A density-based algorithm like SIMP [50] may produce designs featuring microstructures, as illustrated in Fig. 1.10. In these designs, the conductive material (represented by the black region) is distributed in branching structures to minimize material usage. However, such designs are often impractical, as they are difficult or even impossible to manufacture.

This lack of solution existence is primarily due to the homogenization phenomenon, which appears both theoretically and numerically. From a numerical perspective, optimization algorithms often exploit mixtures of different material phases or voids to enhance performance at fine scales.

For a detailed introduction to homogenization theory and its implications in shape optimization, refer

to [9, 8]. Specifically, §3.1 of [15] discusses the non-existence of optimal designs due to the homogenization phenomenon.

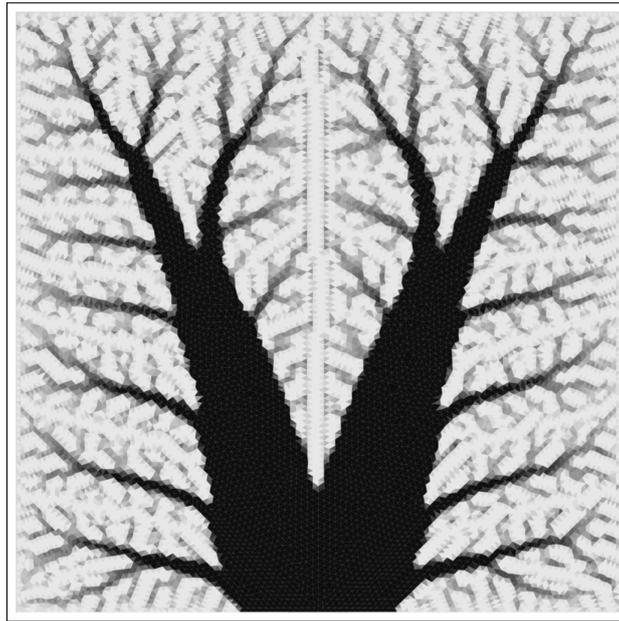


Figure 1.10: The image shows a branching structure (black regions) representing the conductive material, embedded in a lighter background (voids), characteristic of a homogenization phenomenon in shape optimization. The design features fine-scale microstructures that spread out like tree branches, aiming to minimize material usage while maintaining performance. This illustrates the emergence of complex geometries that are challenging to realize in practical applications, yet represent an optimal solution in the homogenized sense.

Despite these drawbacks, practical studies, such as [134], demonstrate that working within convex design spaces can yield local optima, although heavily dependent on the choice of parameters. Furthermore, theoretical studies like [71, 70] have established the existence of solutions under general objective functionals, by imposing additional constraints (e.g., shape regularity) on the design space.

## 1.8 State of the art for shape and topology optimization of surfaces

Despite significant advances in shape and topology optimization, most research has focused on volumetric domains, with limited attention given to the optimization of a region on a surface. Existing methods, as reviewed in this chapter, are predominantly designed for optimizing domains  $\Omega \subset \mathbb{R}^d$ . However, surface-based problems present additional complexities due to the intricate geometry of surfaces, requiring specialized computational techniques.

To compound the issue, and as highlighted in the introduction, surface optimization has applications across various multiple fields. In heat transfer, optimizing heat exchanger surfaces improves thermal efficiency, reduces material usage, and enhances overall heat management [203, 33]. In aerodynamics and hydrodynamics, optimizing surfaces such as airplane wings, boat hulls, and wind turbine blades reduces drag and enhances lift or propulsion efficiency [232, 160, 334]. In acoustics, surface shape is critical for controlling sound, from reducing vehicle noise to enhancing concert hall acoustics [224, 272]. Additionally, optimizing thin-walled structures such as shells and membranes used in ship hulls, roofs, and aircraft fuselages has advanced through both simplicial mesh and grid-based methods [91, 318, 338]. The diversity of these applications underscores the urgent need for a comprehensive framework to address these challenges.

This thesis bridges the gap in surface-based shape and topology optimization through four key contributions:

1. Theoretical developments and generalizations to manifolds, extending classical optimization techniques to Riemannian manifold settings.

2. Numerical evolution of domains on hypersurfaces embedded in  $\mathbb{R}^d$ , with computational strategies tailored to the challenges of surface optimization.
3. Optimization of boundary conditions for various PDEs, introducing new methods for optimizing regions that support boundary conditions across different physical contexts.
4. The development of open-source programming tools that facilitate the implementation of the methods introduced in this thesis.

In the final sections of this chapter we aim to detail already existing work in the literature and detail our contributions in these key areas.

### 1.8.1 Theoretical developments and generalizations to manifolds

---

*The main contribution related to this topic is contained in [Chapter 2](#).*

---

We introduce a novel framework for the geometric optimization of regions on submanifolds within a general ambient Riemannian manifold  $M$ . While the classical boundary variation method, originally introduced by Hadamard [177], has been extensively explored in Euclidean spaces [22, 116, 185, 301], our contribution extends this approach to the Riemannian setting, a context that has not been thoroughly addressed in previous work. By employing the exponential map in a Riemannian manifold, we provide an equivalent representation to the “perturbation of identity” method, traditionally used in Euclidean spaces, and adapt it for submanifolds in a more general geometric setting.

While some prior works, such as [311] and [285], have explored related ideas—like applying Hadamard’s theorem to submanifolds and computing second-order directional shape derivatives—these studies are restricted to submanifolds of the ambient space  $\mathbb{R}^n$ . In contrast, our framework is situated in the more general setting of a complete Riemannian manifold, where the use of the exponential map allows us to handle the intrinsic curvature of the manifold, providing a more versatile and rigorous foundation for shape optimization.

Our approach builds upon previous advancements, such as the work of [188], which uses differential forms for shape derivatives, and offers a coordinate-independent and invariant-preserving model. While adopting a Riemannian viewpoint increases mathematical rigor, we argue that it results in more elegant proofs and a deeper understanding of the geometric aspects of shape optimization. Furthermore, this framework differs from other Riemannian perspectives, such as that in [287], which treats the set of all shapes as an abstract infinite-dimensional manifold. Instead, our focus remains on submanifolds of an ambient manifold, utilizing the boundary variation method to deform them.

In summary, the main contribution is the development of a comprehensive framework that generalizes classical shape optimization techniques to the Riemannian setting, providing new insights and tools for optimizing regions on curved submanifolds, a topic that has been relatively unexplored in previous literature.

### 1.8.2 Numerical evolution of domains on hypersurfaces embedded in $\mathbb{R}^d$

---

*The main contribution related to this topic is contained in [Chapter 3](#).*

---

The problem of representing and tracking the evolution of a domain  $G(t) \subset \mathbb{R}^d$  (where  $d = 2$  or  $3$ ) has been a central focus in various applied disciplines, such as computer graphics, fluid dynamics, fracture propagation, and shape optimization. Multiple numerical methods have been developed to tackle this challenge, but a common issue remains: balancing the accurate representation of  $G(t)$  with the robust handling of its evolution, especially when the velocity field  $V(t, x)$  depends on complex geometric features like curvature or the solution to boundary value problems.

Lagrangian methods, which track the motion of the mesh directly, often suffer from mesh degradation as the motion progresses, leading to invalid geometries and computational challenges. While there have been improvements, such as remeshing techniques or the modification of internal mesh velocities [37, 75], these methods are typically limited to small deformations. Some newer approaches, like the Deformable Simplicial Complex [89] and the X-mesh method [240], have pushed the boundaries of Lagrangian methods by addressing large deformations and topology changes. However, these techniques still struggle with accurately solving boundary value problems on degenerate meshes.

On the other hand, Eulerian methods, particularly the level set method [258], offer a more flexible framework for representing large deformations of  $G(t)$  without the need for explicit meshing. Despite their ability to handle complex motions, these methods are less suited for solving PDEs on the evolving domain due to the implicit nature of the representation.

While the evolution of domains in Euclidean space has been extensively studied, relatively little work has been done when the ambient medium is a manifold, such as a surface  $S \subset \mathbb{R}^3$ . This is surprising given the number of important applications, including geometric flows, surface-based texture generation in computer graphics, and physical processes occurring on surfaces (e.g., phase changes in materials). Although early efforts like [87] and [204] applied level set methods to track geodesic curvature flow on surfaces, these approaches were limited to specific surface parametrizations or graphs. More general frameworks were introduced later, such as those using a fixed level set function to describe the surface  $S$  and the domain  $G(t)$  evolving within it [54, 84], or the closest point method for solving PDEs on surfaces [230].

Our main contribution, is a new numerical methodology that advances the state of the art by enabling the tracking of arbitrarily large motions of  $G(t)$  on an ambient surface  $S \subset \mathbb{R}^3$ , while preserving both the topology and a high-quality mesh representation throughout the process. Building on the previous work on evolving domains in Euclidean space [12, 13, 14], we extend these techniques to the case where the ambient medium is a surface. Our approach combines two complementary representations:

1. An explicit discretization of  $G(t)$  as a submesh of a high-quality triangulation of  $S$ , allowing for precise geometric calculations and the solution of boundary value problems.
2. An implicit level set representation of  $G(t)$ , enabling arbitrarily large deformations and topological changes.

The core innovation of our work lies in the efficient meshing algorithms and numerical schemes that allow for seamless transitions between these two representations, overcoming limitations of both Lagrangian and Eulerian methods in previous studies.

### 1.8.3 Optimization of the regions supporting boundary conditions for different PDEs

---

*The main contribution related to this topic is contained in Chapter 4.*

---

The optimization of regions supporting boundary conditions for various partial differential equations is a growing field with numerous applications, yet it remains underexplored. In elasticity, for example, optimizing regions where Dirichlet and Neumann boundary conditions are imposed (e.g., clamp-locator systems) can improve the design of fixations or load application points. Previous studies have utilized density-based topology optimization, genetic algorithms, neural networks, and level-set methods [228, 201, 289, 332, 333, 345]. In acoustics, research has focused on optimizing the distribution of absorbing materials (Robin boundary conditions) to minimize sound pressure, as demonstrated in [119].

Recent works, such as [318], commonly use density-based optimization on fixed meshes, while body-fitted mesh evolution methods, derived from works like [14], have been applied to track large deformations, as seen in [234], which optimized Neumann eigenvalues on a 3D sphere. However, these methods are often case-specific and lack a unified framework for handling shape and topology optimization of regions supporting boundary conditions. This gap leaves theoretical frameworks and strategies for addressing large deformations or topology changes relatively undeveloped, despite the widespread importance of these problems.

Building on the theoretical contributions of [108] and [56], our work proposes a comprehensive framework for shape and topology optimization of regions  $G \subset \partial\Omega$  supporting boundary conditions for PDEs on a fixed domain  $\Omega$ . The work [108] offers a rare study on shape optimization, focusing on the sensitivity of a functional  $J(G)$  to small diffeomorphic perturbations of  $G$ , while [56] extends this understanding to singular perturbations where boundary conditions switch between Dirichlet and Neumann.

Our contribution combines and extends these approaches to develop a robust workflow. We integrate shape and topological derivatives to assess the sensitivity of  $J(G)$  to both small deformations of  $\partial G$  and singular perturbations by introducing small surface regions. While formal calculations are presented in the context of the conductivity equation, our methods are adaptable to more complex applications, such as acoustics and structural mechanics.

A key innovation of our approach is the dual application of asymptotic analysis: first, to smooth singular transitions between regions with different boundary conditions, simplifying the calculation of shape derivatives; and second, to explore singular perturbations by introducing small zones with modified boundary conditions to quantify topological sensitivity. This strategy allows us to propose a novel, flexible, and generalizable framework that has not been previously available in the literature.

From a theoretical perspective, we propose formal methods for calculating both shape and topological derivatives. These methods, detailed in the simpler context of the conductivity equation, can be extended to more complex scenarios like acoustics and structural mechanics. While shape derivatives are relatively straightforward for a range of problems, topological derivatives require more intricate adaptations, which we address in detail. Our work ultimately demonstrates how asymptotic analysis can simplify boundary condition repartition and enable the study of singular perturbations, expanding the applicability of shape and topology optimization techniques.

#### 1.8.4 Open-source programming tools for developing shape and topology optimization algorithms

---

*The main contribution related to this topic is contained in [Chapter 5](#).*

---

The release of the 99-line topology optimization MATLAB code, known as top99 [297], by Ole Sigmund in 2001 marked a pivotal moment in the field of structural and multidisciplinary optimization. This concise and accessible code quickly became a widely used educational tool, inspiring numerous contributions that distilled complex optimization algorithms into simplified formats. The impact of top99 extended beyond its immediate utility, fostering a culture of learning, experimentation, and innovation within the optimization community. In recent years, the number of educational tools and papers in this area has steadily grown [324], with many relying on simplified scripts or tools designed for specific tasks.

While such tools are valuable for demonstrating basic concepts and performing simple optimizations, they often fall short when addressing the complexities of real-world problems. Many shape optimization tasks involve nonlinear problems [310], integration with simulation software [4], mesh generation, and diverse mathematical solvers [202]—challenges that simplified tools cannot fully manage. Recent works, such as [104], have introduced useful libraries and repositories, but these often come with multiple dependencies and require significant technical expertise. Similar trends are seen in other scientific tools like [227, 107, 18], which, while technically rich, are better suited for academic prototyping rather than robust, scalable applications required in more complex scenarios. To truly meet the needs of engineers and researchers, shape optimization software must prioritize scalability, integration, and performance. Without a robust and well-engineered library, users are forced to develop ad-hoc solutions, leading to inefficient workflows, increased development time, and inconsistent results.

In response to these limitations, we introduce Rodin, a C++ library designed to provide a comprehensive solution for shape and topology optimization. Unlike existing tools, Rodin is a lightweight, modular finite element framework that integrates shape and topology optimization algorithms into a single application programming interface (API). Rodin is engineered for scalability, allowing users to seamlessly move from prototyping to larger, more complex problems without the need for extensive technical expertise or multiple dependencies.

Rodin offers a full suite of essential functionalities for advanced shape and topology optimization, all integrated under the hood. These include:

- Shape refinement and remeshing via MMG [103].
- Domain distancing using MSHDIST [105].
- Advection of level set functions via integration with the ISCD computational toolbox [66, 106].

These tools have been tried and tested for years in the context of shape optimization and are directly integrated into Rodin thanks to the fact that they are coded in C. As a C++ library, Rodin can interoperate with C natively, ensuring seamless integration and optimal performance. Rodin also introduces a FreeFem++-like approach for defining variational problems, combining quick prototyping and user-friendliness without sacrificing interoperability. It employs the finite element method, to assemble the associated stiffness operators and mass vectors that are often employed to simulate physical problems. This approach enhances the ease of use while maintaining the flexibility to handle complex problems. The library provides access to a range of linear solvers, including Krylov solvers like GMRES [279] and

direct solvers like UMFPACK [112], all fully integrated within the C++ ecosystem. This eliminates the need for external software, with third-party libraries directly incorporated into the library's build system.

To the best of our knowledge, Rodin is a novel contribution in the field, with no other C++ libraries offering this level of integration, flexibility, and scalability for shape and topology optimization.

## Chapter 2

# Towards a framework for the optimization of domains on manifolds

---

In this chapter, our objective is to establish a framework for the geometric optimization of regions on submanifolds within a general ambient manifold  $M$ . We adopt the setting of Riemannian differential geometry, allowing us to leverage various classical results from this field. Under this framework, we provide proofs for a few well-known results in the classical shape optimization literature. Additionally, we delve into several methodologies for optimizing different geometries in diverse contexts. Throughout this chapter, we primarily employ the boundary variation method initially introduced by Hadamard in his seminal article [177]. This method has since been explored in various works such as [22, 116, 185, 184, 301]. In a Riemannian setting, we utilize the exponential mapping, which provides an equivalent representation of the “perturbation of identity” approach [184, 301]. This approach is one of the most widely used methods in the literature. Furthermore, this choice of representation can be seen as an analog to the “velocity method” [116, 354, 355], where a vector field  $\theta$  on  $\mathbb{R}^n$  induces a flow representing the trajectory of a particle evolving on the boundary. We will not delve into the details of these classical methods here, as they are extensively discussed in the cited works. It is worth noting that similar ideas to the ones presented in this chapter have been explored in [311], where the Hadamard structure theorem is proven on submanifolds, and crack surfaces are examined under this model. More recently, [285] has utilized equivalent ideas to those presented in this chapter for the computation of second-order directional shape derivatives of integrals on submanifolds. However, these works differ slightly from our framework in that they specifically consider submanifolds of the ambient space  $\mathbb{R}^n$ . In contrast, we consider the ambient space to be a complete Riemannian manifold  $M$ , which naturally leads to the use of the exponential mapping as an equivalent form of perturbation to the identity or velocity method. This choice allows us to leverage the well-developed and extensively studied theory of Riemannian differential geometry. Our selection of this theory is not unprecedented. For example, the study [188], which focuses on shape derivatives via differential forms, presents several elegant approaches and advantages. These include the coordinate-independent description of models and the clear separation of invariants under homeomorphic transformations. In our study, we have found that adopting a Riemannian viewpoint generally leads to more elegant proofs and offers a more intrinsic understanding of the geometric aspects of shape optimization. However, it comes at the cost of increased mathematical rigor due to the language of Riemannian differential geometry. Lastly, we note that a Riemannian perspective, approached differently, has already been explored in [287], where the set of shapes is viewed as a Riemannian manifold and  $W^{1,\infty}(\mathbb{R}^n, \mathbb{R}^n)$  is regarded as the tangent space. In general, the aim of this approach is to regard the set of all shapes as an abstract infinite-dimensional Riemannian manifold. This is distinct from our framework, which views shapes as submanifolds of an ambient manifold and seeks to deform them using the boundary variation method.

## Foreword on the theoretical content

Most of the results presented in this chapter are not directly utilized in the subsequent chapters, and therefore, it is not necessary to perform an in-depth reading to understand the rest of the work. However, we believe that the insights provided here offer a valuable alternative perspective on shape optimization, enriching the overall understanding of the subject and highlighting different approaches that may be of interest to those seeking a deeper exploration of the topic.

### 2.1 Shape differentiability on manifolds

In this section, we will designate the Riemannian manifold  $(M, g)$  as the ambient manifold. Our aim is to establish a notion of boundary variation for a regular domain  $G$  embedded in  $M$ . However, it is important to note that when extending the concept of boundary perturbation to a general manifold, the mapping  $x \mapsto (\text{Id} + \theta)(x)$  no longer represents a deformation in the ambient manifold. To address this issue, we must introduce the notion of geodesics, which induce a concept of “trajectory” on the manifold. This notion will be further defined in the subsequent discussion. Consequently, it becomes pertinent to consider the exponential mapping as a form of perturbation by a fixed vector field  $\theta \in \mathfrak{X}(M)$ . Furthermore, we examine a regular domain  $G \subset M$  with a closed codimension-1 submanifold  $\partial G$  as its boundary. Both  $G$  and  $\partial G$  possess induced metrics derived from the ambient metric.

#### 2.1.1 The deformation of space

For a shape  $G$  embedded in a Riemannian manifold, a vector field assigns tangent vectors to points on the shape’s surface. The exponential map translates these tangent vectors into new points on the manifold, effectively moving the shape in the direction of the vector field. This results in a smooth deformation that respects the manifold’s geometry. By adjusting the vector field and using the exponential map, various deformations can be explored to optimize the shape according to specific criteria, making it a crucial tool in shape optimization.

**Definition 2.1 (Deformation of space).** *Let  $(M, g)$  be a Riemannian manifold and a vector field  $\theta \in \mathfrak{X}(M)$ . The smooth map  $F_\theta^M : M \rightarrow M$  defined by:*

$$F_\theta^M(p) := \exp_p(\theta(p)) \quad (2.1)$$

*is referred to as the **deformation of space on  $M$  by  $\theta$** . We denote the **deformed domain** by  $G_\theta := F_\theta^M(G)$ .*

The following lemma states that on a compact Riemannian manifold, there exists a neighborhood around the zero section in the space of vector fields such that any vector field within this neighborhood induces a smooth, invertible map. This is crucial for shape optimization and deformation analysis as it ensures that small perturbations via vector fields result in well-behaved, reversible deformations. It provides stability for the exponential map, enabling smooth and reversible shape changes, which are essential for iterative optimization and geometric analysis.

**Lemma 2.1.** *Let  $(M, g)$  be a Riemannian compact manifold. Then, there exists a neighborhood  $\mathcal{O} \subset \mathfrak{X}(M)$  of the zero section  $0_M$  such that for any  $\theta \in \mathcal{O}$ , the smooth map  $F_\theta^M$  is a diffeomorphism.*

*Proof.* Note that since the manifold is compact, then its injectivity radius  $r = \text{inj}(M)$  is strictly positive. Let  $\mathcal{O} \subset \mathfrak{X}(M)$  be defined by:

$$\mathcal{O} := \{\theta \in \mathfrak{X}(M) \mid \forall k \in \mathbb{N}, \|\theta\|_k < r\},$$

which is clearly open in the induced  $C^\infty$  topology. Recall that for any  $p \in M$ ,  $\exp_p$  is a diffeomorphism from the ball  $B(0, r)$  in  $T_p M$  onto its image  $U \subset M$ , which is a normal neighborhood (see [Definition D.17](#)). Furthermore, for any  $v \in T_p M$ , the geodesic  $\gamma(t)$  with  $\gamma(0) = p$ ,  $\gamma'(0) = v$  lies entirely within  $U$ . Consider then any two distinct points  $p_1, p_2 \in M$  and notice that we can always choose normal neighborhoods  $U_1, U_2 \subset M$ , around these points such that they are the image of the balls  $B(0, \epsilon_i)$  with sufficiently small

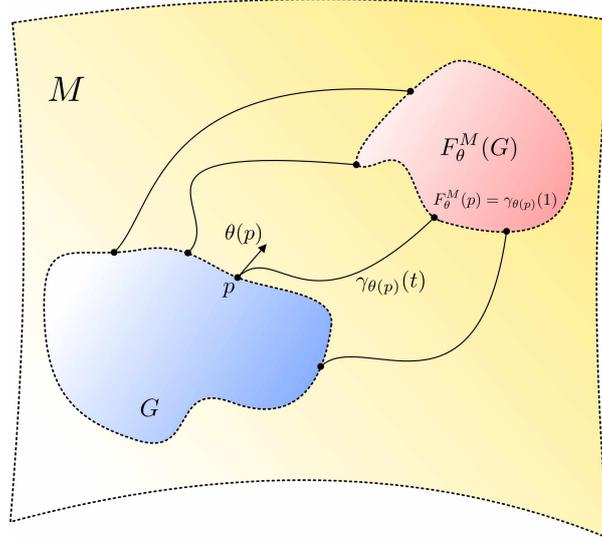


Figure 2.1: Illustration of the deformation of a regular domain  $G$  in a possibly curved manifold  $M$ . The deformation is governed by the map  $F_\theta^M$ , which smoothly transforms points in  $G$  to a deformed domain, tracking the path  $\gamma_{\theta(p)}(t)$  for each point  $p$ .

radii  $\epsilon_1, \epsilon_2 < r$ , and  $B(0, \epsilon_1) \cap B(0, \epsilon_2) = \emptyset$ . To see why  $F_\theta^M$  is injective, assume that  $q := F_\theta^M(p_1) = F_\theta^M(p_2)$ . Then there exist geodesics  $\gamma_i : [0, 1] \rightarrow U_i$  such that:

$$\gamma_i(0) = p_i, \quad \gamma_i'(0) = \theta(p_i), \quad q = \gamma_i(1),$$

for  $i = 1, 2$ , and so  $U_1$  and  $U_2$  forcibly overlap, which can only be the case if  $p_1 = p_2$ . ■

**Remark 2.1.** For  $\theta \in \mathcal{O}$ , note that since  $F_\theta^M$  is a diffeomorphism, the deformed boundary of  $G$  is equal to the boundary of the deformed domain  $G_\theta$ . In symbols:

$$\partial(G_\theta) = F_\theta^M(\partial G). \quad (2.2)$$

This allows us to fix the notation  $\partial G_\theta := \partial(G_\theta) = F_\theta^M(\partial G)$  which from now on we will utilize.

A key result of significant importance is the differential of the deformation in space. This differential can be implicitly described as the solution to the Jacobi equation [Theorem D.6](#) with suitable initial conditions. The Jacobi equation characterizes the infinitesimal behavior of  $F_\theta^M$  when varying the point  $p \in M$  in the direction  $v \in T_p M$ . Moreover, it establishes a connection between this perturbation and the manifold's curvature, specifically through the curvature endomorphism.

**Lemma 2.2 (Differential of the deformation).** Let  $\gamma_{\theta(p)} : I \rightarrow M$  be the geodesic such that  $\gamma_{\theta(p)}(0) = p$ ,  $\gamma_{\theta(p)}'(0) = \theta(p)$ . For all  $v \in T_p M$ , the differential  $d_p F_\theta^M : T_p M \rightarrow T_{F_\theta(p)} M$  of  $F_\theta^M$  at a point  $p \in M$  is given by:

$$d_p F_\theta^M(v) = J_v(1), \quad (2.3)$$

where  $J_v(t)$  satisfies the Jacobi equation:

$$D_t^2 J_v + R(J_v, \gamma_{\theta(p)}') \gamma_{\theta(p)}' = 0, \quad (2.4)$$

with initial conditions:

$$J_v(0) = v \quad \text{and} \quad D_t J_v(0) = \nabla_v \theta(p), \quad (2.5)$$

where  $R$  is the curvature endomorphism, and  $\nabla_v \theta$  is the covariant derivative of  $\theta$  in the direction  $v \in T_p M$ .

*Proof.* Let  $\gamma_{\theta(p)}(t) = \exp_p(t\theta(p))$  and choose a curve  $\sigma(s) : (-\epsilon, \epsilon) \rightarrow M$  satisfying  $\sigma(0) = p$ ,  $\sigma'(0) = v$  for  $v \in T_pM$ . Consider the one-parameter family of curves:

$$\Gamma : (-\epsilon, \epsilon) \times [0, 1] \rightarrow M \quad (2.6)$$

$$(s, t) \mapsto \exp_{\sigma(s)}(t\theta(\sigma(s))) . \quad (2.7)$$

Note  $F_{t\theta}^M(p) \equiv \Gamma(0, t)$ , we may compute the differential of  $F_{\theta}^M$  by computing  $\frac{\partial}{\partial s}\Gamma(0, 1)$ . Define the time varying vector field  $J(t) := \frac{\partial}{\partial s}\Gamma(0, t)$ ; we now show that  $J$  is a Jacobi field. Note that in particular we have that  $\Gamma(0, t) = \gamma_{\theta(p)}(t)$  showing that  $\Gamma$  is a variation of  $\gamma_{\theta(p)}$ . Moreover it is a variation through geodesics by [Definition 2.1](#), thus  $J$  is indeed a Jacobi field by [\(D.29\)](#). Furthermore, we have that:

$$J(0) = \frac{\partial}{\partial s}\Gamma(0, 0) = \sigma'(0) = v , \quad (2.8)$$

and utilizing the symmetry lemma ([Lemma D.2](#)), we have:

$$D_t J(0) = D_t \frac{\partial}{\partial s}\Gamma(0, 0) = D_s \frac{\partial}{\partial t}\Gamma(0, 0) = D_s \theta(p) = \nabla_v \theta(p) , \quad (2.9)$$

where  $D_t$  and  $D_s$  denote covariant differentiation through  $\gamma_{\theta(p)}(t)$  and  $\sigma(s)$  respectively. ■

**Remark 2.2 (Re-scaling of the geodesic path).** *From the proof it is clear that, in particular, the differential of the deformation  $F_{t\theta}(p)$  is equal to  $J_v(t)$  along the geodesic  $\gamma_{\theta(p)}$ .*

Let us discuss the significance of [Lemma 2.2](#). From a practical standpoint, this lemma enables computations involving the differential of  $F_{\theta}^M$ . Additionally, it offers a means to establish connections and reasoning about the varying geometries of  $G$  and  $F_{\theta}^M(G)$ . Its utilization becomes particularly relevant when quantifying infinitesimal variations in the boundary of  $G$ . In the subsequent section, we will explore how parallel transport serves as a vital link when analyzing this spatial deformation.

As in the Euclidean case, the definition of deformation space ([Definition 2.1](#)) naturally gives rise to the notion of shape differentiability with respect to a domain  $G$ . We can o

**Definition 2.2.** *The mapping  $G \mapsto J(G)$  is said to be **shape differentiable** at  $G$  if the underlying mapping*

$$\begin{aligned} \mathfrak{X}(M) &\rightarrow \mathbb{R} \\ \theta &\mapsto J(G_{\theta}) = J(F_{\theta}(G)) \end{aligned} \quad (2.10)$$

*is Gateaux differentiable at  $\theta = 0$ . In this case, the limit*

$$J'(G)(\theta) = \lim_{t \rightarrow 0} \frac{J(G_{t\theta}) - J(G)}{t} = \left. \frac{\partial}{\partial t} \right|_{t=0} J(F_{t\theta}(G)) \quad (2.11)$$

*is called the shape derivative of  $J$  at  $G$  in the direction  $\theta$ .*

## 2.1.2 Relation to parallel transport

This section aims to establish the connection between the concept of “parallel transport” and the deformation of a domain. To begin, let us provide a formal definition.

**Definition 2.3.** *Let  $(M, g)$  be a Riemannian manifold. We say that a smooth vector field  $\theta \in \mathfrak{X}(\gamma)$  along a smooth curve  $\gamma$  is said to be **parallel along  $\gamma$**  if  $D_t \theta \equiv 0$  (where  $D_t$  is covariant differentiation along  $\gamma(t)$ ).*

**Theorem 2.1 (Parallel transport).** *Suppose  $M$  is a smooth manifold with or without boundary. Given a smooth curve  $\gamma : I \rightarrow M$ ,  $t \in I$ , and a vector  $v \in T_{\gamma(t)}M$ , there exists a unique parallel vector field  $P$  along  $\gamma : I \rightarrow M$  such that  $P(t) = v$ . This vector field is called the **parallel transport map of  $v$  along  $\gamma$** . Additionally, for each  $t, s \in I$ , the map defined by:*

$$P_{t,s}^\gamma : T_{\gamma(t)}M \rightarrow T_{\gamma(s)}M \quad (2.12)$$

$$v \mapsto P(s) , \quad (2.13)$$

*is called the **parallel transport map**.*

Parallel transport offers a means to establish a connection between the geometries of two points, denoted as  $p$  and  $q$ , within the manifold  $M$ . This connection is achieved through the consideration of a curve  $\gamma : [a, b] \rightarrow M$ , where  $\gamma(a) = p$  and  $\gamma(b) = q$ . By employing the parallel transport map, we establish a relationship between the tangent spaces  $T_{\gamma(a)}M$  and  $T_{\gamma(b)}M$ . This connection becomes evident through the following definition.

**Definition 2.4.** *Let  $(b_1, \dots, b_n)$  be any basis for  $T_{\gamma(a)}$  and let  $(E_1, \dots, E_n)$  be the  $n$ -tuple of parallel vector fields obtained by parallel transporting each  $b_i$  along  $\gamma$ . Since the parallel transport map is an isomorphism, the vectors  $E_i(t)$  form a basis for  $T_{\gamma(t)}M$  at each point  $\gamma(t)$ . In this case,  $(E_1, \dots, E_n)$  is called a **parallel frame along  $\gamma$** .*

By utilizing the previous definitions and [Lemma 2.2](#), it becomes possible to analyze the differential of the deformation through the examination of the Jacobi field  $J(t)$  along the geodesic  $\gamma_{\theta(p)}$ , for a fixed vector field  $\theta \in \mathfrak{X}(M)$ . This discussion is not without precedence. In [\[322\]](#), the relationship between the exponential map and the Jacobian determinant has been established as a means to analyze volume distortion along geodesic paths. Additionally, this connection provides a consistent way to associate each element of the tangent space family  $\{T_{\gamma(t)}M\}_{t \in I}$  with  $\mathbb{R}^n$ .

**Theorem 2.2.** *Let  $p \in M$ ,  $U \subset M$  a normal neighborhood of  $p$ , let  $b := (b_1, \dots, b_n)$  an orthonormal basis for  $T_pM$  and  $E(t) := (E_1(t), \dots, E_n(t))$  the parallel orthonormal frame with  $E(0) = b$ . In these coordinates the differential of the deformation of space ([Lemma 2.2](#)) is a matrix path  $\mathbf{J} : I \rightarrow \mathbb{R}^{n \times n}$  in the basis  $E(t)$ , along the geodesic  $\gamma_{\theta(p)} : I \rightarrow M$ , and satisfies:*

$$\frac{d^2}{dt} \mathbf{J}(t) + \mathbf{R}(t) \mathbf{J}(t) = 0 \quad (2.14)$$

$$\mathbf{J}(0) = I_n \quad (2.15)$$

$$\frac{d}{dt} \mathbf{J}(0) = \nabla \theta(p) , \quad (2.16)$$

where  $I_n \in \mathbb{R}^{n \times n}$  is the identity matrix,  $\nabla \theta(p)$  is the Jacobian matrix of  $\theta$  at  $p$  in the basis  $b$ , and  $\mathbf{R}(t)$  is the matrix defined by:

$$\mathbf{R}_{i,j}(t) = \text{Riem}(\gamma'(t), E_i(t), \gamma'(t), E_j(t)) \quad 1 \leq i, j \leq n , \quad (2.17)$$

where  $\text{Riem}$  is the Riemann curvature tensor ([Definition D.21](#)).

*Proof.* Note that for every orthonormal basis  $b$  for  $T_pM$ , there is a unique normal coordinate chart  $(U, x^i)$  (cf. e.g. [\[219\]](#)). Moreover,  $b_i = \partial/\partial x^i|_p$ . Hence from now on let us work with the coordinate vectors for simplicity. Firstly, we write  $d_p F_{t\theta}^M(v) = J_v(t) = \sum_{i=1}^n J^i(t) E_i(t)$  and  $\gamma'_{\theta(p)}(t) = \sum_{i=1}^n \gamma^i(t) E_i(t)$  in terms of the frame  $E(t)$ . When expressed in this basis, we may consider the matrix  $\mathbf{J}(t)$  whose entries are given by:

$$\mathbf{J}_{i,j}(t) = g_{\gamma_{\theta(p)}(t)}(J^i(t) E_i(t), E_j(t)) , \quad 1 \leq i, j \leq n . \quad (2.18)$$

From [\(2.4\)](#), we have for all  $t \in I$ ,  $1 \leq j \leq n$  :

$$g_{\gamma_{\theta(p)}(t)} \left( D_t^2 \left( \sum_{i=1}^n J^i(t) E_i(t) \right), E_j(t) \right) + g_{\gamma_{\theta(p)}(t)} \left( R \left( \sum_{i=1}^n J^i(t) E_i(t), \gamma'_{\theta(p)} \right), \gamma'_{\theta(p)}, E_j(t) \right) = 0 .$$

Since each  $E_i$  is parallel ( $D_t E_i \equiv 0$ ), we have:

$$D_t^2 J_v(t) = \sum_{i=1}^n \frac{d^2 J^i}{dt^2}(t) E_i(t) . \quad (2.19)$$

Moreover by the multilinearity of the tensors  $R$  and Riem, we have:

$$\sum_{i=1}^n \text{Riem} \left( \gamma'_{\theta(p)}, E_i(t), \gamma'_{\theta(p)}, E_j(t) \right) J^i(t) = g_{\gamma_{\theta(p)}(t)} \left( R \left( \sum_{i=1}^n J^i(t) E_i(t), \gamma'_{\theta(p)} \right) \gamma'_{\theta(p)}, E_j(t) \right) . \quad (2.20)$$

Both of the equations above give us the first part of (2.14). We can recover the first initial condition by noting that

$$J_v(0) = v = \sum_{i=1}^n v^i \frac{\partial}{\partial x_i} \Big|_p , \quad (2.21)$$

which implies directly  $\mathbf{J}(0) = I_n$ . Lastly, writing the covariant derivative in terms of the coordinate vectors, we can see:

$$D_t J_v(0) = \nabla_v \theta(p) \quad (2.22)$$

$$= \sum_{k=1}^n \left( \sum_{i,j=1}^n v^j \theta^i(p) \Gamma_{ij}^k(p) + \sum_{j=1}^n v^j \frac{\partial \theta^k}{\partial x^j} \right) \frac{\partial}{\partial x_k} \Big|_p \quad (2.23)$$

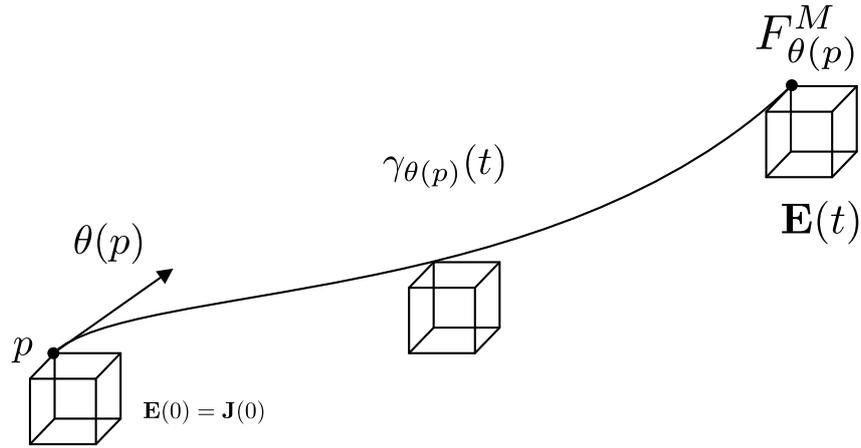
$$= \sum_{k=1}^n \left( \sum_{j=1}^n v^j \frac{\partial \theta^k}{\partial x^j} \right) \frac{\partial}{\partial x_k} \Big|_p , \quad (2.24)$$

where the last equality is obtained from the fact that Christoffel symbols  $\Gamma_{ijk}$  vanish at  $p$  by properties of normal coordinates. Thus:

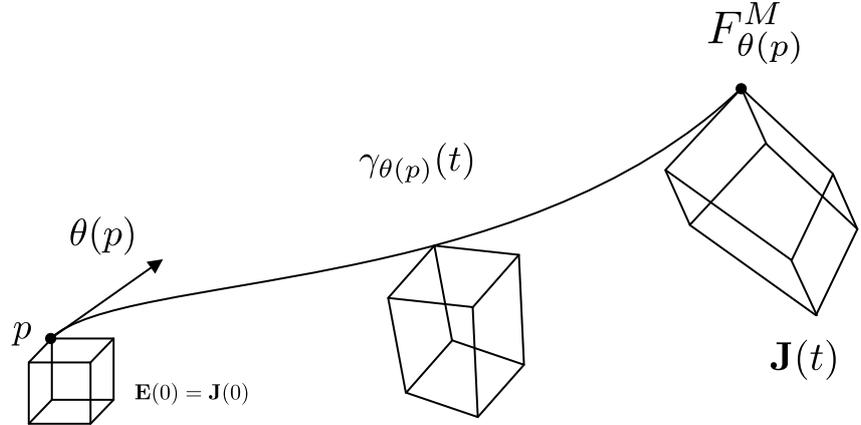
$$\frac{d}{dt} \mathbf{J}(0) = \nabla \theta(x) , \quad (2.25)$$

where  $\nabla \theta(x)$  denotes the Jacobian matrix of  $\theta$  at  $x(p)$  in the basis  $b$ . ■

Apart from establishing the connection with parallel transport, **Theorem 2.2** also provides a framework for reasoning about the differential of the deformation within a normal neighborhood of each point  $p \in M$ . This is particularly valuable when computing derivatives, as it offers insights into the volume distortion. In practical terms, as we will observe, this volume distortion corresponds to the pullback  $dV_{F_\theta^*g}$  of the volume form  $dV_g$  by  $F_\theta$ . To illustrate this concept, we can associate the parallel orthonormal frame  $E(t)$  with a matrix path  $\mathbf{E}(t)$  in  $\mathbb{R}^{n \times n}$ . Consequently, both  $\mathbf{E}(t)$  and  $\mathbf{J}(t)$  can be visualized as parallelograms along the geodesic  $\gamma_{\theta(p)}(t)$ . This is depicted in **Fig. 2.2**. While the volume of  $\mathbf{E}(t)$  remains constant, the effect of curvature distorts this volume, resulting in  $\mathbf{J}(t)$ .



(a) Representation of the parallel orthonormal frame  $E(t)$  when identified with a matrix path  $\mathbf{E}(t)$ .



(b) Representation of the parallel orthonormal frame  $J(t)$  when identified with a matrix path  $\mathbf{J}(t)$ , solution to (2.14).

Figure 2.2: Comparison of volume distortion along a geodesic between the parallel orthonormal frame  $E(t)$  and the Jacobi field  $J(t)$ , both identified with matrix paths, coinciding at  $t = 0$  but potentially diverging at later times, illustrating the impact of curvature on volume distortion.

## 2.2 Shape representation via the signed distance function

The concept of the “signed distance function” to a regular domain  $G$  in the manifold  $M$  finds a wide range of applications across various fields. It is commonly used in general level set methods [293, 256], as well as in collision detection [156] and real-time rendering applications [5]. In the context of shape and topology optimization, signed distance functions play a significant role due to their ability to precisely and conveniently represent the shape being optimized. This is particularly relevant in works related to shape and topology optimization, such as [12, 14, 142]. In this section, we revisit key results concerning the signed distance function from the point of view of Riemannian geometry. The work of [219] serves as a reference for these results. To begin, we provide the definition of the signed distance function.

**Definition 2.5 (Signed distance function).** *Let  $(M, g)$  be a Riemannian manifold and  $G$  a regular domain of  $M$  with closed boundary  $\partial G$ . The **signed distance function to  $G$  on  $M$**  is the function  $d_G^M : M \rightarrow \mathbb{R}$  defined by:*

$$\forall x \in M, d_G^M(x) = \begin{cases} -d^M(x, \partial G) & \text{if } x \in G \\ 0 & \text{if } x \in \partial G \\ d^M(x, \partial G) & \text{if } x \in M \setminus \overline{G} \end{cases}, \quad (2.26)$$

where  $d^M(x, \partial G)$  denotes the Riemannian distance from a point  $x \in M$  to the submanifold  $\partial G$  (see Definition D.23).

We revisit some well known properties regarding the signed distance function.

**Proposition 2.1 (Properties of the signed distance function).** *Let  $(M, g)$  a Riemannian manifold and  $G$  a regular domain in  $M$ . Let  $U$  be a tubular neighborhood around  $\partial G$  in  $M$ . Then:*

1.  $p \mapsto d_G^M(p)$  is a smooth function on  $U$ .
2. The signed distance function  $d_G^M : U \rightarrow \mathbb{R}$  satisfies:

$$\forall p \in U, \quad |\nabla_M d_G^M(p)| = 1. \quad (2.27)$$

3. The gradient of the signed distance function is given by:

$$\nabla_M d_G^M(p) = \begin{cases} n_{\partial G}(p) & \text{if } p \in \partial G, \\ -\frac{\log_p(\pi_{\partial G}(p))}{d_G^M(p)} & \text{if } p \in U \setminus \overline{\partial G}. \end{cases} \quad (2.28)$$

The concept of the projection of a point  $q$  onto a submanifold is closely intertwined with the signed distance function, and appears commonly in shape derivative expressions on the surface.

**Definition 2.6 (Projection onto a submanifold).** *Let  $\Sigma$  be a submanifold in  $M$ . Let  $q \in M$  and define  $\Pi_\Sigma^M(q)$  as the **set of projection onto  $\Sigma$**  given by:*

$$\Pi_\Sigma^M(q) := \{p \in M : d^M(q, p) = d^M(q, \Sigma)\}. \quad (2.29)$$

Whenever  $\Pi_\Sigma^M(q) = \{p\}$  is a singleton, we denote the point  $p$  as the **projection  $\pi_\Sigma(q)$  of a point  $q \in M$  onto  $\Sigma$** .

**Theorem 2.3.** *Let  $G$  be a regular domain of  $M$  with boundary  $\partial G$ . Let  $\theta(p) := n_\Sigma$ . Then there exists  $\delta > 0$  and a tubular neighborhood  $U_\delta$  around  $\partial G$  such that the mapping:*

$$\begin{aligned} (-\delta, \delta) \times \partial G &\rightarrow U_\delta \\ (t, p) &\mapsto F_{t\theta}(p) \end{aligned} \quad (2.30)$$

is a smooth diffeomorphism. For any  $q = F_{t\theta}^M(p) \in U_\delta$  one has:

$$t = d_G^M(p), \quad \text{and } p = \pi_{\partial G}(q). \quad (2.31)$$

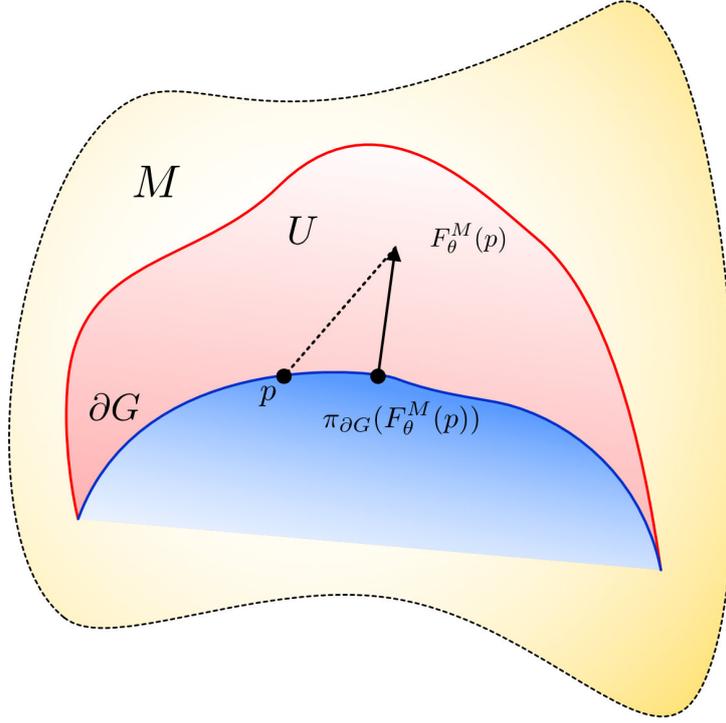


Figure 2.3: The shaded region represents a tubular neighborhood  $U \subset M$  around the boundary  $\partial G$  of a regular domain  $G$ . The solid black line arrow marks the projection of a point  $F_\theta^M(p)$  onto the boundary  $\partial G$  and the solid blue line corresponds to  $\partial G$ .

*Proof.* Since  $\partial G$  is a compact embedded submanifold of  $M$ , by the tubular neighborhood theorem, there exists a uniform tubular neighborhood  $U_\delta$  in  $M$  around  $\partial G$ . This neighborhood is the diffeomorphic image under the exponential map  $\exp|_{N(\partial G)}$  restricted to the normal bundle  $N(\partial G)$  of some subset  $V_\delta \subset N(\partial G)$ , defined by:

$$\begin{aligned} V_\delta &= \{v_p \in N(\partial G) : |v_p| < \delta\} \\ &= \{tn_{\partial G}(p) \in N_p(\partial G) : p \in \partial G, |t| < \delta\} \end{aligned} \quad (2.32)$$

for some constant  $\delta > 0$ . By the definition of tubular neighborhood, we have that there exists a unique element  $v_p \in N(\partial G)$  such that  $q = \exp|_{N(\partial G)}(v_p)$ . Now choose any normal coordinates  $x$  on  $U$  and write  $v_p = \sum_{i=1}^n v^i \frac{\partial}{\partial x_i}|_p$  in terms of the coordinate vectors. Notice that the curve,

$$t \mapsto \gamma(t) := F_{t\theta}(p) = \exp_p(tv_p)$$

is a geodesic starting at  $p \in \partial G$  with initial velocity  $v_p \in N_p(\partial G)$  and  $\gamma(0) = p$ ,  $\gamma(1) = q$ . This fact coupled with [Proposition D.3](#) implies that  $q$  has the coordinate expression  $x(q) = (v_p^1, \dots, v_p^n)$ . In these coordinates, the distance function  $d(q, \partial G)$  satisfies ([Proposition 6.37](#) of [\[219\]](#)):

$$\forall q \in U_\delta, \quad d^M(q, \partial G) = |v_p|, \quad (2.33)$$

whence the following relation:

$$\forall q \in U_\delta, \exists! p \in \partial G \quad \text{s.t.} \quad v_p = d_G^M(q)n_{\partial G}(p). \quad (2.34)$$

Since  $d^M(q, \partial G) = d^M(q, p)$ , by [Definition 2.6](#), we have that  $p = \pi_{\partial G}(q)$ . ■

**Remark 2.3.** *Essentially, the previous theorem states the projection onto the boundary of  $G$  and the signed distance function are always well in defined some tubular neighborhood  $U_\delta$  of radius  $\delta > 0$ .*

The shape derivative of the signed distance function is essential for analyzing shape variations and plays a key role in the proof of the Hadamard structure theorem. It simplifies the treatment of boundary deformations and is crucial for deriving the optimality conditions that characterize the geometric structure of optimal shapes. Next, we will demonstrate how to calculate it.

**Lemma 2.3.** *Let  $G$  be a regular domain of  $M$  with boundary  $\partial G$ . Define  $L : \mathcal{O}_\delta \rightarrow \mathbb{R}$  by:*

$$\forall p \in U_\delta \setminus \overline{\partial G}, \quad L(\theta)(p) := d_{G_\theta}^M(F_\theta^M(p)), \quad (2.35)$$

where  $O_\delta \subset \mathfrak{X}(M)$  as in the start of the proof of [Theorem 2.3](#). Then  $D$  is Gateaux differentiable at  $\theta = 0$  and its derivative reads:

$$\forall p \in U_\delta \setminus \overline{\partial G}, \quad L'(0)(\theta)(p) = \theta(p) \cdot \nabla_M d_G^M(p) - \theta(\pi_{\partial G}(p)) \cdot n_{\partial G}(\pi_{\partial G}(p)). \quad (2.36)$$

*Proof.* For simplicity, assume that  $q \in M \setminus \overline{U_\delta}$ , the complementary case being analogous. From [Definition D.23](#), one has:

$$\begin{aligned} L(\theta)(q) &= \inf_{p \in \partial G} \left\{ \inf_{\substack{\gamma: [a,b] \rightarrow M_\theta \\ \gamma(a)=F_\theta^M(p), \gamma(b)=F_\theta^M(q)}} \int_a^b |\gamma'(t)| dt \right\} \\ &= \inf_{p \in \partial G} \left\{ \inf_{\substack{\sigma: [a,b] \rightarrow M \\ \gamma(a)=p, \gamma(b)=q}} \int_a^b \left| \frac{d}{dt} (F_\theta^M(\sigma(t))) \right| dt \right\} \\ &= \inf_{p \in \partial G} \left\{ \inf_{\substack{\sigma: [a,b] \rightarrow M \\ \gamma(a)=p, \gamma(b)=q}} \int_a^b |J_{\sigma'(t)}(1)| dt \right\}, \end{aligned} \quad (2.37)$$

where the last line is obtained via [Lemma 2.2](#), so that  $J'_\sigma(t)$  is the Jacobi field along the geodesic  $\eta(t)$  satisfying:

$$\begin{aligned} \eta(0) &= p, & \eta'(0) &= \theta(p), \\ J_{\sigma'(t)}(0) &= \sigma'(t), & D_t J(0) &= \nabla_{\eta'(t)} \theta(p). \end{aligned} \quad (2.38)$$

Clearly, for  $\theta = 0$ , we have  $J_{\sigma'(t)}(1) = J_{\sigma'(t)}(0) = \sigma'(t)$ , and clearly the infimum is attained by the unit-speed geodesic connecting  $p = \pi_{\partial G}(q)$  to  $q$ , with  $a = 0$ ,  $b = d_G^M(q)$ . [Theorem B.2](#) entails that  $D$  is differentiable at  $\theta = 0$  and:

$$\begin{aligned} L'(\theta)(0) &= \int_a^b \frac{1}{|\sigma'(t)|} D_s J_{\sigma'(t)}(0) dt \\ &= \int_0^{d_G^M(q)} \nabla_{\sigma'(t)} \theta(\sigma(t)) \cdot \sigma'(t) dt \\ &= \int_0^{d_G^M(q)} D_t \theta(t) \cdot \sigma'(t) dt \end{aligned} \quad (2.39)$$

where we go from the first to second line by [Remark 2.2](#), the fact that  $\sigma(t)$  has unit speed and the definition  $\theta(t) := \theta(\sigma(t))$ . The third line is just using the definition of covariant derivative along the curve  $\sigma(t)$ . Note that:

$$D_t(\theta(t) \cdot \sigma'(t)) = D_t \theta(t) \cdot \sigma'(t) + \theta(t) \cdot D_t \sigma'(t) = D_t \theta(t) \cdot \sigma'(t), \quad (2.40)$$

since  $D_t \sigma'(t) = 0$  by [Definition D.14](#). Hence, applying the fundamental theorem of calculus:

$$\begin{aligned} L'(\theta)(0) &= \int_0^{d_G^M(q)} \frac{d}{dt} (\theta(t) \cdot \sigma'(t)) dt \\ &= \theta(d_G^M(q)) \cdot \sigma'(d_G^M(q)) - \theta(0) \cdot \sigma'(0) \\ &= \theta(q) \cdot \nabla_M d_G^M(q) - \theta(\pi_{\partial G}(q)) \cdot n_{\partial G}(\pi_{\partial G}(q)) \end{aligned} \quad (2.41)$$

The last line follows from [Proposition 2.1](#). ■

**Remark 2.4.** The formula (2.36) is of ‘Lagrangian’ nature: for given  $p \in M$ , the derivative of the distance function to the perturbed set  $G_\theta$  at the perturbed point  $F_\theta^M(p)$  is calculated. Using the more classical notation  $\dot{d}_G^M(\theta) = L'(0)(\theta)$  for this derivative, the corresponding Eulerian derivative  $(d_G^M)'(\theta)$  of  $d_G^M$  is then defined by the formula:

$$\begin{aligned} (d_G^M)'(\theta)(p) &:= \dot{d}_G^M(\theta)(p) - \nabla_M d_G^M(p) \cdot \theta(p) \\ &= -\theta(\pi_{\partial G}(p)) \cdot n_{\partial G}(\pi_{\partial G}(p)). \end{aligned}$$

We shall give more details on the difference between Lagrangian and Eulerian derivatives in the next sections.

## 2.3 Generalities of shape derivatives on manifolds

In this section, we provide a broad overview of the key concepts and mathematical tools necessary for understanding and working with shape derivatives on manifolds. We will introduce the fundamental concepts, relevant mathematical tools, and provide an overview of the key results that are pivotal for understanding and computing shape derivatives in manifold settings.

### 2.3.1 Structure of shape derivatives for a regular domain in the ambient space

In shape optimization, Hadamard’s structure theorem reveals that the derivative of a shape functional depends solely on how the domain’s boundary moves in the normal direction. Tangential movements along the boundary have no impact on the derivative. We are now poised to extend and prove this powerful theorem in the context of Riemannian manifolds.

**Theorem 2.4 (Hadamard’s structure theorem).** Let  $G \subset M$  be a regular domain with closed boundary  $\partial G$ , let  $n_{\partial G}$  denote the outward-pointing unit normal vector field along  $\partial G$ , and let  $J(G)$  a shape differentiable functional with shape derivative  $J'(G)(\theta)$  for  $\theta \in \mathfrak{X}(M)$ .

1. There exists a tubular neighborhood  $U_\delta$  of radius  $\delta > 0$  and a continuous linear form  $\ell_\delta : C^\infty(U_\delta) \rightarrow \mathbb{R}$  such that:

$$J'(G)(\theta) = \ell_\delta((\theta \cdot n_{\partial G}) \circ \pi_{\partial G}). \quad (2.42)$$

2. If  $\theta_1, \theta_2 \in \mathfrak{X}(M)$  are two vector fields such that  $\theta_1 \cdot n_{\partial G} = \theta_2 \cdot n_{\partial G}$ , then:

$$J'(G)(\theta_1) = J'(G)(\theta_2). \quad (2.43)$$

*Proof.* Define a neighborhood  $\mathcal{O}_\delta \subset \mathfrak{X}(M)$  by:

$$\mathcal{O}_\delta := \{\theta \in \mathfrak{X}(M) : \forall p \in \partial G, F_\theta(p) \in U_\delta\} \subset \mathcal{O}, \quad (2.44)$$

where  $\mathcal{O}$  is the neighborhood in Lemma 2.1, and  $U_\delta$  is a sufficiently small tubular neighborhood of radius  $\delta > 0$ . Such a tubular neighborhood exists by Theorem 2.3. We seek to prove that, for a judiciously chosen function  $\mathcal{F}$ , we can express the underlying map  $\mathcal{E}(\theta) := J(G_\theta)$  as follows:

$$\mathcal{E}(\theta) = \mathcal{F}(-d_{G_\theta}^M), \quad (2.45)$$

where  $d_{G_\theta}^M$  is the signed distance function to  $G_\theta$ . We can assume that our domain  $G$  is connected; otherwise conduct the same analysis on the individual components. We note that whenever  $\theta_1, \theta_2 \in \mathcal{O}_\delta$ , the following implication holds true:

$$\partial G_{\theta_1} = \partial G_{\theta_2} \implies G_{\theta_1} = G_{\theta_2}. \quad (2.46)$$

Indeed, consider  $\Phi := (F_{\theta_2})^{-1} \circ (F_{\theta_1})$  and suppose  $\partial G_{\theta_1} = \partial G_{\theta_2}$ , then:

$$\Phi(\partial G) = F_{\theta_2}^{M-1}(\partial G_{\theta_1}) = F_{\theta_2}^{M-1}(\partial G_{\theta_2}) = \partial G, \quad (2.47)$$

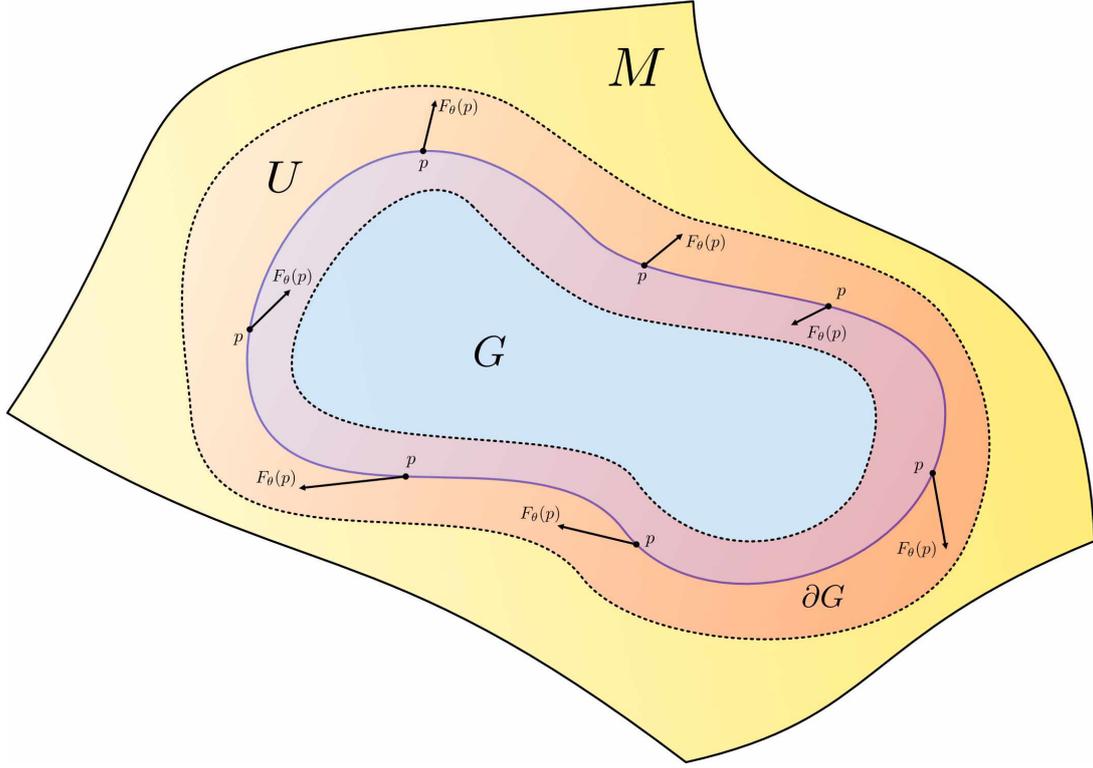


Figure 2.4: The shaded region shows a tubular neighborhood  $U \subset M$  around the boundary  $\partial G$  of a regular domain  $G$ . The solid blue line denotes  $\partial G$ .

and by connexity  $\Phi(G) = G$ . Now let  $\widetilde{d_{G_\theta}^M n_{\partial G}} \in \mathfrak{X}(M)$  be an extension of  $d_{G_\theta}^M n_{\partial G}$  to all of  $U$  (such an extension always exists since  $\partial G$  is closed). At the same time note that, by the previous reasoning, the choice of extension does not affect the following calculations. Define the function  $\mathcal{F} : C^\infty(U) \rightarrow \mathbb{R}$  by:

$$\forall \psi \in C^\infty(U), \quad \mathcal{F}(\psi) := J(F_{\widetilde{\psi n_{\partial G}}}^M(G)). \quad (2.48)$$

Note that for all  $p \in \partial G$ ,  $q := F_\theta^M(p)$ , we have:

$$-d_{G_\theta}^M(\pi_{\partial G}(q)) = d_G^M(q). \quad (2.49)$$

Since  $\theta \in \mathcal{O}_U$ ,  $F_\theta^M$  is a smooth bijection, and by **Theorem 2.3**, we have that for all  $p \in \partial G$ , then

$$F_\theta^M(p) = \exp_{\pi_{\partial G}(q)} \Big|_{N(\partial G)} (d_G^M(q) n_{\partial G}(\pi_{\partial G}(q))) \quad (2.50)$$

$$= \exp_{\pi_{\partial G}(q)} \Big|_{N(\partial G)} (-d_{G_\theta}^M(\pi_{\partial G}(q)) n_{\partial G}(\pi_{\partial G}(q))) \quad (2.51)$$

$$= F_{-d_{G_\theta}^M n_{\partial G}}^M(\pi_{\partial G}(F_\theta^M(p))), \quad (2.52)$$

Hence,

$$\partial G_\theta = F_\theta^M(\partial G) = F_{-d_{G_\theta}^M n_{\partial G}}^M(\pi_{\partial G}(\partial G)) = F_{-d_{G_\theta}^M n_{\partial G}}^M(\partial G) \implies G_\theta = F_{-d_{G_\theta}^M n_{\partial G}}^M(G), \quad (2.53)$$

whence,

$$\forall \theta \in \mathcal{O}_\delta, \quad \mathcal{E}(\theta) = J \left( F_{-d_{G_\theta}^M n_{\partial G}}^M(G) \right) = \mathcal{F}(-d_{G_\theta}^M). \quad (2.54)$$

Taking the Gateaux derivative, coupled with **Lemma 2.3**, we are able to deduce:

$$\mathcal{E}'(0)(\xi) = \mathcal{F}'(0)(-\theta \mapsto d_{G_\theta}^M)'(0)(\xi) = \mathcal{F}'(0)((\xi \cdot n_{\partial G}) \circ \pi_{\partial G}), \quad (2.55)$$

where the last equality is obtained by the first part of the proof. Finally, since  $\ell_\delta := \mathcal{F}(0)$  is a continuous linear form on  $\mathfrak{X}(U)$  by definition, we can conclude:

$$J'(G)(\theta_2) - J'(G)(\theta_1) = \ell_\delta(\theta_2 \cdot n_{\partial G}) - \ell_\delta(\theta_1 \cdot n_{\partial G}) = \ell_\delta((\theta_2 - \theta_1) \cdot n_{\partial G}) = 0. \quad (2.56)$$

■

**Lemma 2.4.** *There exists a continuous linear form  $\ell : C^\infty(\partial G) \rightarrow \mathbb{R}$  such that:*

$$J'(G)(\theta) = \ell(\theta \cdot n_{\partial G}). \quad (2.57)$$

*Proof.* Define the restriction of the projection  $\pi_{\partial G}^\delta : U_\delta \rightarrow \partial G$  as:

$$\pi_{\partial G}^\delta := \pi_{\partial G}|_{U_\delta}, \quad (2.58)$$

and note that, since we are in the tubular neighborhood, the distance from the projection to the point is less than  $\delta$ :

$$\forall q \in U_\delta, \quad d(q, \partial G) = d(\pi_{\partial G}(q), q) \leq \delta. \quad (2.59)$$

Thus, as  $\delta \rightarrow 0$ , we have that  $d(q, \partial G) \rightarrow 0$  and thus there exists a sequence  $\{\pi_{\partial G}^\delta\}$  such that  $\pi_{\partial G}^\delta \rightarrow \text{Id}$ , with  $\text{Id}$  the identity mapping on  $\partial G$ . Since the space of smooth functions  $C^\infty(U_\delta)$  is a Fréchet space, it is by definition complete. Since  $\ell_\delta$  is a continuous linear functional for each sufficiently small  $\delta > 0$ , there exists some continuous linear functional  $\ell$  on  $C^\infty(\partial G)$  such that it is the limit point of  $\ell_\delta$  as  $\delta \rightarrow 0$ .  $\blacksquare$

The validity of the Hadamard structure theorem within this framework of shape derivative is expected. The proof follows a similar approach to that presented in [184], where an appropriate linear functional is constructed and the implicit function theorem is applied. In our proof, we also construct a suitable functional that leads to the desired result. However, instead of using the implicit function theorem, we employ the tubular neighborhood theorem. The key geometric insight is that for  $p \in M$  and sufficiently small  $\theta \in \mathfrak{X}(M)$ , the transported points  $F_\theta^M(p)$  lie within a tubular neighborhood of  $\partial G$ . By definition, this tubular neighborhood is the diffeomorphic image of an open subset in the normal bundle of the boundary under the exponential mapping. This concept is illustrated in Fig. 2.4.

### 2.3.2 Structure of shape derivatives for a regular domain in a hypersurface of the ambient manifold

Utilizing the technique shown in the proof of Theorem 2.4 one can obtain a similar result for the structure of a domain.

**Lemma 2.5.** *Let  $S$  denote a hypersurface in  $M$  and denote its second fundamental form by  $\mathbb{II}$ . Let  $G$  be a regular domain of  $S$  with boundary  $\partial G$ . Define  $L : \mathcal{O}_\delta \rightarrow \mathbb{R}$  by:*

$$\forall p \in U_\delta \setminus \overline{\partial G}, \quad L(\theta)(p) := d_{G_\theta}^S(F_\theta^M(p)), \quad (2.60)$$

where  $\mathcal{O}_\delta \subset \mathfrak{X}(M)$  as in the start of the proof of Theorem 2.3. Then  $L$  is Gateaux differentiable at  $\theta = 0$  and its derivative reads:

$$\begin{aligned} \forall p \in U_\delta \setminus \overline{\partial G}, \quad L'(0)(\theta)(p) &= \theta(p) \cdot \nabla_S d_G^S(p) - \theta(\pi_{\partial G}(p)) \cdot n_{\partial G}(\pi_{\partial G}(p)) \\ &\quad - \int_0^{d_G^S(p)} \theta(\sigma(t)) \cdot \mathbb{II}(\sigma'(t), \sigma'(t)) dt, \end{aligned} \quad (2.61)$$

where  $\sigma(t)$  denotes the geodesic on  $S$  connecting  $\pi_{\partial G}(p)$  to  $p$ .

*Proof.* Note that we can follow exactly the same steps as those in the proof of Lemma 2.3 to deduce that:

$$L'(\theta)(0) = \int_0^{d_G^S(q)} D_t^M \theta(t) \cdot \sigma'(t) dt, \quad (2.62)$$

where  $D_t^M$  denotes covariant differentiation in  $M$ , with  $\theta(t) := \theta(\sigma(t))$ . The difference between this case and the other one, is that  $\sigma(t)$  need not be a geodesic in  $M$ , thus the equality  $D_t^M \sigma'(t) = 0$  is not true in general. We can, however, apply the Leibniz integral rule to obtain:

$$\begin{aligned} L'(\theta)(0) &= \theta(d_G^M(q)) \cdot \sigma'(d_G^M(q)) - \theta(0) \cdot \sigma'(0) + \int_0^{d_G^S(q)} \theta(t) \cdot D_t^M \sigma'(t) dt \\ &= \theta(q) \cdot \nabla_S d_G^S(p) - \theta(0) \cdot \sigma'(0) - \int_0^{d_G^S(q)} \theta(t) \cdot \mathbb{II}(\sigma'(t), \sigma'(t)) dt, \end{aligned}$$

where the last line follows from the identity:

$$D_t^S \sigma'(t) = D_t^M \sigma'(t) + \mathbb{I}(\sigma'(t), \sigma'(t)), \quad (2.63)$$

and the fact that  $D_t^S \sigma'(t) = 0$  since  $\sigma(t)$  is a geodesic on  $S$ . ■

Utilizing the previous result, one can prove the following proposition. Its proof omitted since it is analogous to that of [Theorem 2.4](#).

**Proposition 2.2.** *Let  $S$  denote a hypersurface contained in  $M$  and let  $G \subset S$  be a regular domain with closed boundary  $\partial G$ , let  $n_{\partial G}$  denote the outward-pointing unit normal vector field along  $\partial G$ , and let  $J(G)$  a shape differentiable functional with shape derivative  $J'(G)(\theta)$  for  $\theta \in \mathfrak{X}(M)$ . Then, there exists a tubular neighborhood  $U_\delta \subset M$  of radius  $\delta > 0$  and a continuous linear form  $\ell_\delta : C^\infty(U_\delta) \rightarrow \mathbb{R}$  such that:*

$$J'(G)(\theta) = \ell_\delta \left( p \mapsto \theta(\pi_{\partial G}(p)) \cdot n_{\partial G}(\pi_{\partial G}(p)) + \int_0^{d_G^S(p)} \theta(\sigma(t)) \cdot \mathbb{I}(\sigma'(t), \sigma'(t)) dt \right), \quad (2.64)$$

where  $\sigma(t)$  denotes the geodesic in  $S$ , connecting  $p$  to  $\pi_{\partial G}(p)$ .

This behavior is confirmed and detailed in [Chapter 4](#), where we set  $M = \mathbb{R}^d$  and  $S = \partial\Omega$  for a regular domain  $\Omega \subset M$ . A significant implication of this theorem is that for vector fields  $\theta \in \mathfrak{X}(\mathbb{R}^d)$  satisfying  $\theta \cdot n_{\partial\Omega} = 0$ , the structure can be expressed as:

$$J'(G)(\theta) = \ell_\delta(\theta \cdot n_{\partial G}), \quad (2.65)$$

because  $\mathbb{I}$  is also oriented in the normal direction. This indicates that the final shape derivative depends solely on the normal to the domain  $G$  and not on  $n_{\partial\Omega}$ .

### 2.3.3 Computation of shape derivatives

With a solid understanding of the structure of shape derivatives, we can proceed to their calculation. Typically, it is more convenient to work with a fixed or static domain rather than a deformed one. As a result, many proofs in this context begin with a change of variables. Following this approach, we present the version of the change of variables theorem that will be employed in our calculations. Additionally, it is crucial to consider that the following theorem incorporates the volume distortion caused by the diffeomorphism  $\phi$ , which is reflected in the pullback term  $dV_{\phi^*g}$ . This term accounts for the impact of shape deformation on the volume element.

**Theorem 2.5.** *Consider  $N$  a smooth manifold and  $\phi : N \rightarrow M$  a diffeomorphism. Then for any smooth function  $f : M \rightarrow \mathbb{R}$  we have:*

$$\int_M f dV_g = \int_N f \circ \phi dV_{\phi^*g}, \quad (2.66)$$

where  $\phi^*g$  is the pullback of  $g$  by  $\phi$ .

**Theorem 2.6.** *Let  $f : M \rightarrow \mathbb{R}$  a smooth function and define the shape functional  $J$  by*

$$J(G) := \int_G f dV_g. \quad (2.67)$$

*Then  $J$  is shape differentiable at any  $G$  and*

$$J'(G)(\theta) = \int_{\partial G} f\theta \cdot N dV_{\hat{g}}, \quad (2.68)$$

where  $\hat{g}$  is the induced metric on  $\partial G$ .

*Proof.* By a change of variables we have:

$$J(G_{t\theta}) = \int_{G_{t\theta}} f \, dV_g = \int_G f \circ F_{t\theta}^M \, dV_{F_{t\theta}^*g}, \quad (2.69)$$

where  $F_{t\theta}^*g$  is the pullback of  $g$  by  $F_{t\theta}^M$ . Let  $\{U_\alpha\}_{\alpha \in A}$  be an indexed open cover of  $G$  with associated charts  $\{x_\alpha\}_{\alpha \in A}$ , and  $\{\psi_\alpha\}_{\alpha \in A}$  a smooth partition of unity subordinate to this cover. Then:

$$\int_G f \circ F_{t\theta} \, dV_{F_{t\theta}^*g} = \sum_\alpha \int_{x_\alpha(U_\alpha)} f \circ F_{t\theta} \circ x_\alpha^{-1} \sqrt{|\det (F_{t\theta}^*g)_{ij}| \circ x_\alpha^{-1}} \psi_\alpha \, dx. \quad (2.70)$$

Shrinking  $U_\alpha$  if necessary, we may assume that it is a normal neighborhood. From [Theorem 2.2](#) we know that, for a parallel orthonormal frame  $E(t) := (E_1(t), \dots, E_n(t))$ , the matrix components of the pullback metric can be expressed in terms of the Jacobi field  $J_v(t)$  along  $\gamma_{\theta(p)}$ :

$$(F_{t\theta}^*g)_{ij}(p) = g_{\gamma_{\theta(p)}(t)}(J_{E_i(t)|_p}(t), J_{E_j(t)|_p}(t)) \quad (2.71)$$

$$= g_{\gamma_{\theta(p)}(t)} \left( \sum_{k=1}^n J_{E_i(t)|_p}^k E_k(t), \sum_{k=1}^n J_{E_j(t)|_p}^k E_k(t) \right) \quad (2.72)$$

$$= g_{\gamma_{\theta(p)}(t)}(J^i E_i(t), J^j E_j(t)), \quad (2.73)$$

for  $1 \leq i, j \leq n$ . Whence:

$$\sqrt{|\det (F_{t\theta}^*g)_{ij}| \circ x_\alpha^{-1}} = \sqrt{|\det (\mathbf{J}^T(t)\mathbf{J}(t))|}, \quad (2.74)$$

where  $\mathbf{J}(t)$  is the matrix path described in [Theorem 2.2](#). Differentiating each chart at  $t = 0$  shows:

$$\frac{\partial}{\partial t} \Big|_{t=0} \left[ \int_{x_\alpha(U_\alpha)} f \circ F_{t\theta} \circ x_\alpha^{-1} \sqrt{|\det (\mathbf{J}^T(t)\mathbf{J}(t))|} \psi_\alpha \, dx \right] = \int_{x_\alpha(U_\alpha)} \left( \sum_{i=1}^n \frac{\partial f}{\partial x^i} \theta^i + f \sum_{i=1}^n \frac{\partial \theta^i}{\partial x^i} \right) \circ x_\alpha^{-1} \psi_\alpha \, dx,$$

where the left side of the product rule is computed by utilizing the fact that  $d_p \exp_p(0) = \text{Id}_{T_p M}$ , and the right side is an application of Jacobi's formula ([Theorem B.1](#)) coupled with [\(2.14\)](#). Finally, recall the divergence of a vector field will read in local coordinates:

$$\nabla_M \cdot \theta = \frac{1}{\sqrt{\det g_{ij}}} \sum_{i=1}^n \frac{\partial}{\partial x^i} \left( \theta^i \sqrt{\det g_{ij}} \right) \quad (2.75)$$

and note that for  $t = 0$ , we have  $F_{t\theta}^*g = g$ ,  $\sqrt{|\det g_{ij}|} = \sqrt{|\det F_{t\theta}^*g_{ij}|} = 1$ . Therefore, we obtain:

$$\frac{\partial}{\partial t} \Big|_{t=0} J(G_{t\theta}) = \int_G \nabla_M \cdot (f\theta) \, dV_g = \int_{\partial G} f\theta \cdot n_{\partial G} \, dV_{\hat{g}}, \quad (2.76)$$

where the last step is just the divergence theorem. ■

### 2.3.4 Shape functionals constrained by a partial differential equation

In this section we analyze functionals which are of the form:

$$J(G) = \int_G j(u_G) \, dV_g, \quad (2.77)$$

where  $u_G \in C^\infty(G, \mathbb{R}^n)$  is the solution to a PDE posed on a regular domain  $G$  of the manifold  $M$  and  $j : \mathbb{R}^n \rightarrow \mathbb{R}$  is a suitably chosen function. For simplicity, we analyze the case where  $u_G$  is the smooth solution to the Laplace-Beltrami equation:

$$\begin{cases} -\Delta_M u_G = f & \text{in } G \\ u_G = 0 & \text{on } \partial G, \end{cases} \quad (2.78)$$

where  $-\Delta_M$  denotes the Laplace-Beltrami operator on  $M$ . The fact that this equation be on a surface poses does not add complexity for its numerical resolution and theoretical treatment. The study [\[55\]](#)

provides an extensive survey of numerical methods and provides all the theoretical results needed for deriving the weak formulation on the surface. Alternatively, texts such as [220] provide the rigorous extension to Riemannian manifolds for results such as the divergence theorem.

In general, the form (2.77) is very common in applications and the literature of shape optimization [15]. Let us mention that even though we assume that the solution  $u_G$  be smooth, the same techniques presented in this section could be applied to functions belonging to  $C^k(G, \mathbb{R}^n)$ ,  $k \geq 0$  without too much additional work. Moreover, when working with Sobolev spaces and a weaker notion of differentiability, one can often deduce the  $C^\infty$  regularity of the solution  $u_G$  via variational estimates or, most notably, the classical theory of elliptic regularity (cf. e.g. [172, 139, 150, 165]). In order to express the shape derivative of (2.77), it proves useful to consider the definitions of transport mapping and Lagrangian derivative of  $u_G$ . Note that these concepts have been defined for the Euclidean case in Section 1.2.

**Definition 2.7 (Transport mapping).** For all  $\theta \in \mathfrak{X}(M)$ , the *transport mapping* of  $u_G$  is formally defined by:

$$\overline{u_G}(\theta) := u_{G_\theta} \circ F_\theta^M, \quad (2.79)$$

where  $F_\theta$  is the deformation of domain on  $M$ .

**Definition 2.8 (Lagrangian derivative).** Assume that  $\theta \mapsto \overline{u_G}(\theta)$  is Gateaux differentiable at  $\theta = 0$ . If this is the case, we say that  $G \mapsto u_G$  has a **Lagrangian derivative** at  $G$  in the direction  $\theta$  and we denote it by  $\dot{u}_G(\theta)$ .

The definition of Lagrangian derivative contrasts with the concept of Eulerian derivative (see Definition 1.4 for the Euclidean counterpart). In the case of the latter, we consider the Gateaux derivative of the mapping:

$$\theta \mapsto u_{G_\theta}, \quad (2.80)$$

which, for small  $\theta$ , is well defined on the deformed domain  $G_\theta$ . On the contrary, the value of  $u_G$  is not well defined on the deformed boundary  $\partial G_\theta$ . Indeed, for  $x \in \partial G$  the value  $u_G(F_\theta^M(x))$  at the transported point  $F_\theta^M(x) \in M$  will not necessarily make sense because it need not lie on  $G$  or  $\partial G$ . To this end, it proves useful to seek an alternative characterization of the Eulerian derivative in terms of the Lagrangian derivative. Formally, writing the definition of  $\dot{u}_G(\theta)$  at  $p \in M$  yields:

$$\dot{u}_G(\theta)(p) = \left. \frac{\partial}{\partial t} \right|_{t=0} \left( u_{F_{t\theta}^M(G)} \circ F_{t\theta}^M \right) (p) \quad (2.81)$$

$$= \left. \frac{\partial}{\partial t} \right|_{t=0} \left( u_{F_{t\theta}^M(G)} \right) (p) + \nabla_M u_G(p) \cdot d_p F_0^M(\theta) \quad (2.82)$$

$$= u'_G(\theta)(p) + \nabla_M u_G(p) \cdot \theta(p), \quad (2.83)$$

where we have used the chain rule and Lemma 2.2. This naturally suggests the following definition of Eulerian derivative.

**Definition 2.9 (Eulerian derivative).** If  $u_G \in C^\infty(G)$  has a Lagrangian derivative at  $G$ , then the **Eulerian derivative**  $u'_G(\theta)$  is defined by:

$$u'_G(\theta) := \dot{u}_G(\theta) - \nabla_M u_G \cdot \theta. \quad (2.84)$$

The Eulerian derivative is notably utilized in the proof of Theorem 2.4, where the expression for the Eulerian derivative of the signed distance function is employed. This connection suggests that the Eulerian derivative is closely related to the concept of the surfacic expression for the shape derivative (see Section 1.2). We shall omit the use of the Eulerian derivative since it

**Theorem 2.7.** Let  $J(G)$  be defined by (2.77) where  $u_G \in C^\infty(M)$  depending on  $G$ . Then its shape derivative in the direction  $\theta \in \mathfrak{X}(M)$  reads:

$$J'(G)(\theta) = \int_G j'(u_G) \dot{u}_G(\theta) + j(u_G) \nabla_M \cdot \theta \, dV_g. \quad (2.85)$$

*Proof.* Making a change of variables yields:

$$J(G_{t\theta}) = \int_G j(u_{G_{t\theta}} \circ F_{t\theta}^M) dV_{F_{t\theta}^*g} = \int_G j(\overline{u_G}(\theta)) dV_g. \quad (2.86)$$

After choosing smooth coordinate charts  $\{(U_\alpha, x_\alpha)\}_{\alpha \in A}$  and a subordinate partition of unity  $\{\psi_\alpha\}_{\alpha \in A}$ , we compute (similarly as in the proof of [Theorem 2.6](#)):

$$\frac{\partial}{\partial t} \Big|_{t=0} \int_{x_\alpha(U_\alpha)} j(\overline{u_G}(t\theta) \circ x_\alpha^{-1}) \sqrt{|\det(F_{t\theta}^*g)_{ij} \circ x_\alpha^{-1}|} \psi_\alpha dx \quad (2.87)$$

$$= \int_{x_\alpha(U_\alpha)} \left( j'(u_G) \dot{u}_G(\theta) + j(u_G) \sum_{i=1}^n \frac{\partial \theta^i}{\partial x^i} \right) \circ x_\alpha^{-1} \psi_\alpha dx, \quad (2.88)$$

whence the result. ■

Following very similar approaches, one can eventually obtain the exact same analogous expression for the Euclidean case.

**Proposition 2.3.** *Assume that  $u_G$  and  $p_G$  are in  $H^2(G)$ . Then,*

$$\forall \theta \in \mathcal{O}, \quad J'(G)(\theta) = \int_{\partial G} j(u_G) \theta \cdot n_{\partial G} ds - \int_{\partial G} \frac{\partial u_G}{\partial n_{\partial G}} \frac{\partial p_G}{\partial n_{\partial G}} \theta \cdot n_{\partial G} ds - \int_{\partial G} f p_G \theta \cdot n_{\partial G} ds,$$

where  $p$  is the solution to the boundary value problem:

$$\begin{cases} -\Delta_M p_G = -j'(u_G) & \text{in } G \\ p_G = 0 & \text{on } \partial G. \end{cases}$$

## 2.4 Future work

The primary objective of this chapter has been to lay the groundwork for shape derivatives within the Riemannian context, setting the stage for future research. While some significant results were not fully developed here due to time constraints, the findings presented are intended as a starting point for further exploration. For instance, various versions of Hadamard's structure theorem could be derived, each suited to different Riemannian settings. Additionally, exploring the connection between tubular neighborhoods and shape derivatives offers a promising avenue for extending the theory introduced in this chapter. Furthermore, deriving additional results for shape calculus in the Riemannian context would provide a robust theoretical foundation for advancing geometric shape optimization.

## Chapter 3

# Body-fitted tracking within a surface via a level set based mesh evolution method

---

The broad task of representing the evolution of a domain  $G(t) \subset \mathbb{R}^d$  ( $d = 2$  or  $3$  in practice) has sparked extensive mathematical and numerical investigations. This topic plays a central role in various applied disciplines, ranging from computer graphics and vision [79, 350] to the numerical simulation of physical phenomena such as fracture propagation [60] or fluid interface dynamics [90, 96], including inverse problems and shape optimization [22, 29, 69].

Multiple numerical strategies have been proposed to address this task, with competing assets and drawbacks. However, all implementations face a major challenge: in complex, realistic situations, the velocity field  $V(t, x)$  driving the motion of  $G(t)$  has a physical origin and depends on high-order geometric features of this region (notably, the curvature of  $\partial G(t)$ ), or on the solution to a boundary value problem posed on the latter. It is then notoriously difficult to find a framework reconciling an accurate description of  $G(t)$  allowing for precise calculations of these quantities at any time  $t$  – which ideally demands an exact, high-quality mesh – with the robust treatment of its evolution.

In this regard, Lagrangian strategies, tracking the evolution of an exact mesh of  $G(t)$  by displacing its vertices according to  $V(t, x)$  between successive iterations of the process, are usually undermined by a severe degradation of the quality of the mesh, which rapidly becomes invalid and incompatible with computations, see e.g. [132, 133]. Admittedly, several heuristics enhance the robustness of this practice. For instance, one may alternate deformations of the mesh with occasional remeshing steps aimed to improve its quality. Going further, one may detect and remove ill-shaped elements before they lead to complete degeneracy, or even modify the velocity of the internal vertices of the mesh to reduce the onset of overlapping patterns, see e.g. [37, 46, 75, 123, 341] about these ideas.

In this spirit, the recent Deformable Simplicial Complex technique has demonstrated the ability to cope with impressively large motions by using the formation of nearly degenerate mesh elements near the boundary of the domain as a trigger for topological changes, see [89, 88, 239]. Even more recently, the X-mesh method [240] proceeds by displacing the vertices of the mesh of  $G(t)$  according to  $V(t, x)$  up the point where the measure of some elements equals zero. The motion is then relayed between neighboring nodes while preserving the connectivity of the mesh. This strategy rests on the assumption that boundary value problems can be effectively solved on meshes with degenerate elements, under appropriate assumptions on their aspect ratios. Despite these noteworthy achievements, let us emphasize that such Lagrangian mesh deformation strategies are usually reserved for the description of “relatively small” motions of the set  $G(t)$ .

To overcome the weaknesses of Lagrangian strategies, Eulerian interface capturing techniques are based on an implicit description of the moving shape  $G(t)$ . Among these, the level set method, introduced

---

in [258], features a description of  $G(t)$  as the negative subdomain of an auxiliary “level set function”  $\phi(t, \cdot) : \mathbb{R}^d \rightarrow \mathbb{R}$  defined on the whole ambient space  $\mathbb{R}^d$ , see [257, 293]. The domain  $G(t)$  is never meshed explicitly, being rediscovered at each iteration of the process from the values of  $\phi(t, \cdot)$ . Although it allows to describe arbitrarily large motions, such an implicit representation is unfortunately less amenable to the accurate solution of partial differential equations defined on  $G(t)$ .

These central and popular questions in the numerical analysis of the motion of a domain  $G(t)$  have received surprisingly little attention when the ambient medium is a manifold – notably a surface  $S$  in  $\mathbb{R}^3$ ; yet, this alternative context embraces multiple applications of interest:

- Geometric flows, such as the mean curvature flow or the Willmore flow, where the velocity field  $V(t, x)$  depends on high-order geometric features of  $G(t)$ , can be adapted to the case of a region within a surface [316].
- Diverse operations of interest in the field of computer graphics are conveniently formulated in terms of the evolution of a region within a surface. For instance, one popular model for the generation of textures on a surface relies on the resolution of a reaction-diffusion equation, see e.g. [319, 328]; also, image segmentation on surfaces can be addressed thanks to a suitable adaptation of the Chan-Vese algorithm [316].
- Various physical evolution problems occur within a surface, see for instance [248] about the solidification of a thin fluid film front on a surface substrate, and [315] about the dynamics of phase changes on surfaces in material science.
- The wish to optimize the shape of regions within a given ambient surface arises for instance in the optimal design of shells [318], of curvilinear electronic devices [208], in the context of surface flows [117], or in the identification of optimal fixation systems for mechanical structures, see e.g. [108, 333].

To the best of our knowledge, the first numerical simulations of the evolution of a region  $G(t)$  within a surface  $S$ , proposed in [87] and [204], were concerned with the geodesic curvature flow. These leverage a version of the level set method tailored to the datum of  $S$  as a parametrized patch, or as the graph of a function defined on the 2d space, respectively. A more general setting is proposed in [54] and [84], where the closed surface  $S := \{x \in \mathbb{R}^d, \psi(x) = 0\}$  is represented as the 0 level set of a fixed function  $\psi : \mathbb{R}^d \rightarrow \mathbb{R}$ , and  $G(t) = \{x \in S, \phi(t, x) < 0\}$  is the negative subregion of  $S$  induced by another (time-dependent) level set function  $\phi(t, \cdot) : \mathbb{R}^d \rightarrow \mathbb{R}$ . Here and in the subsequent investigations in this framework [53, 169, 281], the equation governing the evolution of the level set function  $\phi(t, \cdot)$  is formulated in the whole space  $\mathbb{R}^d$  with the help of projection operators. Another series of contributions [231, 278] leverages the so-called closest point method of [230], devoted to the solution of partial differential equations on surfaces. As the latter solely requires the datum of a mapping associating to any point  $x \in \mathbb{R}^d$  one closest point (in terms of Euclidean distance) to the surface  $S$ , this latter framework leaves the room for  $S$  to be open.

The present chapter aims to introduce a robust numerical methodology for tracking arbitrarily large motions of a region  $G(t)$  within an ambient surface  $S \subset \mathbb{R}^3$  – including changes of its topology – while maintaining an exact meshed representation of the latter throughout the process. The proposed strategy is a natural extension of the earlier works [12, 13, 14] – which were devoted to evolving domains of the Euclidean space  $\mathbb{R}^d$  – to the present context where the ambient medium is a surface  $S$  in  $\mathbb{R}^d$ . It combines two complementary representations of  $G(t)$  at each stage of the evolution: on the one hand,  $G(t)$  is explicitly discretized, as a submesh of a high-quality surface triangulation  $\mathcal{T}$  of the ambient surface  $S$ , which allows to accurately calculate its geometric features or to solve related boundary value problems via the finite element method – and thereby to evaluate the velocity field  $V(t, x)$  precisely. On the other hand,  $G(t)$  is described implicitly via the level set method, as the negative subdomain of a scalar function  $\phi(t, \cdot) : S \rightarrow \mathbb{R}$ , so that arbitrarily large motions of  $G(t)$  can be realized. The cornerstone of this strategy is a set of efficient meshing algorithms and numerical schemes for passing from one representation to the other.

This chapter is organized as follows. In the next Section 3.1, we present in more details the issue of tracking the evolution of a region within a surface. Then, in Section 3.2, we describe the proposed numerical strategy to realize this task and we detail its main ingredients; we notably discuss the calculation of the signed distance function to a region on a surface, the resolution of the level set evolution equation in this context, and our main remeshing operations dedicated to surface triangulations. Interestingly, these numerical methods are implemented in open-source codes which can easily be used in a black-box fashion. A few numerical applications of our framework are presented in Sections 3.3 to 3.5: after appraising its

efficiency on an analytical test-case, we consider the motion of a complex interface accounting for a fire front, whose expansion within a fixed landscape is driven by geometric quantities attached to the fire and those of the landscape.

### 3.1 Evolution of a region on a surface in the level set framework

This section introduces the issue of evolving regions within a surface and sets the main notations used throughout.

Let  $S$  be a smooth hypersurface, with or without boundary, in the  $d$ -dimensional space  $\mathbb{R}^d$ . Here,  $d$  equals 2 or 3, and although we shall focus on the three-dimensional case  $d = 3$ , which is on any point more challenging than its 2d counterpart, we retain the generic notation  $d$  whenever possible. For simplicity of the discussion, the hypersurface  $S$  is assumed to be oriented, but our developments and numerical methods extend readily to the case of a non-orientable surface. For any point  $x \in S$ ,  $n_S(x)$  is the unit normal vector to  $S$  at  $x$  and the tangent plane to  $S$  at  $x$  – that is, the vector plane orthogonal to  $n_S(x)$  – is denoted by  $T_x S := \{v \in \mathbb{R}^d, v \cdot n_{\partial\Omega}(x) = 0\}$ .

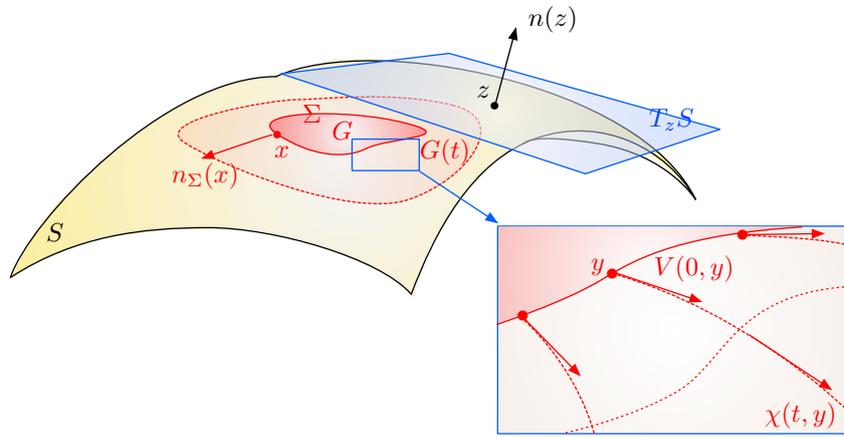


Figure 3.1: Example of a region  $G(t)$  of a three-dimensional surface  $S$  evolving according to a tangential velocity field  $V(t, x)$ .

Let  $G \subset S$  be a smooth open subdomain of  $S$  with boundary  $\Sigma := \partial G$ . For  $x \in \Sigma$ , we denote by  $n_\Sigma(x) \in T_x S$  the conormal vector to  $\Sigma$  at  $x$ , pointing outward  $G$ , see Fig. 3.1. Let  $V : (0, T) \times S \rightarrow \mathbb{R}^3$  be a smooth vector field defined over the considered time period  $(0, T)$ , which is tangential to  $S$ , that is:

$$\forall t > 0 \text{ and } x \in S, \quad V(t, x) \in T_x S. \quad (3.1)$$

We wish to track the region  $G(t)$  evolving from  $G(0) \equiv G$  under the effect of  $V(t, x)$  over  $[0, T]$ . This notion of evolution is intuitively defined as follows: for any  $x \in S$ , let  $t \mapsto \chi(t, 0, x)$  be the characteristic curve of  $V$  emerging from  $x$  at time 0, i.e. the solution to the ordinary differential equation

$$\begin{cases} \frac{d\chi}{dt}(t, 0, x) = V(t, \chi(t, 0, x)), & \text{for } t \in (0, T), \\ \chi(0, 0, x) = x. \end{cases} \quad (3.2)$$

The region  $G(t)$  is then defined as the set of the positions occupied at time  $t$  by the points lying in  $G$  at time 0:

$$G(t) = \left\{ \chi(t, 0, x), \quad x \in G \right\}. \quad (3.3)$$

The numerical simulation of domain evolution has long been a challenging issue. Among the numerous frameworks implemented to achieve this goal, the level set method has proved to be particularly convenient since its introduction in [258]; we refer to [257, 293] for a presentation of various aspects of the level set method and its countless applications in scientific computing, see also [163] about its mathematical aspects and its impact on the theory of moving domains.

In the context of the present chapter, where the ambient medium is a surface  $S \subset \mathbb{R}^d$  and  $G \subset S$  is an open region with boundary  $\Sigma := \partial G$ , an auxiliary “level set” function  $\phi : S \rightarrow \mathbb{R}$  is introduced, whose

negative subset coincides with  $G$ , that is:

$$\forall x \in S, \quad \begin{cases} \phi(x) < 0 & \text{if } x \in G, \\ \phi(x) = 0 & \text{if } x \in \Sigma, \\ \phi(x) > 0 & \text{otherwise.} \end{cases} \quad (3.4)$$

A formal use of the chain rule reveals that if  $G(t)$  is a smooth region of  $S$  evolving according to a smooth tangential velocity field  $V(t, x)$  over a time period  $[0, T]$ , any associated level set function  $\phi(t, x)$  satisfies the so-called level set advection equation:

$$\frac{\partial \phi}{\partial t}(t, x) + V(t, x) \cdot \nabla_S \phi(t, x) = 0, \quad t \in (0, T), \quad x \in S, \quad (3.5)$$

where  $\nabla_S \phi := \nabla \phi - (\nabla \phi \cdot n)n$  denotes the tangential gradient of the function  $\phi$  (with respect to the spatial variable). Alternatively, introducing the component  $v(t, x)$  of  $V(t, x)$  in the direction of the conormal vector  $n_{\Sigma(t)}(x) = \frac{\nabla_S \phi(t, x)}{|\nabla_S \phi(t, x)|}$  to  $G(t)$ , that is

$$v(t, x) := V(t, x) \cdot n_{\Sigma(t)}(x), \quad (3.6)$$

the equation (3.5) rewrites as a Hamilton-Jacobi equation:

$$\frac{\partial \phi}{\partial t}(t, x) + v(t, x)|\nabla_S \phi(t, x)| = 0, \quad t \in (0, T), \quad x \in S. \quad (3.7)$$

In our applications, notably those targeting the description of the evolution of physical interfaces, the velocity field  $V(t, x)$  may depend on the moving region  $G(t)$  in a very complicated way, often not only through geometric quantities but also via the solution to partial differential equations involving  $G(t)$ . Hence, the velocity field  $V(t, x)$  and its normal component  $v(t, x)$  in (3.6) depend in an implicit way on  $G(t)$ , and thus on  $\phi$  itself. The only realistic means to address the numerical resolution of (3.5) is a fully explicit procedure: the total time interval  $(0, T)$  is decomposed into a series of subintervals of the form  $(t^n, t^{n+1})$ , which are “small enough” so that  $V(t, x)$  can be frozen in time, that is

$$\forall t \in (t^n, t^{n+1}), \quad V(t, x) \approx V(t^n, x);$$

this practice leads to the solution of a series of “true” advection equations of the form (3.5) with time-independent fields  $V(t^n, \cdot)$  computed from the region  $G(t^n)$  (or the level set function  $\phi(t^n, \cdot)$ ). An alternative approach consists in freezing only the normal component  $v(t, x)$  over each subinterval  $(t^n, t^{n+1})$ , i.e.

$$\forall t \in (t^n, t^{n+1}), \quad v(t, x) \approx v(t^n, x),$$

which leads to the resolution of a series of “true” non linear Hamilton-Jacobi equations of the form (3.7), with time-independent normal velocities  $v(t^n, \cdot)$ . This second possibility retains more information from the original evolution equation (3.5) (namely, the fact that the motion is consistently oriented in the direction of the normal vector  $n_{\Sigma(t)}$ ), but it requires the solution of more complex, non linear evolution equations.

**Remark 3.1.** *In physical applications, the velocity  $V(t, x)$  of the region  $G(t)$  often makes sense only on the boundary  $\Sigma(t)$ , while the formalism of the level set method requires that it should be extended to the whole surface  $S$ , see (3.5) and (3.7). Actually, it is a classical feature of the level set method that under “mild assumptions”, the 0 level set of the solution to (3.5) or (3.7) does not depend on the choice of such an extension for  $V(t, \cdot)$  outside  $\Sigma(t)$ , see [163]. In practice, however, the choice of a particular extension may have a great impact on the numerical realization of the motion of  $G(t)$ , and it may be conducted differently depending on the application, see Section 3.3 for several examples.*

**Remark 3.2.** *From the mathematical viewpoint, the intuitive definition (3.3) of the evolving region  $G(t)$  makes sense as long as  $G(t)$  and  $V(t, x)$  are “smooth enough”, which is usually the case when the time  $t$  is “small enough”. Unfortunately, even in the case of a “simple” flow, featuring a “smooth” initial state  $G(0)$ , the region  $G(t)$  or the velocity  $V(t, x)$  will inevitably become singular in finite time [27, 163]. One possibility to define a generalized motion (3.3) beyond this point rests on the level set method: considering one level set function  $\phi(0, \cdot)$  for the initial region  $G(0)$ , the evolution equation (3.5) is solved in the generalized sense of viscosity, see [97], or again [27, 163] about this notion. It turns out that, under reasonable assumptions, (3.5) has a unique viscosity solution  $\phi(t, \cdot)$ , from which  $G(t)$  is then defined by*

$$G(t) := \left\{ x \in S, \phi(t, x) < 0 \right\}.$$

*Such theoretical questions have been extensively studied, particularly in situations where  $V(t, x)$  is composed of geometric quantities attached to  $G(t)$ , see for instance [192] about the study of the mean curvature flow within a surface. Without entering into details, let us mention a few concurrent attempts to the level set method aimed at generalizing this motion of  $G(t)$  past the onset of singularities.*

- *Parametric methods insist on the description (3.3) of the evolving region  $G(t)$ , relying on a (tedious) classification of the various types of possible singularities and on an appropriate selection of what is the “correct” evolution of  $G(t)$  in each case, see for instance [135].*
- *Varifold solutions were initially proposed in [61] to deal with the problem of domain evolution. These are measure-theoretic solutions which unfortunately lack uniqueness in their characterization of the evolving set  $G(t)$ .*
- *Phase field methods were introduced in this mathematical context in [64, 82, 115]. They encode the evolution problem of  $G(t)$  into a scalar “phase field” function, taking values  $-1$  “well inside”  $G(t)$  and  $1$  “well outside”  $G(t)$ ; the thin transition region between both zones is sought as the solution to an energy minimization problem, see [128] for a recent overview.*

## 3.2 Presentation of the numerical strategy

In this section again,  $G(t)$  denotes a region of a fixed surface  $S \subset \mathbb{R}^d$ , evolving over a time period  $(0, T)$  according to a tangential velocity field  $V(t, x)$ , see (3.1). We do not specify the nature of  $V$  for the moment, but we assume that the calculation of  $V(t, \cdot) : S \rightarrow \mathbb{R}^d$  at one particular time is difficult and costly, as it involves either geometric quantities, or the solution to a boundary value problem attached to  $G(t)$ .

The proposed numerical strategy for tracking the evolution of  $G(t)$  is summarized in Section 3.2.1. The pivotal ingredients involved in its implementation are fairly classical when the ambient medium is the 2d or 3d space, but much less so when it is a surface  $S$  in  $\mathbb{R}^3$ . Hence, we present them in this context with some details in the next Sections 3.2.2 to 3.2.4.

### 3.2.1 Outline of the numerical algorithm

The time interval  $(0, T)$  is split into a series of subintervals of the form  $(t^n, t^{n+1})$ ,  $n = 0, \dots, N - 1$ , where  $t^n = n\Delta t$  and  $\Delta t$  is a “small” time step; we indicate with a superscript  $n$  the value of a time-dependent object at  $t^n$ : for instance,  $G^n$  stands for  $G(t^n)$ ,  $V^n(\cdot)$  denotes the velocity field  $V(t^n, \cdot)$ , etc.

The proposed algorithm for tracking the motion of  $G(t)$  is based on two complementary representations of each intermediate configuration  $G^n$ :

- *A level set representation.* On the one hand,  $G^n$  is known as the negative subset of a level set function  $\phi^n : S \rightarrow \mathbb{R}$ , i.e. (3.4) holds. In practice,  $\phi^n$  is supplied at the vertices of a triangular mesh of  $S$ , see Fig. 3.2 (a).
- *A meshed representation.* On the other hand,  $G^n$  is meshed exactly. More precisely, a triangular mesh  $\mathcal{T}^n$  of  $S$  is available, with the following properties:
  - (i)  $\mathcal{T}^n$  is valid: the intersection between any two different open triangles  $T_i, T_j \in \mathcal{T}^n$  is empty;

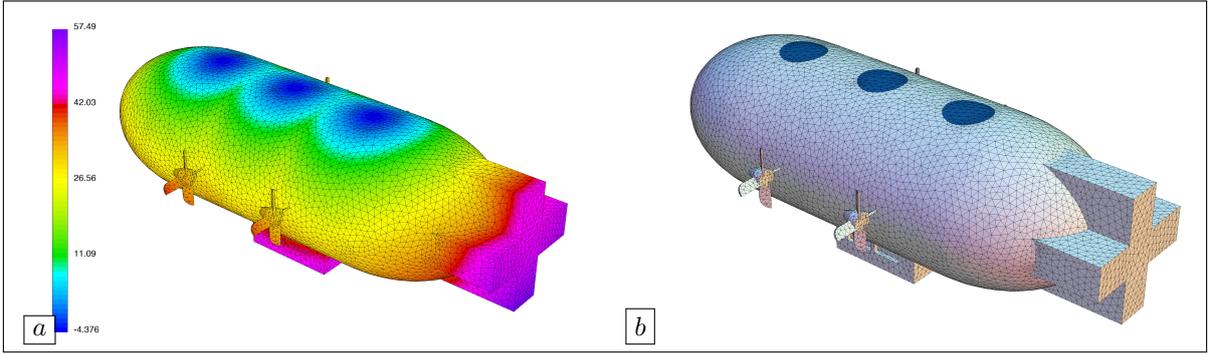


Figure 3.2: (a) Level set function  $\phi : S \rightarrow \mathbb{R}$  for a region  $G \subset S$  defined at the vertices of the triangulation of a surface  $S \subset \mathbb{R}^3$ ; (b) High-quality mesh  $\mathcal{T}$  of  $S$ , enclosing a submesh  $\mathcal{T}_{\text{int}}$  of  $G$  (in dark blue).

- (ii)  $\mathcal{T}^n$  is conforming: the intersection  $\overline{T_i} \cap \overline{T_j}$  between the closures of any two different triangles  $T_i, T_j \in \mathcal{T}^n$  is either a vertex or an edge of  $\mathcal{T}^n$ ;
- (iii)  $\mathcal{T}^n$  has high quality: the triangles  $T \in \mathcal{T}^n$  are close to being equilateral;
- (iv)  $\mathcal{T}^n$  is made of two submeshes  $\mathcal{T}_{\text{int}}^n, \mathcal{T}_{\text{ext}}^n$  – that is, collections of subsets of its triangles – associated to the respective regions  $G^n$  and  $S \setminus \overline{G^n}$ .

The requirements (i-iii) are ubiquitous in the scientific computing literature, notably when it comes to guaranteeing the accuracy of finite element computations on  $S$  [92]; they are illustrated on Fig. 3.3. The property (iv) is more specific to our meshed representation, see Fig. 3.2 (b).

The meshed representation of  $G^n$  is particularly useful when it comes to calculating some of its geometric features (e.g. the normal vector field  $n_{\Sigma^n}$  to  $\Sigma^n$ , its curvature, etc.), or to solve related “physical” boundary value problems – operations which are involved in the definition of the velocity field  $V_{\partial\Omega}^n(x)$  or its normal component  $v_{\partial\Omega}^n(x)$ . In turn, the level set representation  $\phi^n$  allows for a robust description of the motion of  $G(t)$  between the times  $t^n$  and  $t^{n+1}$  via the solution of the advection equation (3.5) with velocity field  $V^n$ , or that of the Hamilton-Jacobi equation (3.7) with normal velocity  $v^n$ . Efficient numerical algorithms make it possible to switch between these representations so that every operation involving  $G^n$  can be carried out within the most appropriate framework.

Our numerical algorithm for the simulation of the motion of  $G(t)$  proceeds as follows, see Algorithm 2 for a summarizing sketch. Every iteration  $n = 0, \dots$  starts with the datum of the region  $G^n$  under meshed representation: a valid, conforming and high-quality mesh  $\mathcal{T}^n$  of  $S$  is available, a submesh  $\mathcal{T}_{\text{int}}^n$  of which is an explicit mesh for the region  $G^n$ . By performing geometric or mechanical calculations related to  $G^n$  on this mesh, the velocity field  $V^n : S \rightarrow \mathbb{R}^d$  is calculated at the vertices of  $\mathcal{T}^n$ . A particular level set representation  $\phi^n : S \rightarrow \mathbb{R}$  for  $G^n$  is then calculated at the vertices of  $\mathcal{T}^n$ , as the signed distance function  $d_{G^n}$  to  $G^n$ . The evolution of  $G(t)$  between times  $t^n$  and  $t^{n+1}$  is carried out by solving the evolution equation (3.5) over the time period  $(t^n, t^{n+1})$  with the velocity field  $V^n$  and the initial datum  $\phi^n$ . This yields a level set representation  $\phi^{n+1} : S \rightarrow \mathbb{R}$  for  $G^{n+1}$ , at the vertices of the mesh  $\mathcal{T}^n$ . Finally, a meshed representation for  $G^{n+1}$  is obtained from these data, with the help of suitable remeshing algorithms; this produces a new high-quality mesh  $\mathcal{T}^{n+1}$  of  $S$  is produced, a submesh  $\mathcal{T}_{\text{int}}^{n+1}$  that explicitly discretizes  $G^{n+1}$ .

The main stages of this method are described in more detail in the next sections: in Section 3.2.2, we discuss the numerical computation of the signed distance function to a subregion  $G$  of a surface  $S$ , which allows to pass from a meshed description of  $G$  to a level set description. In Section 3.2.3, we describe the numerical solution of the level set advection equation (3.5) on the surface  $S$ , accounting for the update of  $G(t)$  between successive iterations. Finally, in Section 3.2.4, we outline the remeshing operations involved in the construction of a meshed representation of a region  $G \subset S$  from a level set representation  $\phi : S \rightarrow \mathbb{R}$ .

### 3.2.2 Computation of the signed distance function on a surface

Let  $\mathcal{T}$  be a triangulation of a surface  $S \subset \mathbb{R}^d$ , and let  $\mathcal{T}_{\text{int}}$  denote a submesh for a region  $G \subset S$ ; we wish to generate a level set function  $\phi : S \rightarrow \mathbb{R}$  for  $G$  at the vertices of  $\mathcal{T}$ , i.e. a function  $\phi$  satisfying (3.4).

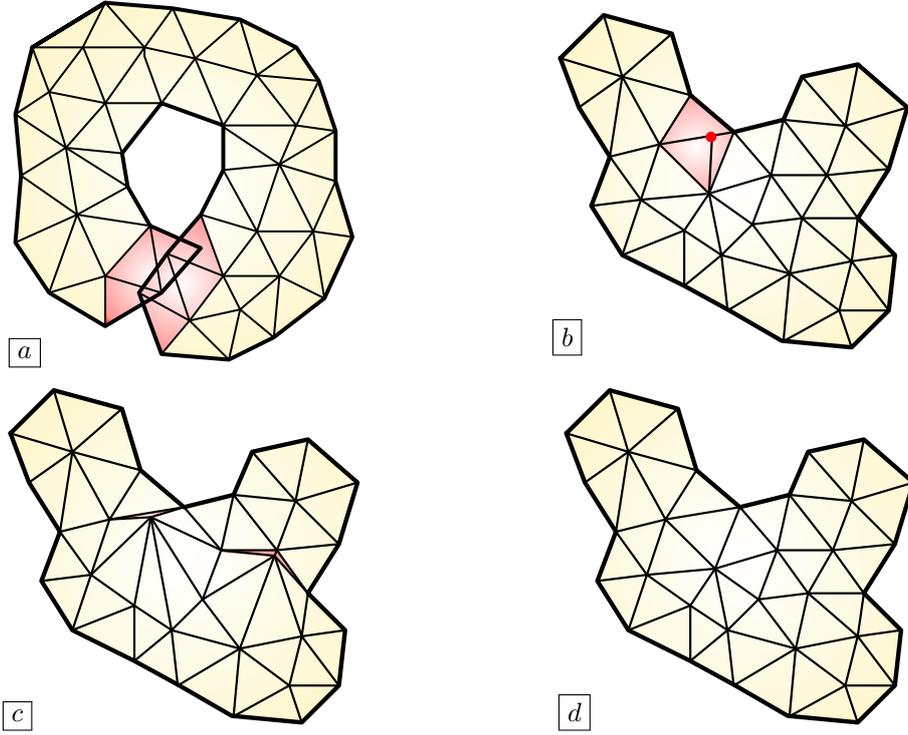


Figure 3.3: (a) Invalid two-dimensional triangulation, presenting overlapping triangles (in red); (b) Valid, yet non conforming mesh (the red node lies strictly inside an edge); (c) Valid, conforming, but ill-shaped mesh (the red triangles are nearly degenerate); (d) High-quality computational mesh.

---

**Algorithm 2:** Body-fitted tracking of the evolution of a region  $G(t) \subset S$ .

---

**Input:** Mesh  $\mathcal{T}^0$  of  $S$  featuring an explicit discretization of the initial region  $G^0$ .

**for**  $n = 0, \dots, N - 1$  **do**

1. Compute the velocity field  $V_{\partial\Omega}^n(x)$  at the vertices  $x$  of  $\mathcal{T}^n$  with the help of the meshes  $\mathcal{T}_{\text{int}}^n, \mathcal{T}_{\text{ext}}^n$  of  $G^n$  and  $S \setminus \overline{G^n}$ , respectively.
2. Compute the signed distance function  $d_{G^n}$  to  $G^n$  at the vertices of the mesh  $\mathcal{T}^n$  of  $S$ .
3. Solve the advection equation

$$\begin{cases} \frac{\partial\phi}{\partial t}(t, x) + V_{\partial\Omega}^n(x) \cdot \nabla_S \phi(t, x) = 0 & \text{for } (t, x) \in (0, \Delta t) \times S, \\ \phi(0, x) = d_{G^n}(x) & \text{for } x \in S, \end{cases}$$

on the total mesh  $\mathcal{T}^n$  of  $S$ . A new level set function  $\phi^{n+1} = \phi(\Delta t, \cdot)$  is obtained for

$$G^{n+1} = \{x \in S, \phi^{n+1}(x) < 0\}.$$

4. From the datum of  $\phi^{n+1}$  at the vertices of  $\mathcal{T}^n$ , create a new, high-quality mesh  $\mathcal{T}^{n+1}$  of  $S$  made of two submeshes  $\mathcal{T}_{\text{int}}^{n+1}$  and  $\mathcal{T}_{\text{ext}}^{n+1}$  for  $G^{n+1}$  and  $S \setminus \overline{G^{n+1}}$ , respectively.

**end**

**Output:** Mesh  $\mathcal{T}^N$  of  $S$  featuring an explicit discretization  $\mathcal{T}_{\text{int}}^N$  of  $G^N$ .

---

Although many choices are possible, stability issues in the numerical practice of the level set method raise the need to select one which presents “neither too steep, nor too flat” variations, see e.g. [87]. To comply with this requirement, and due to its desirable properties in connection with the geometry of  $G$  (see e.g. [116]), we calculate the signed distance function  $d_G$  to  $G$ , defined by

$$\forall x \in S, \quad d_G(x) = \begin{cases} -d^S(x, \Sigma) & \text{if } x \in G, \\ 0 & \text{if } x \in \Sigma, \\ d^S(x, \Sigma) & \text{if } x \in S \setminus \overline{G}, \end{cases} \quad (3.8)$$

where the distance  $d^S(x, \Sigma) = \inf_{p \in \Sigma} d^S(x, p)$  from a point  $x \in S$  to  $\Sigma$  is defined from the geodesic distance

$$\forall x, y \in S, \quad d^S(x, y) = \inf \left\{ \int_0^1 |\gamma'(u)| \, du, \quad \gamma \in \mathcal{C}^1([0, 1], S), \quad \gamma(0) = x, \quad \gamma(1) = y \right\}.$$

Multiple numerical algorithms allow to calculate the signed distance function to a subdomain of  $\mathbb{R}^2$  or  $\mathbb{R}^3$ : the fast marching method [291], the fast sweeping method [349], among others. Much fewer allow to deal with the case where the ambient medium is a surface of  $\mathbb{R}^d$  equipped with a surface triangulation, as is our concern in the present work; in our implementation, we rely on the non trivial extension of the fast marching algorithm proposed in [205].

### 3.2.3 Resolution of the level set advection equation on the surface $S$

In this section, we discuss the numerical realization of the evolution of  $G(t)$  over a generic time period  $(0, T_g)$  (which stands for any of the intervals  $(t^n, t^{n+1})$  featured in [Algorithm 2](#)) according to a stationary tangential vector field  $V(x)$  (accounting for  $V_{\partial\Omega}^n(x)$ ), or normal velocity  $v(x)$ , starting from an initial datum  $\phi_0$  (representing  $\phi^n$ ). We consider the solution of the level set evolution equation under advection form:

$$\begin{cases} \frac{\partial \phi}{\partial t}(t, x) + V(x) \cdot \nabla_S \phi(t, x) = 0, & \text{for } t \in (0, T_g), \quad x \in S, \\ \phi(0, x) = \phi_0(x), & \text{for } x \in S, \end{cases} \quad (3.9)$$

or that of its Hamilton-Jacobi counterpart:

$$\begin{cases} \frac{\partial \phi}{\partial t}(t, x) + v(x) |\nabla_S \phi(t, x)| = 0, & \text{for } t \in (0, T_g), \quad x \in S, \\ \phi(0, x) = \phi_0(x), & \text{for } x \in S. \end{cases} \quad (3.10)$$

This topic is quite classical in the literature when the ambient medium is the Euclidean space  $\mathbb{R}^2$  or  $\mathbb{R}^3$ . Efficient numerical schemes are available if the latter is discretized with a Cartesian grid, see notably [296, 302, 198] about (weighted) Essentially Non Oscillatory finite difference methods, and [348] for a survey. In the case when the computational support is a simplicial mesh, we refer to [1, 47, 131, 255] for adapted numerical schemes for the Hamilton-Jacobi equation (3.10), and more recently to [36, 120] about discontinuous Galerkin methods for the advection equation (3.9).

By contrast, the resolution of (3.9) or (3.10) has been seldom considered in the present context where the ambient medium is a surface  $S$  in  $\mathbb{R}^3$ . The aforementioned articles [53, 169, 230, 278, 281], which are based on a level set or closest point description of  $S$ , rely on the construction of reformulations of the equations (3.9) or (3.10) on the whole ambient space  $\mathbb{R}^3$ . In [132], a finite element method on a triangulation of  $S$  is proposed for the conservative counterpart of the advection equation (3.9).

In our implementation, following [66], we solve the advection equation (3.9) thanks to the *method of characteristics* [264], a procedure which can be understood as a semi-Lagrangian scheme for the original evolution equation (3.5), see [309]. This method relies on the explicit expression of the solution to (3.9) in terms of the characteristic curves  $t \mapsto \chi(t, t_0, x)$  of the velocity field  $V$ . Like in [Section 3.1](#), for  $t_0 \in (0, T_g)$  and  $x \in S$ ,  $\chi(\cdot, t_0, x)$  is characterized by the following ordinary differential equation:

$$\begin{cases} \frac{d\chi}{dt}(t, t_0, x) = V(t, \chi(t, t_0, x)), & \text{for } t \in (0, T_g), \\ \chi(t_0, t_0, x) = x. \end{cases} \quad (3.11)$$

The solution to (3.9) then reads:

$$\phi(t, x) = \phi_0(\chi(0, t, x)), \quad t \in [0, T_g], \quad x \in S, \quad (3.12)$$

that is, the value of  $\phi$  at time  $t$  and point  $x$  is the value taken by the initial function  $\phi_0$  at the position initially occupied by the particle lying in  $x$  at time  $t$ .

We leverage this property by discretizing explicitly the formula (3.12). In our implementation, the surface  $S$  is equipped with a triangulation  $\mathcal{T}$ ; the velocity field  $V(x)$  and the initial datum  $\phi_0$  are Lagrange  $\mathbb{P}_1$  finite element functions on  $\mathcal{T}$ : they are defined by their values at the vertices of  $\mathcal{T}$ , and their evaluation at other points on  $S$  is achieved by piecewise linear interpolation. For every vertex  $x$  of  $\mathcal{T}$ , we solve the ordinary differential equation (3.11) for the position  $\chi(0, T_g, x)$  thanks to a classical Euler scheme, or by a more involved Runge-Kutta strategy. This task brings into play (yet another) subdivision of the time interval  $(0, T_g)$ ; it relies on efficient data structures for locating the neighbors of the triangles in the mesh  $\mathcal{T}$ , and some care is needed when realizing linear combinations of the various velocity vectors attached to different points  $x \in S$ , which belong to different tangent planes. We refer to [245] for the implementation of Runge-Kutta methods for the solution of ordinary differential equations on surfaces.

**Remark 3.3.** *In practical situations, some of the characteristic lines  $u \mapsto \chi(u, T_g, x)$  may not be defined over the whole interval  $[0, T_g]$ . This happens when  $S$  is open and  $V(x) \cdot n_S(x) < 0$  at some points  $x \in \partial S$ , where  $n_S(x) \in T_x S$  is the conormal vector to the surface  $S$ . Physically, the velocity field enters the surface at such points, and the equation (3.9) has to be complemented with adequate boundary conditions at such “entrant” regions of the boundary  $\partial S$ . When this situation occurs, we simply linearly interpolate the values of  $V$  and  $\phi$  outside the surface  $\partial S$  from their values on  $\partial S$  to complete the integration of (3.11) with consistent values.*

### 3.2.4 Meshing of the negative subdomain of a level set function

Let  $\mathcal{T}$  be a triangulation of a surface  $S$  in  $\mathbb{R}^3$ , and let  $\phi : S \rightarrow \mathbb{R}$  be a level set function for a region  $G \subset S$ , which is supplied by its values at the vertices of  $\mathcal{T}$ . We aim to construct a new, high-quality computational mesh  $\tilde{\mathcal{T}}$  of  $S$  which comprises two submesh  $\tilde{\mathcal{T}}_{\text{int}}$  and  $\tilde{\mathcal{T}}_{\text{ext}}$  for the regions  $G$  and  $S \setminus \bar{G}$ . This operation can be achieved by various strategies, see e.g. [153, 154], and we adopt here that of our previous work [102].

The latter proceeds in two steps:

1. The triangles  $T \in \mathcal{T}$  crossing the 0 level set  $\Sigma = \partial G$  of  $\phi$  are identified from the values of this function at the vertices of  $\mathcal{T}$ , and  $\Sigma$  is discretized explicitly into  $\mathcal{T}$ . This pretty simple operation is based on the so-called marching tetrahedra algorithm [127] – a variant of the famous marching cubes method [226]: pre-defined patterns are used to split each triangle  $T \in \mathcal{T}$  into a valid, conforming configuration where the line segments  $\Sigma \cap T$  appears explicitly. This step results in a surface triangulation  $\mathcal{T}_{\text{temp}}$  of  $S$  featuring explicit submeshes  $\mathcal{T}_{\text{temp, int}}$  and  $\mathcal{T}_{\text{temp, ext}}$  of  $G$  and  $S \setminus \bar{G}$ . Unfortunately,  $\mathcal{T}_{\text{temp}}$  is ill-shaped: it inevitably features thin, nearly flat elements, which makes it unsuitable for the practice of accurate geometric and finite element calculations, see again [92] about this classical issue.
2. The intermediate mesh  $\mathcal{T}_{\text{temp}}$  is iteratively modified to improve the quality of its elements, i.e. to make them close to equilateral, insofar as possible. A new, high-quality mesh  $\tilde{\mathcal{T}}$  of  $S$  is obtained, which provides explicit discretizations of  $G$  and  $S \setminus \bar{G}$ .

The latter step is by far the most complicated of the process, and it deserves a few comments. It starts with a series of geometric computations, such as the normal vector  $n$  to  $S$ , the conormal vector  $n_\Sigma$  to  $\Sigma$  and the deviation of their values at neighboring vertices of  $\mathcal{T}$ . This allows to identify the suitable local size of the elements of the mesh ensuring an accurate approximation of  $S$  and  $G$ .

Then, guided by this information, four local remeshing operations are intertwined, provided they improve the global quality of the mesh, see e.g. [152].

- *Edge split.* An edge  $pq$  in  $\mathcal{T}$  which is “too long” is split by introducing a new point  $m$  and reconnecting the triangles sharing  $pq$  as an edge accordingly, see Fig. 3.5 (a).
- *Edge collapse.* The endpoints of an edge  $pq$  which is “too short” are merged, see Fig. 3.5 (b).
- *Edge swap.* The edge  $pq$  between two adjacent triangle  $pqr$  and  $pqs$  is suppressed and the alternate configuration, featuring the edge  $rs$  and the triangles  $rsp$  and  $rsq$ , is retained, see Fig. 3.5 (c).
- *Vertex relocation.* A vertex  $p$  of  $\mathcal{T}$  is slightly moved on the continuous surface, see Fig. 3.5 (d).

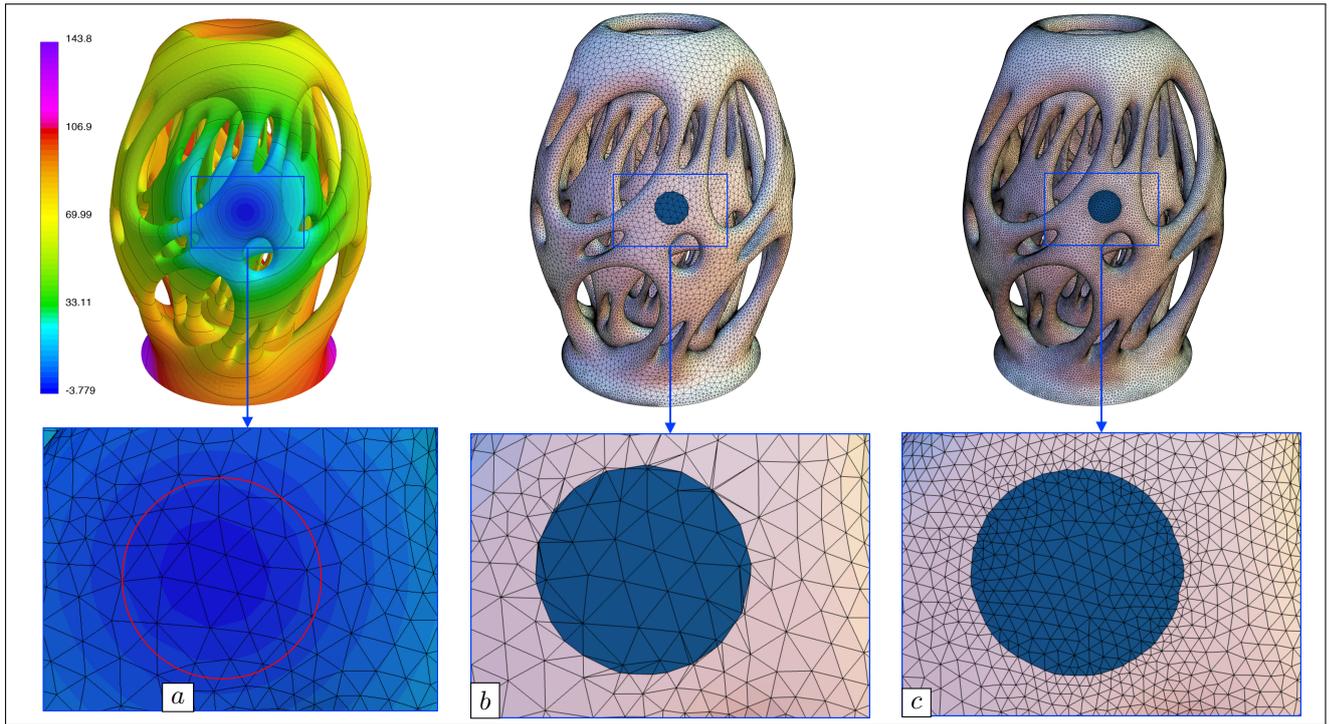


Figure 3.4: (a) Values of a level set function  $\phi : S \rightarrow \mathbb{R}$  for a region  $G \subset S$ ; (b) Ill-shaped mesh  $\mathcal{T}_{\text{temp}}$  obtained after explicit discretization of  $G$  into  $\mathcal{T}$ ; (c) High-quality mesh  $\tilde{\mathcal{T}}$  obtained after remeshing  $\mathcal{T}_{\text{temp}}$ .

Importantly, this remeshing stage leaves room for an adaptation of the computational mesh  $\mathcal{T}$  with respect to geometric quantities of  $S$  or  $G$ , or to a priori or a posteriori error estimates attached to the resolution of partial differential equations on surfaces.

**Remark 3.4.** *In some applications, such as those of [Chapter 4](#), it happens that the considered surface  $S$  is the boundary  $\partial\Omega$  of a three-dimensional domain  $\Omega$  equipped with a tetrahedral mesh  $\mathcal{K}$ , and that the considered surface triangulation  $\mathcal{T}$  is the boundary part of  $\mathcal{K}$ . In such a situation, the above operations can be applied to the whole tetrahedral mesh  $\mathcal{K}$ , producing a new mesh  $\tilde{\mathcal{K}}$  of  $\Omega$ , whose surface part  $\tilde{\mathcal{T}}$  is a triangulation of  $\partial\Omega$  enclosing submeshes  $\tilde{\mathcal{T}}_{\text{int}}$  and  $\tilde{\mathcal{T}}_{\text{ext}}$  of  $G$  and  $S \setminus \bar{G}$ , respectively.*

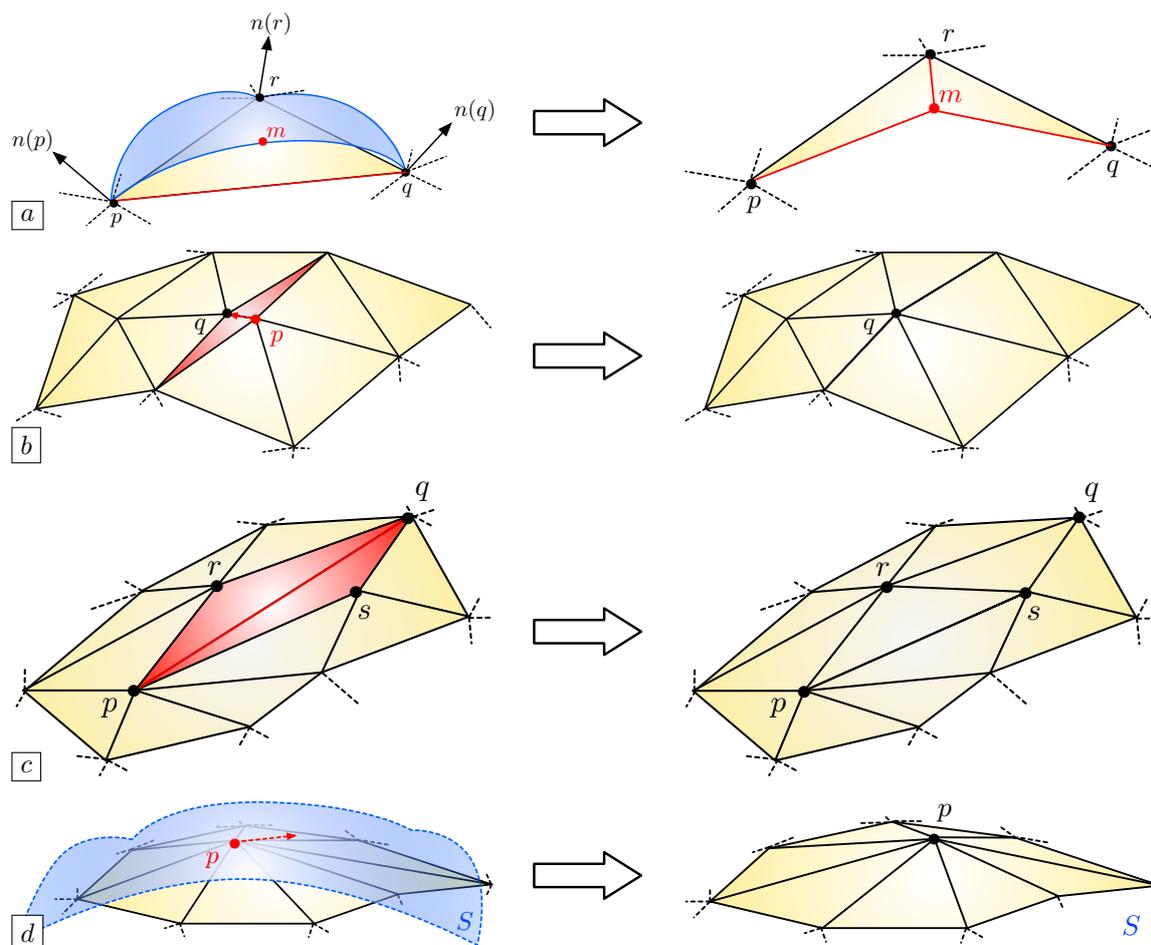


Figure 3.5: Illustration of the four operations involved in the remeshing strategy of Section 3.2.4; (a) Split of the “long” edge  $pq$  into two edges  $pm$  and  $mq$ , where the new point  $m$  is introduced on  $S$ ; (b) Collapse of the “short” edge  $pq$ ; (c) Swap of the edge  $pq$  between the triangles  $pqr$  and  $pqs$  for the alternate configuration, featuring the edge  $rs$  and the triangles  $rsp$  and  $rsq$ ; (d) Relocation of the vertex  $p$  which slides along  $S$ .

### 3.3 Numerical framework

In this section, we present numerical examples illustrating the main features of the proposed evolution [Algorithm 2](#). After a short description of our practical implementation in [Section 3.1](#), the first example in [Section 3.4](#) deals with a simple situation meant to appraise its accuracy, that of a region  $G(t)$  on the unit sphere  $\mathbb{S}^2$  in  $\mathbb{R}^3$  evolving according to its conormal vector field. In [Section 3.5](#) we turn to the numerical simulation of the physical evolution of a fire front, which is driven by the geometric features of the front and those of the underlying landscape  $S$ .

As presented in [Section 3.2](#), the numerical implementation of the examples proposed in the subsequent sections relies on a surface triangulation  $\mathcal{T}$  of the ambient surface  $S \subset \mathbb{R}^d$ , which is modified between the consecutive steps  $n = 0, \dots$  of the evolution process. Typically, various scalar and vector fields are defined and handled on  $S$ , such as level set functions  $\phi : S \rightarrow \mathbb{R}$  for regions  $G \subset S$  or velocity fields  $V : S \rightarrow \mathbb{R}^d$ . They are discretized as Lagrange  $\mathbb{P}_1$  finite element functions on  $\mathcal{T}$ , i.e. they are continuous, and their restriction to each triangle  $T \in \mathcal{T}$  is affine. As such, these quantities are characterized by their values at the vertices of  $\mathcal{T}$  and their evaluation at other points  $x \in S$  is realized by linear interpolation from these data. The numerical experiments discussed in the next [Sections 3.4](#) and [3.5](#) are executed on a regular laptop Apple MacBookPro 18,3 (M1 Pro chip) with 10 cores and 16 GB of memory. The code implementation has been carried out using the Rodin library, discussed in detail in [Chapter 5](#).

### 3.4 Motion in the direction of the conormal vector field

This first example aims to evaluate the efficiency of our numerical [Algorithm 2](#) in the context of a simple motion where an analytical solution is available. The ambient surface  $S$  is the unit sphere  $\mathbb{S}^2 \subset \mathbb{R}^3$ ; it is equipped with the spherical coordinates  $(\alpha, \beta)$  centered at the origin  $0$ , which induce the following (non injective) parametrization:

$$\begin{aligned} \sigma : [0, 2\pi] \times [0, \pi] &\rightarrow \mathbb{S}^2 \\ (\alpha, \beta) &\mapsto (\cos \alpha \sin \beta, \sin \alpha \sin \beta, \cos \beta). \end{aligned}$$

In this setting, we wish to track the region  $G(t) \subset S$  evolving from the upper spherical cap  $\mathcal{C}_a$  with azimuth angle  $a \in (0, \frac{\pi}{2})$ ,

$$G(0) = \mathcal{C}_a := \sigma([0, 2\pi] \times [0, a]),$$

according to the conormal vector field to the interface  $\Sigma(t) = \partial G(t)$ :

$$\forall t \in (0, T), \quad x \in \Sigma(t), \quad V(t, x) = n_{\Sigma(t)}(x),$$

over the time period  $(0, T)$ , with  $T = \frac{\pi}{2} - a$ , see [Fig. 3.6](#). Note that  $V(t, x)$  can be conveniently extended to the whole surface  $S$  as:

$$V(t, x) = \nabla_S d_{G(t)}(x), \quad \text{for a.e. } x \in S, \quad (3.13)$$

where  $d_{G(t)}$  is the signed distance function [\(3.8\)](#) to  $G(t)$ , see e.g. [\[116\]](#) about this property and [Remark 3.1](#).

As we have mentioned, in this simple setting, all the features attached to the evolution of  $G(t)$  can be expressed analytically. Indeed, for  $t \in (0, T)$ , the deformed domain  $G(t)$  is the upper spherical cap with azimuth angle  $a + t$ :

$$G(t) = \mathcal{C}_{a+t} = \sigma([0, 2\pi] \times [0, a + t]). \quad (3.14)$$

In particular, since  $T = \frac{\pi}{2} - a$ , the final region  $G(T)$  coincides with the upper half-sphere. Besides, the geodesic signed distance function  $d_{G(t)}$  to  $G(t)$  has the following analytical expression:

$$\forall (\alpha, \beta) \in [0, 2\pi] \times [0, \pi], \quad d_{\mathcal{C}_a(t)}(\sigma(\alpha, \beta)) = \beta - t - a. \quad (3.15)$$

Finally, let us observe that, in this very specific case where the normal component of the velocity field  $V(t, x)$  identically equals 1, the exact solution  $\phi(t, x)$  to the evolution equation [\(3.5\)](#) when initialized with the signed distance function  $\phi(0, \cdot) = d_{G(0)} = d_{\mathcal{C}_a}$  is given by:

$$\phi(t, x) = d_{\mathcal{C}_a(t)}(x), \quad (3.16)$$

in particular,  $\phi(t, \cdot)$  is the signed distance function to  $G(t)$  for all  $t > 0$ .

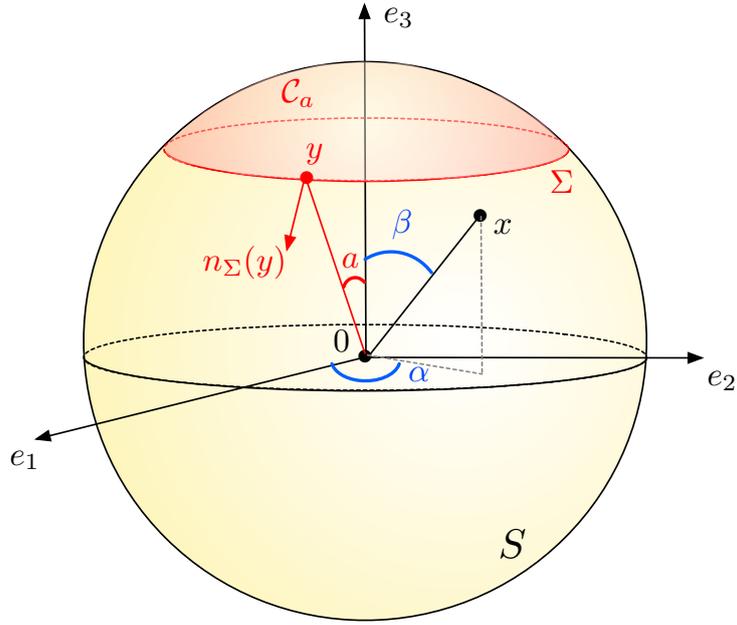


Figure 3.6: Setting of the motion of a region  $G(t)$  within the unit sphere  $\mathbb{S}^2$  along the conormal vector  $n_{\Sigma(t)}$  considered in [Section 3.4](#)

In our numerical simulation of the motion of  $G(t)$  by means of [Algorithm 2](#), the time interval  $(0, T)$  is discretized into  $N$  subintervals of the form  $(t^n, t^{n+1})$ , where  $0 \leq n \leq N - 1$ ,  $t^n = n\Delta t$ , and  $\Delta t$  is a “small” time step. In order to appraise the convergence of our strategy, we measure the least-square difference  $\mathcal{E}$  between the numerical level set function  $\phi^N$  for the tracked region  $G^N$  (that is, the outcome of the calculation of the signed distance function  $d_{G^N}$  with the help of the final mesh  $\mathcal{T}^N$  of  $S$ ) at the final iteration  $N$ , and the exact value  $\phi(T, \cdot)$  of the solution to [\(3.9\)](#), given by [\(3.15\)](#) and [\(3.16\)](#):

$$\mathcal{E} := \left( \int_S |\phi^N(x) - \phi(T, x)|^2 \, ds(x) \right)^{1/2}; \quad (3.17)$$

thus,  $\mathcal{E}$  measures the potential accumulation of errors incurred by the various stages of [Algorithm 2](#) throughout the iterations  $n = 0, \dots$ , namely the inexact calculation of the signed distance function to the regions  $G^n$ , the error in the numerical resolution of the advection equation [\(3.9\)](#) between consecutive times  $t^n$  and  $t^{n+1}$  and the approximation of the geometry of  $G^n$  caused by its explicit discretization into the mesh  $\mathcal{T}^n$ .

We wish to study the behavior of the error  $\mathcal{E}$  in the limit where the time step  $\Delta t$  vanishes and the mesh is refined, i.e. the average size  $h$  of the edges of the mesh tends to 0. Since a fully explicit procedure is used for approximating the solution to [\(3.5\)](#) by that of a series of linear evolution equations, featuring a frozen in time velocity field (see [Section 3.1](#)), we anticipate that it is relevant to relate these two parameters as  $\Delta t = ch$ , where the parameter  $c$  plays the role of a CFL number, accounting for the number of mesh elements crossed by the interface  $\Sigma(t)$  during a single time step.

Several numerical simulations of the motion of  $G(t)$  are performed, associated with different values of  $c$  and  $h$ ; a few iterations of the evolution process in one of these instances are depicted on [Fig. 3.7](#), and the behavior of the error  $\mathcal{E}$  in the various considered situations is shown in [Fig. 3.8](#). As expected, for a fixed, “not too small” value of the ratio  $c$ ,  $\mathcal{E}$  tends to 0 as the mesh is refined (i.e. as  $h$  tends to 0). On the contrary, this convergence is not observed when  $c$  is “small”, which can be attributed to the large accumulation of numerical errors during the various stages of [Algorithm 2](#) when the algorithm has to perform a “too large” number of iterations with respect to the limited precision guaranteed by a fixed mesh size. The results of [Fig. 3.8](#) suggest to select a value of  $c$  within the range  $(\frac{1}{3}, 1)$ , so that at least  $\frac{1}{3}$  element is crossed at each iteration of the process. Note that identical (unreported) experiments taking place in the context of a planar ambient surface  $S$  show similar trends as regards the behavior of our strategy with respect to the values of  $c$  and  $h$ .

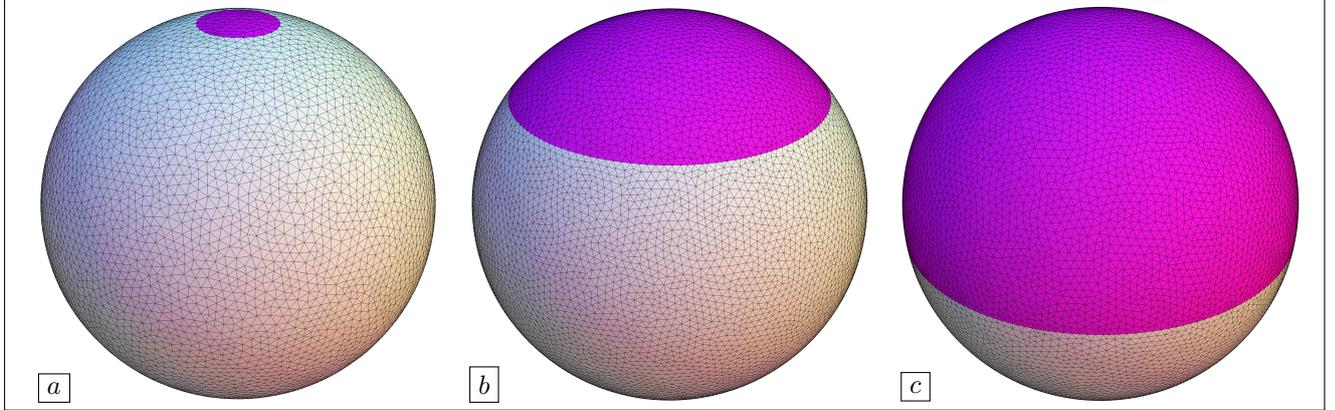


Figure 3.7: Evolution  $G(t)$  of the spherical cap  $\mathcal{C}_a$  under unit conormal velocity considered in Section 3.4; (a) Initial configuration  $G(0) = \mathcal{C}_a$ ; (b) Intermediate configuration  $G(t)$  at time  $t = \pi/4$ ; (c) Final region  $G(T)$  corresponding to the upper half-sphere.

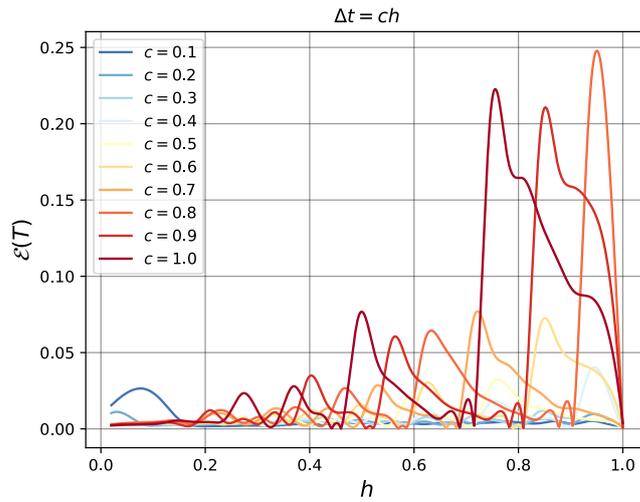


Figure 3.8: History of the least-square error  $\mathcal{E}$  in (3.17) for different values of the mesh size  $h$  and “CFL number”  $c$  in the conormal advection example of Section 3.4.

### 3.5 Wildland fire propagation

With the increasing global warming and the intensification of human activity, among other factors, wildfires have emerged as a significant concern in recent decades, and much endeavour has been made towards understanding, and ultimately predicting, their dynamics. The obvious limitations of practical experiments raise the need to develop accurate models and efficient numerical simulation algorithms, see e.g. [42, 44, 233, 290, 28, 273] and [243]. In this section, we show how the strategy of Section 3.2.1 for tracking the evolution of a region  $G(t)$  within a surface  $S \subset \mathbb{R}^3$  is a valuable tool in the numerical realization of a model for the simulation of such a phenomenon, which is driven by the geometric characteristics of the evolving front and of the landscape surface. We rely on the physical model of [41], which has been validated in e.g. [44, 145, 146] and is used in softwares such as **ForeFire** [145]. For the convenience of the reader, we provide a short, intuitive presentation of this model, referring to [41] for the details.

The ambient surface  $S \subset \mathbb{R}^3$  represents the topography of the landscape. It is defined as the graph of a function over a domain in the 2d horizontal plane, or, equivalently, in terms of the height function  $s : S \rightarrow \mathbb{R}$ :

$$\forall x = (x_1, x_2, x_3) \in S, \quad s(x_1, x_2, x_3) = x_3.$$

The ground slope vector  $p$  is the unit tangential vector field pointing in the direction of the largest variation of altitude:

$$\forall x \in S, \quad p(x) = \frac{\nabla_S s(x)}{|\nabla_S s(x)|},$$

and the local slope of the land is the angle between this vector and the tangential directions:

$$\alpha : S \rightarrow \left[-\frac{\pi}{2}, \frac{\pi}{2}\right], \quad \alpha = \arcsin(p \cdot e_3),$$

see Fig. 3.9. The evolving subdomain  $G(t) \subset S$  and its boundary  $\Sigma(t) = \partial G(t)$  represent the burnt region and the fire front, respectively, while the complement  $S \setminus G(t)$  stands for the vegetal stratum, i.e., the part of the land covered with vegetation that has not yet been ignited. The motion of  $\Sigma(t)$  is oriented along its conormal vector field:

$$\forall t \in (0, T), \quad x \in \Sigma(t), \quad V(t, x) = R(t, x)n_{\Sigma(t)}(x), \quad (3.18)$$

where the scalar component  $R(t, x) > 0$  is the rate of spread of the front.

The quantity  $R(t, x)$  depends on the geometric features of the landscape  $S$ , the fire front  $\Sigma(t)$ , and on the physical characteristics of the actual situation. These data, whose values are determined through measurements are the following:

- The rate of spread  $R_0 > 0$  of the burnt region in the absence of slope and wind (expressed in  $\text{ms}^{-1}$ );
- The velocity  $u_0$  of the combustion gas in the absence of slope (in  $\text{ms}^{-1}$ );
- The (dimensionless) ratio  $A > 0$  between the incident radiant energy and the ignition energy of the (wet) vegetal fuel;
- The velocity of the wind  $U : S \rightarrow \mathbb{R}^d$  (in  $\text{ms}^{-1}$ ).

The behavior of the rate of spread  $R(t, x)$  at some point  $x$  in the fire front  $\Sigma(t)$  depends on whether the flame is directed towards the burnt region or the vegetal stratum. This feature is measured by the so-called tilt angle  $\gamma(t, \cdot) : \Sigma(t) \rightarrow \mathbb{R}$ , whose values depend on the velocity of the wind  $U(x)$ , on the local slope  $\alpha(x)$ , and on the vector  $n_{\Sigma(t)}(x)$  via the following relation:

$$\forall x \in \Sigma(t), \quad \tan \gamma(t, x) = \tan \alpha(x) \cos \beta_p(t, x) + \frac{|U(x)|}{u_0} \cos \beta_U(t, x), \quad (3.19)$$

where  $\beta_p(t, x)$  is the angle between the ground slope vector  $p(x)$  and  $n_{\Sigma(t)}(x)$ , and  $\beta_U(t, x)$  is the angle between the wind  $U(x)$  and  $n_{\Sigma(t)}(x)$ . Roughly speaking, at some point  $x \in \Sigma(t)$ , the flame is tilted towards the burnt region if  $\gamma(t, x) \leq 0$  and towards the vegetal stratum if  $\gamma(t, x) \geq 0$ , see Fig. 3.10. The rate of spread  $R(t, x)$  then takes “small” values in the former situation, and large values in the latter one; its definition of  $R(t, x)$  brings into play the following two regimes:

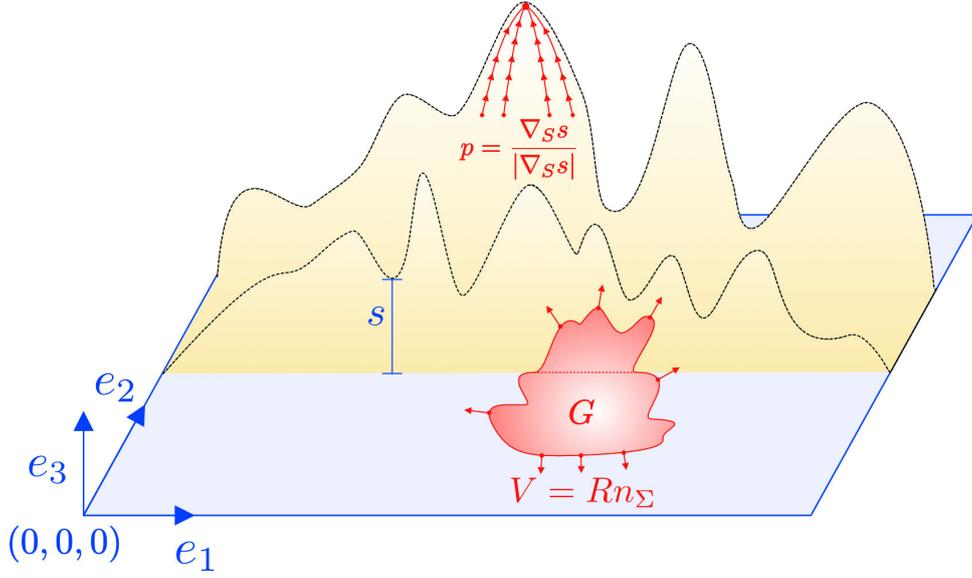


Figure 3.9: Illustration of the evolution of a fire on a topography. The slope vector  $p$  indicates the direction of greatest ascent as visualized by the arrows pointing towards the peaks. The burning region  $G$  is indicated by the yellow-red zone whose motion is prescribed by the arrows representing  $V$ .

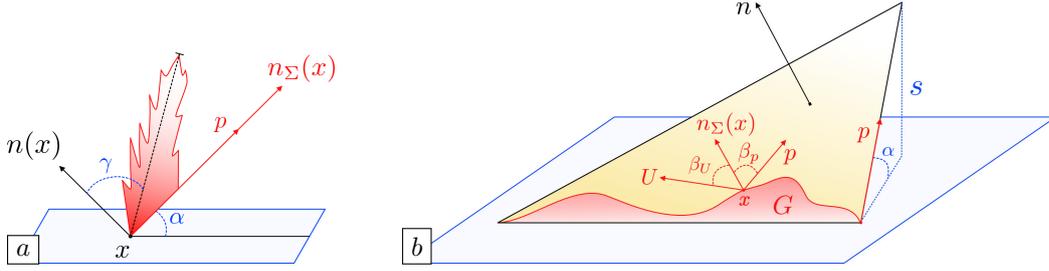


Figure 3.10: Illustration of the geometric relations involved in the computation of the rate of spread  $R$  at the level of one triangle  $T \in \mathcal{T}$ , in the fire propagation model of Section 3.5.

- *Case 1 (Slow backing fire spread)*  $\gamma(t, x) \leq 0$ . The flame axis at  $x$  is tilted towards the burnt region and  $R(t, x) = R_0$ .
- *Case 2 (Fast fire spread)*  $\gamma > 0$ . The flame axis at  $x$  is tilted towards the unburnt vegetation, thus accelerating the ignition of the latter and the expansion of the front;  $R(t, x)$  is given by:

$$R(t, x) = 0.5 \left( R_a(t, x) + \left( R_a(t, x)^2 + \frac{48R_0^2}{\cos \gamma(t, x)} \right)^{\frac{1}{2}} \right), \text{ where}$$

$$R_a(t, x) = R_0 + 12AR_0 \frac{1 + \sin \gamma(t, x) - \cos \gamma(t, x)}{\cos \gamma(t, x)} - 12R_0 \frac{1}{\cos \gamma(t, x)}. \quad (3.20)$$

We conduct three experiments of the evolution of a burning region within a landscape  $S$  with complex topography, associated to as many different scenarios. The surfaces  $S$  used in these examples are fictitious: they are generated as graphs of random non negative functions defined over the horizontal base  $[0, 50] \times [0, 50]$  – where the retained unit for spatial coordinates is the km; see Fig. 3.11 (a,c) for an illustration. For simplicity, we assume that a homogeneous vegetal stratum covers the entire terrain, so that the physical parameters  $R_0$ ,  $u_0$  and  $A$  are constant; their values are taken from [41] and are reported in Table 3.1. We also report in there the common values of the numerical parameters used in all three situations.

The use of Algorithm 2 in this context is straightforward: at each iteration  $n = 0, \dots$ , corresponding to the time  $t^n = n\Delta t$ , the computation of the velocity field  $V^n = V(t^n, \cdot)$  in (3.18) and (3.20) depends

Parameter	$R_0$	$u_0$	$A$	$\Delta t$	$h_{\max}$
Value	$2.7 \text{ ms}^{-1}$	$39.8 \text{ ms}^{-1}$	1.25	600 s	400 m

Table 3.1: Values of the parameters used in the experiments concerning the propagation of a fire front in Section 3.5.

on that of geometric quantities attached to  $S$  and  $G^n$ , that is conveniently realized with the help of the mesh  $\mathcal{T}^n$  of  $S$ , where  $G^n$  is explicitly discretized. Note that, in keeping with Remark 3.1, in this example again, the velocity field  $V(t, x)$  can be extended to the whole surface  $S$  by leveraging the same extension formula (3.13) for the conormal vector field  $n_{\Sigma(t)}$  as in the previous Section 3.4.

The landscape  $S$  considered in our first experiment is represented in Fig. 3.11 (b,d), and the initially burnt region  $G(0)$  is a surface disk with radius 1 km, centered at the point  $x = (x_1, x_2, x_3) \in S$  with coordinates  $(x_1, x_2) = (19.5, 19.5)$ . In this situation, the effect of the wind is neglected, i.e.  $U \equiv 0$ . We simulate the evolution of  $G(t)$  thanks to Algorithm 2 over the time period  $[0, T]$ , where  $T = 400$  mn, using the parameters reported in Table 3.1. A few intermediate meshes  $\mathcal{T}^n$  obtained in the course of the evolution are displayed in Fig. 3.12. In this experiment, the values of the rate of spread  $R(t, x)$  range between  $0.4$  and  $27.35 \text{ ms}^{-1}$ , with an average value of  $2.26 \text{ ms}^{-1}$ . The values presented here align with findings in [41] as well as in other studies like [312, 83], which also provide data on the mean, minimum, and maximum rates of spread. As expected from the formulas in (3.20), the fire spreads rapidly towards regions where the flame is tilted towards the unburnt region (typically in mountains).

From the technical vantage, let us point out that, at each iteration of the process, the complex landscape surface  $S$  and burnt region  $G^n$  are equipped with exact, high-quality meshes, which are refined in the vicinity of their sharp features. This allows for accurate calculations of quantities such as the conormal vector  $n_{\Sigma(t)}$  to the fire front as opposed to “classical” simulation methods and implementations in the fire dynamics literature, that use marker methods on Cartesian grids and projections or reconstructions of the fire front, see [147, 146, 145, 28].

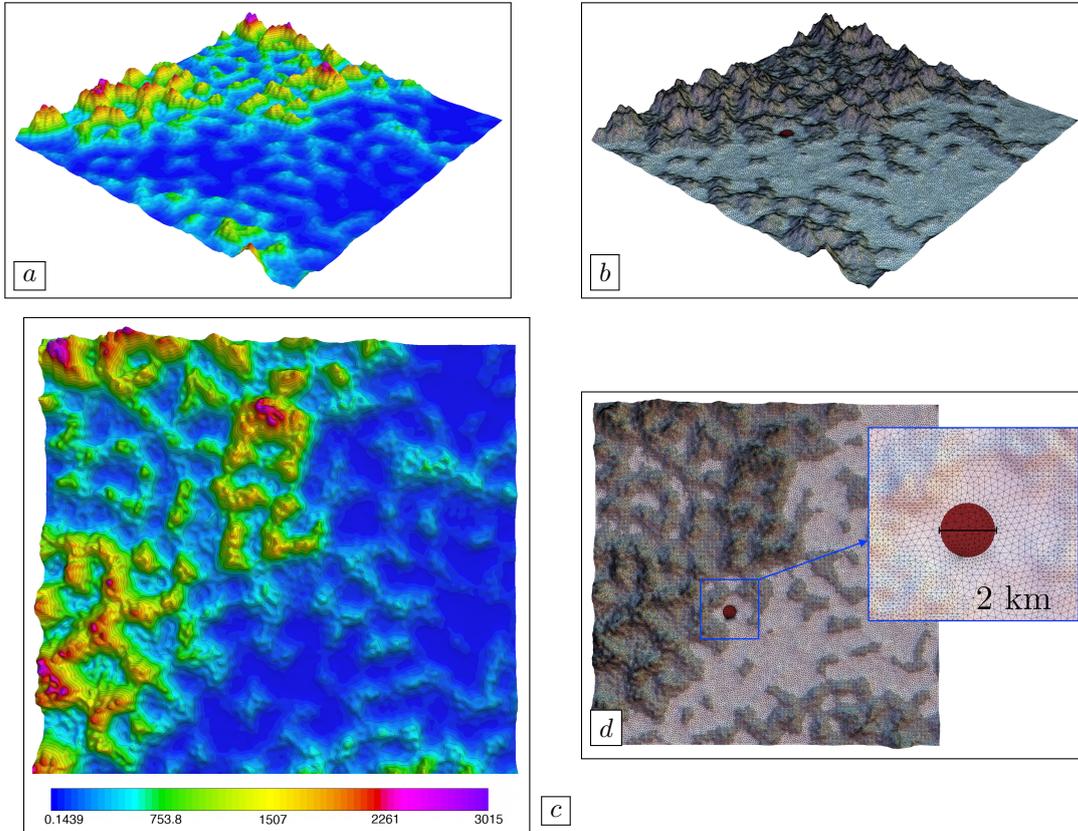


Figure 3.11: Landscape surface  $S$  used in the first experiment of Section 3.5; the initially burnt region  $G(0)$ , visualized in (b) and (d), is represented in red, and the color scale in (a) and (c) accounts for the height function  $s$ . The lower left corner of  $S$  is set to  $(0, 0, 0) \in \mathbb{R}^3$ .

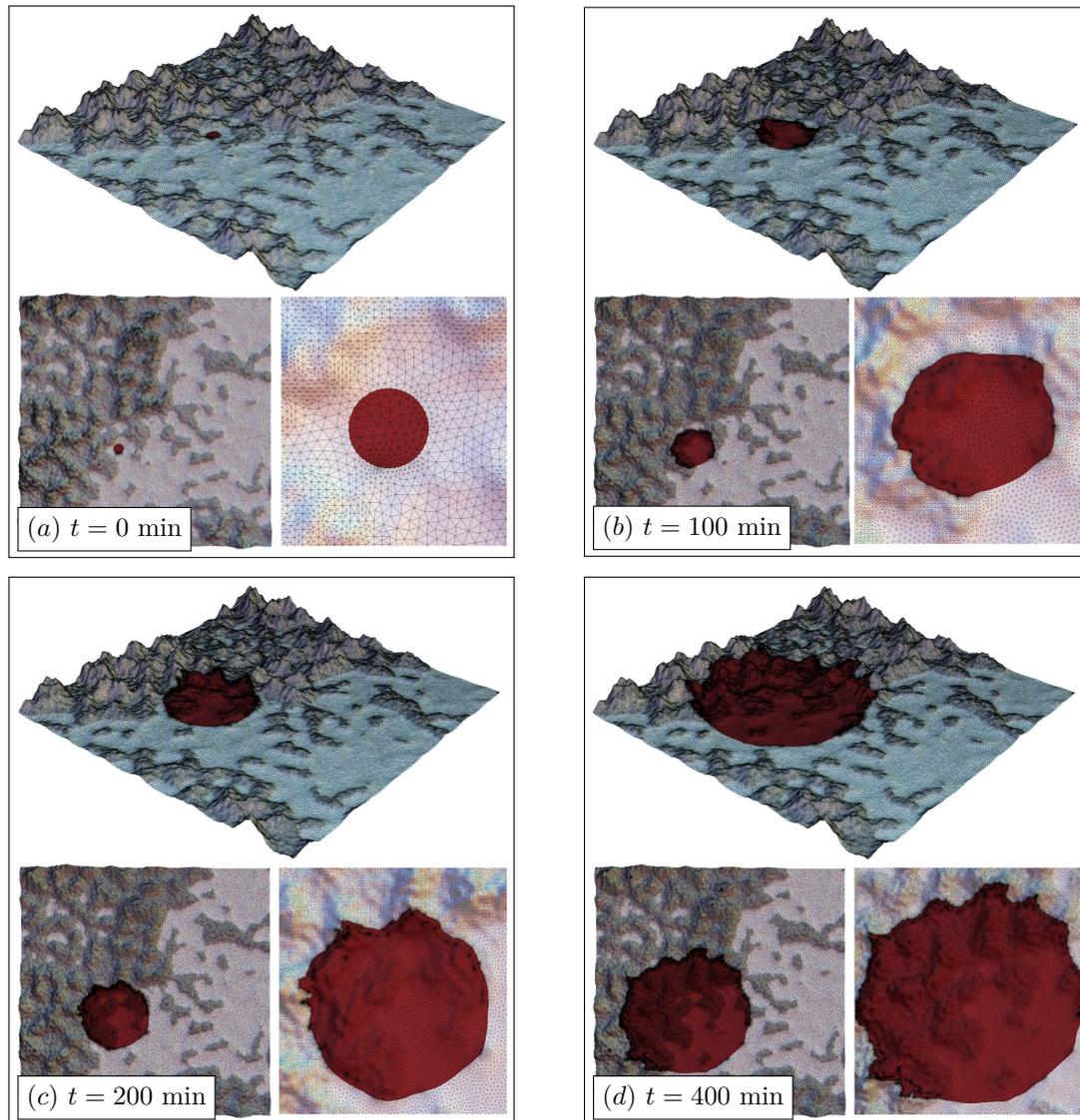


Figure 3.12: Snapshots of the first experiment of propagation of a fire front conducted in [Section 3.5](#).

We next turn to the study of a second scenario, aimed to appraise the behavior of our method in dealing with regions with multiple connected components. The landscape  $S$  is the same as that used in the first experiment, see again [Fig. 3.11](#) (b,d); the initial region  $G(0)$  is the reunion of four disjoint surface disks with radius 1 km, centered at the points of  $S$  whose horizontal coordinates equal  $(12.5, 12.5)$ ,  $(12.5, 37.5)$ ,  $(37.5, 12.5)$ ,  $(37.5, 37.5)$ , respectively. Still, wind is omitted.

The evolution of  $G(t)$  is tracked until the final time  $T = 500$  mn. A few snapshots of the evolution process are depicted on [Fig. 3.13](#). In the course of the evolution, the four initially burnt regions expand and eventually merge – the description of such complex topological changes being considerably eased by the use of our mesh evolution [Algorithm 2](#), which takes advantage of the level set method to deal with the update of the moving region.

We finally turn to a third experiment, where the motion of the burning region  $G(t)$  is influenced by the presence of a rotating wind. The landscape  $S$  is represented on [Fig. 3.14](#) (a), and the wind velocity  $U : S \rightarrow \mathbb{R}^d$  accounts for a rotation around the center of the landscape:

$$\forall x \in S, \quad U(x_1, x_2, x_3) = (x_2 - 25, -(x_1 - 25), 0).$$

A few snapshots of the simulation of the evolution of  $G(t)$  are presented in [Fig. 3.14](#); here the final time is  $T = 160$  mn. Notably, the algorithm successfully accounts for the rotational characteristics of the wind vector field and effectively managed the merging of burning regions resulting from such complex motions. Understandably enough, the burning region tends to expand more rapidly when the fire is tilted

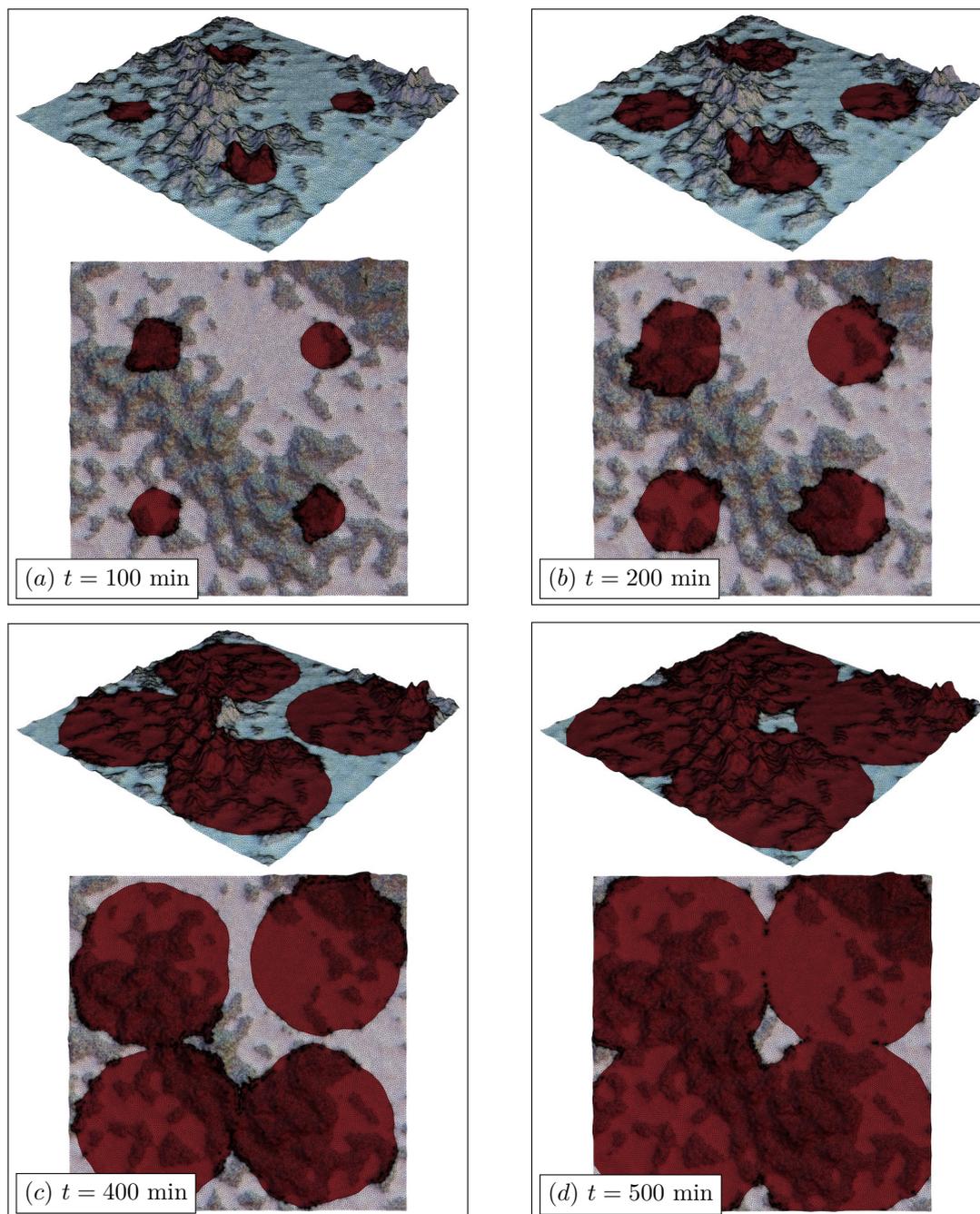


Figure 3.13: Snapshots of the second experiment of propagation of a fire front conducted in [Section 3.5](#).

towards the unburnt vegetation and the latter is located in the general direction of the wind. Moreover, the motion exhibits a faster rate of spread  $R(t, x)$  compared to that in the previous experiments, which is, again, caused by the wind. Again, the algorithm successfully manages to account for the merging of the various components of the burnt region.

Let us conclude this section with two general comments. Firstly, the meshes  $\mathcal{T}^n$  involved in each of the above three experiments contain on average about 240,000 triangles, and each iteration of our algorithm takes approximately 25 seconds, which suggests that our implementation proves quite efficient at simulating fire propagation in real time. Secondly, as far as the sensitivity of the simulation with respect to the choice of the parameters  $h$  and  $\Delta t$  is concerned, the same trends as in [Section 3.4](#) are observed: unreported numerical tests indicate that the general aspect of the burnt region does not depend very much on the choice of these parameters, as long as the time step  $\Delta t$  is of the order of the mesh size  $h$ .

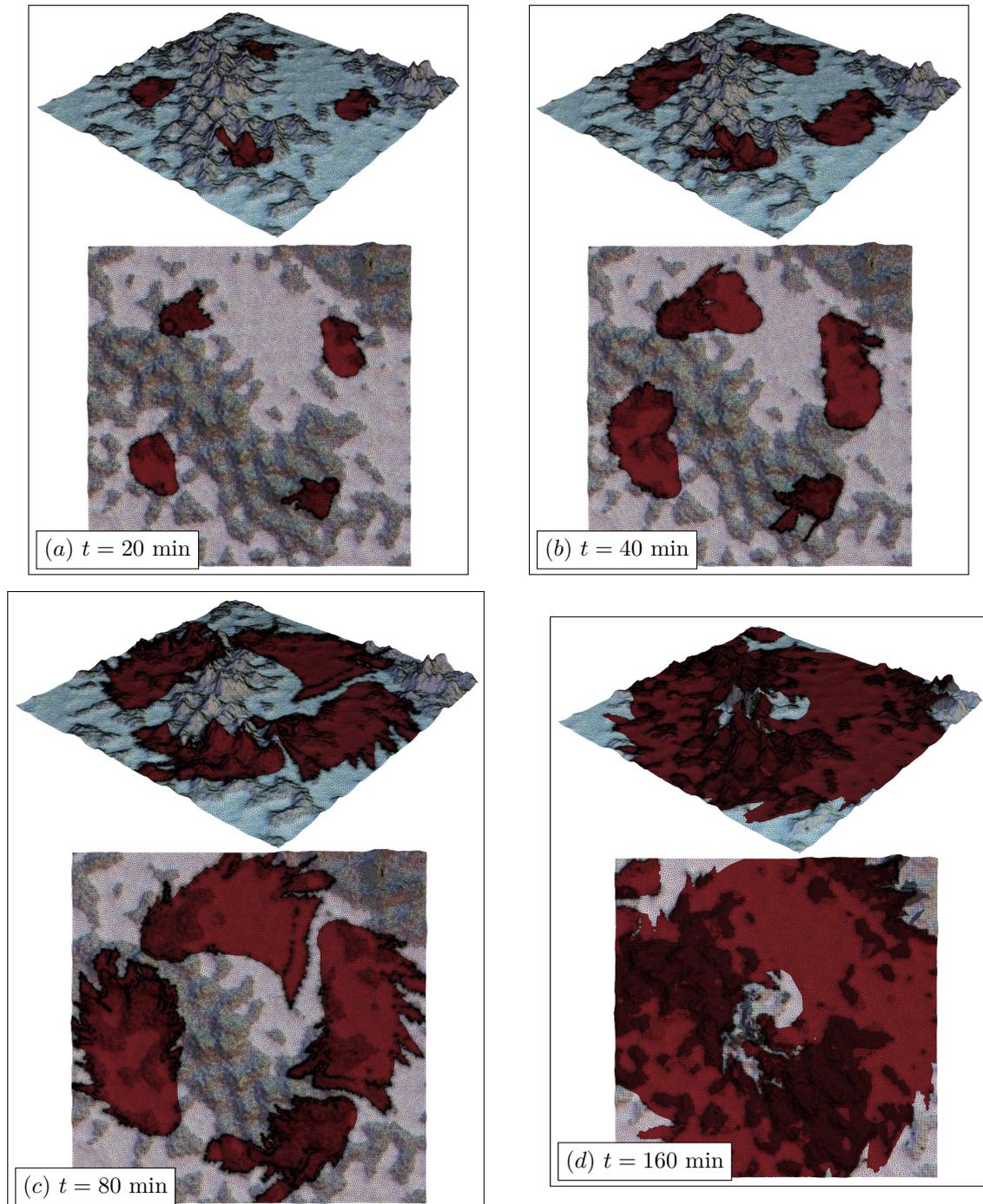


Figure 3.14: Snapshots of the third experiment of propagation of a fire front conducted in [Section 3.5](#).

# Chapter 4

## Shape and topology optimization of the regions supporting boundary conditions

---

The looming concerns caused by the scarcity of material resources and the blatant need to realize energy savings have made shape and topology optimization techniques just as topical as ever within the academic and industrial communities, where they find applications in fields so diverse as structural mechanics [52, 298], civil engineering and architecture [3, 48], fluid mechanics [6, 58], electromagnetism [158, 196, 218, 251], and biomedical engineering [191, 270, 351].

In the classical instances of such problems, the design under scrutiny is a “bulk” domain  $\Omega$  in  $\mathbb{R}^d$  ( $d = 2, 3$ ), which is optimized with respect to a performance criterion  $J(\Omega)$ , under constraints about e.g. its volume or perimeter. In applications,  $J(\Omega)$  usually depends on a physical “state” function  $u$ , characterized as the solution to a partial differential equation posed on  $\Omega$ . Most often, the regions of  $\partial\Omega$  supporting specific boundary conditions attached to the latter are imposed by the context, and are not subject to optimization.

The present chapter investigates optimal design problems where the variable is precisely one of those regions of  $\partial\Omega$  supporting a particular type of boundary conditions in the formulation of the physical problem at play. Among the various instances of such questions, let us mention the following:

- In electrostatics,  $\Omega$  represents a conductor, and the voltage potential  $u : \Omega \rightarrow \mathbb{R}$  within is the solution to the conductivity equation. It is grounded on a subset  $\Gamma_D$  of  $\partial\Omega$  and a flux  $g : \Gamma_N \rightarrow \mathbb{R}$  is imposed on a disjoint region  $\Gamma_N \subset \partial\Omega$ : these effects are modeled by a homogeneous Dirichlet condition on  $\Gamma_D$  and an inhomogeneous Neumann condition on  $\Gamma_N$ . The remaining part  $\Gamma$  of  $\partial\Omega$ , which is insulated from the outside, is subject to a homogeneous Neumann boundary condition. Although  $\Gamma_D$  and  $\Gamma_N$  are usually fixed, one may wish to minimize the amplitude of the electric field in  $\Omega$  with respect to their placement on  $\partial\Omega$ .
- In acoustics,  $u : \Omega \rightarrow \mathbb{R}$  is the sound pressure within a room  $\Omega$ , solution to the Helmholtz equation. The boundary  $\partial\Omega$  is decomposed into two regions  $\Gamma_N$  and  $\Gamma_R$ : Neumann boundary conditions are imposed on  $\Gamma_N$ , where an incoming wave undergoes perfect reflection, while  $\Gamma_R$  bears Robin boundary conditions, accounting for a partial absorption of the latter. One may then wonder how to arrange  $\Gamma_N$  and  $\Gamma_R$  within  $\partial\Omega$  to minimize the sound pressure in  $\Omega$ .
- In structural mechanics,  $\Omega$  is a mechanical piece, attached on a subset  $\Gamma_D$  of its boundary  $\partial\Omega$ , and submitted to surface loads  $g : \Gamma_N \rightarrow \mathbb{R}^d$ , applied on a disjoint region  $\Gamma_N \subset \partial\Omega$ ; the vector field  $u : \Omega \rightarrow \mathbb{R}^d$ , representing the displacement of the structure, is the solution to the linear elasticity system. Usually,  $\Gamma_D$  and  $\Gamma_N$  are given by the context, and only the remaining, traction-free

---

---

boundary  $\Gamma$  is optimized. However, it may be relevant to optimize the placement of the fixation region  $\Gamma_D$ , to minimize the displacement of the structure.

These questions fit in the general shape optimization framework of a subset  $G$  of a fixed ambient surface  $S \subset \mathbb{R}^d$ . Early studies in this context are devoted to the simulation of geometric flows within  $S$ , notably the mean curvature flow. In [230], this task is investigated thanks to the level set method, in a situation where  $S$  is equipped with a triangular mesh of  $S$ ; it is also considered in [84], where  $S$  itself is defined in an implicit way. The perhaps most natural instance of a physical optimal design problem posed on a surface  $S \subset \mathbb{R}^d$  certainly concerns the optimal reinforcement of a shell structure:  $S$  then plays the role of the midsurface, in which the optimized region  $G$  is that made of a stiffer material. Popular numerical strategies feature a fixed mesh of  $S$  which serves as the support of density-based topology optimization techniques [260, 318]. The level set method is also employed in such setting in [317], and it is coupled with a geometric optimization procedure for the midsurface  $S$  itself in [189]. On a different note, the work [338] combines the level set method with a conformal mapping strategy, reducing the surface  $S$ , and thereby the whole shape optimization problem, to a more classical planar situation.

In spite of their natural character and ubiquity in concrete applications, optimization problems of regions supporting the boundary conditions of a physical problem have been relatively seldom considered in the literature. Without anticipating too much on the more complete overviews of related contributions in the particular applications provided in Sections 4.7, 4.11, 4.13 and 4.14, let us solely mention that density-based topology optimization methods are prevailing in this context also, see e.g. [74, 228] in the context of the optimal design of a fixture system. The practice of the level set method on a fixed mesh of a computational domain is also observed in [333, 332] for the concurrent optimization of the shape and the Dirichlet region of a mechanical structure in 2D; see also [345] for similar ideas. Closer to the framework of the present chapter, the contribution [119] leverages the level set method on a fixed mesh of a box-shaped room  $\Omega$  to optimize the distribution of sound-soft and sound-hard materials on the boundary of the latter.

The problematic of optimizing regions supporting boundary conditions raises challenging issues from various perspectives. From the theoretical viewpoint, the realization of this task goes through the calculation of the derivative of such a function  $J(G)$ . This information is indeed the basic ingredient of the optimality conditions of the optimization problem. It is also the main ingredient of iterative optimization algorithms, starting from the simple, unconstrained gradient descent method to more advanced constrained optimization algorithms, such as those proposed in [130, 143, 313]. Interestingly, these derivatives can also be used to make the optimization problem “robust” with respect to small perturbations of the geometry of  $G$ , as in e.g. [10, 11, 235]. From the numerical viewpoint, optimal design problems of regions supporting boundary conditions raise in particular the need to track the possibly dramatic evolution of a region within an ambient surface – a task which is already notoriously difficult when the ambient medium is (a bounded domain of) the Euclidean space  $\mathbb{R}^d$ .

The present chapter is the natural continuation of the works [56, 63, 108]. The article [108] deals with the shape sensitivity of a function  $J(G)$  depending on a region  $G$  of the boundary of a domain  $\Omega$  bearing the boundary conditions of a state problem, in the spirit of the method of Hadamard: the derivative of  $J(G)$  with respect to “small”, diffeomorphic perturbations of  $G$  is considered. The situation where  $G$  is the support of Dirichlet conditions and the complement  $\partial\Omega \setminus \bar{G}$  is equipped with Neumann conditions is of particular interest in that work; indeed, the weakly singular behavior of the state function  $u$  in that case makes the treatment of shape derivatives particularly difficult – a fact which was previously acknowledged in [151]. To alleviate this issue, an approximation of the state problem is proposed, which lends itself to simpler calculation and numerical treatment, see Section 4.2.3 below for a brief presentation. This material paves the way to a numerical algorithm for the shape optimization of the region  $G$ . The article [56] deals with singular perturbations of  $G$ , at the theoretical level: asymptotic formulas are derived for the solution  $u$  to the conductivity equation in the case where homogeneous Neumann boundary conditions are replaced by homogeneous Dirichlet equations (and vice-versa) in a “vanishing” zone  $\omega_\varepsilon \subset \partial\Omega$ . A preliminary application of these results to the device of a notion of topological derivative for functions  $J(G)$  depending on regions  $G$  bearing boundary conditions was described in Chapter 3. The latter stands at the numerical level; elaborating on the ideas of [14], it introduces a body-fitted mesh evolution method to track dramatic motions of a region within a fixed ambient surface, which efficiently combines the level set method with remeshing algorithms. Note that this algorithm was used in the very recent work [234] with the purpose to optimize Neumann eigenvalues on the unit 3D sphere.

In the present chapter, we leverage these ingredients to propose a general shape and topology optimization workflow for a region  $G \subset \partial\Omega$  supporting the boundary conditions attached to a partial differential equation posed on the fixed ambient domain  $\Omega$ . Our strategy combines (adapted versions of)

the notions of shape and topological derivatives to appraise the sensitivity of a function  $J(G)$  with respect to small, diffeomorphic perturbations of  $\partial G$  and to singular perturbations, via the addition of a small surfacic disk  $\omega_\varepsilon$ , respectively. From the theoretical vantage, one of our contributions is to provide formal calculation methods for both types of derivatives. These are detailed within a simple setting based on the conductivity equation, and they can be adapted to treat novel situations. More precisely, our calculation method for the shape derivative of  $J(G)$  allows readily to treat a whole gammut of problems in the more intricate contexts of acoustics and structural mechanics. The calculation of a topological derivative for such a function  $J(G)$  is slightly more subtle, and we focus on the description of the needed adaptations of our methods to achieve this purpose. Interestingly, this chapter illustrates two different forays of asymptotic analysis in the realm of shape and topology optimization: on the one hand, these concepts are used to smooth a singular transition between two zones bearing different boundary conditions, leading to a simplified calculation of shape derivatives. On the other hand, it allows to investigate singular perturbations of a smooth background problem, via the introduction of a “small” zone where boundary conditions are altered.

## 4.1 Mathematical framework

Let  $\Omega$  be a smooth bounded domain in  $\mathbb{R}^d$  ( $d = 2, 3$ ). We consider a model shape and topology optimization problem of the form:

$$\min_{G \subset \partial\Omega} J(G), \quad (\mathcal{P})$$

where  $J$  is an objective function of a region  $G \subset \partial\Omega$ . The treatment of the problem  $(\mathcal{P})$  relies on the sensitivity of the function  $J(G)$  with respect to “small” variations of  $G$ . We will introduce two complementary means to appraise differentiation with respect to a boundary region:

- In [Section 4.2](#), we adapt the classical boundary variation method of Hadamard to the context of a boundary region  $G$ . This paves the way to a notion of shape derivative, accounting for the sensitivity of  $J(G)$  with respect to small perturbations of the boundary of  $G$ .
- In [Section 4.3](#), we define a suitable version of the concept of topological derivative, which appraises the sensitivity of  $J(G)$  with respect to the addition of a “small” surfacic disk to  $G$ .

In essence, these approaches extend the concepts of shape and topological derivatives to address optimization involving the regions bearing boundary conditions of a physical problem, which are essential tools in shape analysis.

## 4.2 Computation of shape derivatives with respect to variations in the support of boundary conditions

In this section, we precise the mathematical framework of our adopted notion of shape derivatives. Throughout, we assume that  $\Omega \subset \mathbb{R}^d$  is a smooth, fixed domain with boundary  $\partial\Omega$ . We assume that the boundary contains a regular region  $G$  bearing a certain type of boundary conditions in the formulation of a physical problem of interest. We denote  $\Sigma := \partial G$ .

### 4.2.1 Hadamard’s boundary variation method on a domain contained within a surface

We consider the problem of computing the shape derivative of the criterion  $J(G)$  featured in the problem  $(\mathcal{P})$ . Similar to the previous works, we rely on an adapted version of the classical Hadamard’s boundary variation method, involved in the notion of shape derivative of a function depending on a domain in  $\mathbb{R}^d$ ; see e.g. [\[22, 116, 185, 247, 301\]](#). We consider variations of the region  $G$  in the form:

$$G_\theta := (\text{Id} + \theta)(G), \quad \text{where } \theta \in W^{1,\infty}(\mathbb{R}^d, \mathbb{R}^d), \quad \|\theta\|_{W^{1,\infty}(\mathbb{R}^d, \mathbb{R}^d)} < 1, \quad (4.1)$$

as illustrated in [Fig. 4.1](#). Furthermore, we consider the set  $\Theta^\parallel$  of tangential vector fields to  $\partial\Omega$ , i.e.

$$\Theta^\parallel := \{\theta \in W^{1,\infty}(\mathbb{R}^d, \mathbb{R}^d), \|\theta\|_{W^{1,\infty}(\mathbb{R}^d, \mathbb{R}^d)} < 1 \mid \theta \cdot n_{\partial\Omega} = 0\}. \quad (4.2)$$

This leads to the following definition of shape differentiability for a function  $J$  of the bounded region  $G$ .

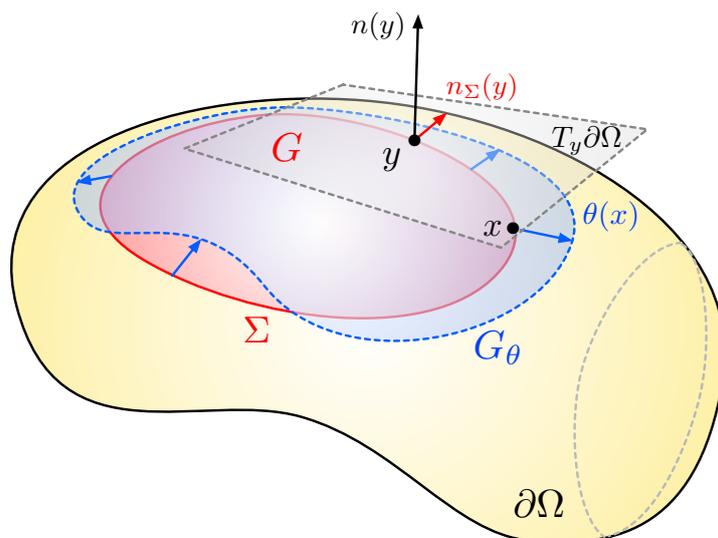


Figure 4.1: Optimization of the region  $G$  of the boundary  $\partial\Omega$  of a shape by the method of Hadamard.

**Definition 4.1.** *The criterion  $J(G)$  is shape differentiable at  $G \subset \partial\Omega$  if the underlying mapping  $\theta \mapsto J(G_\theta)$ , from  $W^{1,\infty}(\mathbb{R}^d, \mathbb{R}^d)$  into  $\mathbb{R}$  is Fréchet differentiable at  $\theta = 0$  and  $\theta \cdot n_{\partial\Omega} = 0$ . The shape derivative of  $J'(G)(\theta)$  is the Fréchet derivative and the following expansion holds:*

$$J(G_\theta) = J(G) + J'(G)(\theta) + o(\theta), \quad \text{where } \frac{o(\theta)}{\|\theta\|_{W^{1,\infty}(\mathbb{R}^d, \mathbb{R}^d)}} \xrightarrow{\theta \rightarrow 0} 0. \quad (4.3)$$

**Remark 4.1.** *When the considered variations  $G_\theta$  of the region  $G \subset \partial\Omega$  in (4.1) are defined from tangential vector fields  $\theta \in \Theta^\parallel$ , they account for a sliding displacement of  $G$  so that the ambient surface  $\partial\Omega$  remains unaltered. Note that this operation does not exactly leave  $\partial\Omega$  invariant. This behavior is only observed at “first order” in terms of  $\theta$ ; the definition of the variation  $G_\theta$  like in Chapter 2 or a definition of the variation involving the flow of the tangential vector field  $\theta$  would enforce strictly this property, as in the so-called velocity method discussed in [301]. For simplicity, and since both approaches yield the same notion of first-order derivative of a function  $J(G)$ , we ignore this technicality in the following.*

**Remark 4.2.** *On a different note, variations (4.1) of a region  $G \subset \partial\Omega$  can be defined from non tangential vector fields  $\theta$  (i.e.  $\theta \cdot n_{\partial\Omega} \neq 0$ ), resulting in an additional “normal” motion of the surface  $\partial\Omega$  itself. This allows to optimize jointly the shape of the ambient boundary  $\partial\Omega$  and that of the region  $G$ , as in the previous work [108]. In practice, as we shall see, one can compute the shape derivative  $J'(G)(\theta)$ , introduced in Definition 4.1 is a particular case of the classical shape derivative  $J'(\Omega)(\theta)$ , defined in Definition 1.1. The precise statement justifying this fact is stated in Lemma 2.5, where choosing a vector field on  $\mathbb{R}^d$  tangent everywhere to  $\partial\Omega$  yields the non-dependence on the normal part.*

Similar to the classical results (see Section 1.2), under mild assumptions, the shape derivative  $J'(G)(\theta)$  of a function  $J(G)$  of the region  $G \subset \partial\Omega$  turns out to be of the form

$$J'(G)(\theta) = \int_{\partial G} f_G \theta \cdot n_{\partial G} \, ds, \quad (4.4)$$

where  $n_{\partial G}$  denotes the outward pointing unit normal on  $\partial G$  and the scalar field  $f_G : \partial G \rightarrow \mathbb{R}$  depends on the region  $G$  and the considered objective function  $J(G)$ . Depending on the nature of the latter, the expression of  $f_G$  may involve the solution to one or several boundary value problems attached to  $\partial\Omega$  and  $G$ , see for instance (4.4) below. The structure (4.4) readily yields a descent direction for  $J(G)$  as  $\theta = -f_G n_{\partial G}$ . Indeed, we have:

$$J'(G)(\theta) = - \int_{\partial G} f_G^2 \, ds < 0.$$

Hence, substituting into the definition of shape derivative we have:

$$J(G_{t\theta}) = J(G) - tJ'(G)(\theta) + o(t) < J(G),$$

for small  $t > 0$ . Introducing a pseudo-time  $t$ , one considers the evolution  $G(t)$  of the region starting from an initial guess  $G^0 \subset \partial\Omega$ , under the effect of the velocity field

$$V(t, x) = -f_{G(t)}n_{\partial G(t)},$$

where  $f_G$  is defined in (4.4). This resulting sequence of regions  $G(t)$  smoothly decreases the value of the criterion  $J(G)$  until a (local) minimizer of  $(\mathcal{P})$  is attained.

The Hadamard method, which relies on variations of a region  $G$  in the form of (4.1), does not permit all types of topological changes: separate parts of the boundary  $\partial G$  of the optimized region  $G$  may collide and merge, but no hole can emerge inside another part of the ambient surface  $\partial\Omega$ . Since the problem  $(\mathcal{P})$  has multiple local minimizers, the optimization process is very sensitive to the initial design. In Section 4.3, we will see how the topological sensitivity can be used to alleviate this issue.

Before moving on to the next sections, let us mention that the perhaps most simple examples of functionals depending on a boundary region  $J(G)$  are the area  $\text{Area}(G)$  and contour  $\text{Cont}(G)$  of  $G$ , which are respectively defined by:

$$\text{Area}(G) = \int_G ds, \text{ and } \text{Cont}(G) = \int_{\partial G} d\sigma. \quad (4.5)$$

The next proposition provides the shape derivatives of these quantities, in a slightly more general context.

**Proposition 4.1.** *Let  $G$  be smooth region of the boundary  $\partial\Omega$ . Then,*

(i) *For any smooth function  $f : \mathbb{R}^d \rightarrow \mathbb{R}$ , the functional  $J(G)$  defined by:*

$$J(G) := \int_G f ds$$

*is shape differentiable; its shape derivative reads, for any tangential deformation  $\theta$ :*

$$J'(G)(\theta) = \int_{\partial G} f\theta \cdot n_{\partial G} d\sigma.$$

(ii) *For any smooth function  $g : \mathbb{R}^d \rightarrow \mathbb{R}$ , the functional  $K(G)$  defined by:*

$$K(G) := \int_{\partial G} g ds$$

*is shape differentiable; its shape derivative reads, for any tangential deformation  $\theta$ :*

$$K'(G)(\theta) = \int_{\partial G} (\nabla_{\partial\Omega} g \cdot n_{\partial G} + \kappa g)\theta \cdot n_{\partial G} d\sigma,$$

*where  $\kappa := \nabla_{\partial\Omega} \cdot (n_{\partial G})$  is the mean curvature of  $\partial G$ .*

## 4.2.2 Smoothed interfaces for weakly-singular problems

Beyond the classical issues associated to the computation of shape derivatives, there is in particular one problematic that often appears when optimizing the criterion in  $(\mathcal{P})$ . Consider the case where,  $\Gamma$  supports homogeneous Neumann conditions,  $G$  supports homogeneous Dirichlet conditions, and  $u_G$  is solution to the boundary value problem:

$$\begin{cases} -\Delta u_G = f & \text{in } \Omega, \\ u_G = 0 & \text{on } G, \\ \frac{\partial u_G}{\partial n_{\partial\Omega}} = 0 & \text{on } \Gamma. \end{cases} \quad (4.6)$$

An application of the classical Lax-Milgram theorem yields that the solution  $u_G$  is in  $H^1(\Omega)$ . Furthermore, given that the boundary is smooth, the classical elliptic regularity theory ensures that  $u_G$  has  $H^2$  regularity

except near the transition zone  $\Sigma := \overline{G} \cap \overline{\Gamma}$ . In this zone, where the boundary condition changes type,  $u_G$  fails to enjoy  $H^2$  regularity. Furthermore, an extensive mathematical theory exists in the literature, devoted to the precise study of the weakly singular behavior of  $u_G$  in these regions. For a comprehensive treatment, we refer the interested reader to the monographs [110, 172, 213]. In this context, we say that the solution  $u_G$  is called weakly singular, meaning it belongs to  $H^1(\Omega)$  but not  $H^2(\Omega)$ . This poses a problem when computing the shape derivative of  $J(G)$ . For instance, using the formal C ea’s method (see [22, 78] for an overview), which relies on the smoothness of  $u_G$ , results in an erroneous expression of the shape derivative in this context. Even if one were to avoid using a formal method like C ea’s method, it would still be necessary to characterize the precise singular behavior of the solution  $u_G$  at the transition zones, as done in [108]. However, even with a successful characterization of the singular behavior, there is no guarantee that the final expression would be readily usable from a numerical perspective. For a detailed discussion of the additional drawbacks associated with the weakly singular behavior, we refer the reader to [108].

To circumvent the issues articulated by the weakly singular behavior of  $u_G$ , we consider trading the exact solution  $u_G$  for an approximate regular solution  $u_{G,\varepsilon}$  in the definition of  $J(G)$ . In a nutshell, the “sharp” transition between the boundary conditions on  $G$  and  $\Gamma$  is replaced by a “smoothed” version over a thin tubular neighborhood of  $\Sigma$ . In this manner, the exact optimization problem is replaced (P) by its regularized counterpart:

$$\min_{G \subset \partial\Omega} J_\varepsilon(G) = \int_{\Omega} j(u_{G,\varepsilon}) \, dx,$$

where  $j : \mathbb{R} \rightarrow \mathbb{R}$ , and  $u_{G,\varepsilon}$  is solution to the boundary value problem:

$$\begin{cases} -\Delta u_{G,\varepsilon} = f & \text{in } \Omega, \\ \frac{\partial u_{G,\varepsilon}}{\partial n_{\partial\Omega}} + h_{G,\varepsilon} u_{G,\varepsilon} = 0 & \text{on } \Gamma \cup G. \end{cases} \quad (4.7)$$

In this formulation, the coefficient  $h_{G,\varepsilon} : \partial\Omega \rightarrow \mathbb{R}$  is defined by:

$$\forall x \in \partial\Omega, \quad h_{G,\varepsilon}(x) = \frac{1}{\varepsilon} h \left( \frac{d_G^{\partial\Omega}(x)}{\varepsilon} \right),$$

where  $d_G^{\partial\Omega}$  denotes the signed distance function to  $G$  on  $\partial\Omega$  (see Definition 2.5), and  $h \in C^\infty(\mathbb{R})$  satisfies:

$$0 \leq h \leq 1, \quad h \equiv 1 \text{ on } (-\infty, -1), \quad h(0) > 0, \quad h \equiv 0 \text{ on } [1, \infty). \quad (4.8)$$

Additionally, the weak formulation for the solution  $u_{G,\varepsilon} \in H^1(\Omega)$  to (4.7) is:

$$\forall v \in H^1(\Omega), \quad \int_{\Omega} \nabla u_{G,\varepsilon} \cdot \nabla v \, dx + \int_{\Gamma \cup G} h_{G,\varepsilon} u_{G,\varepsilon} v \, ds = \int_{\Omega} f v \, dx$$

In contrast to  $u_G$ , for a fixed value of  $\varepsilon > 0$ , the solution  $u_{G,\varepsilon}$  enjoys  $H^2$  regularity in a neighborhood of  $\Sigma$ , due to the smoothness of  $\Omega$  and  $h_\varepsilon$ . This is a consequence of the standard regularity theory for elliptic equations [62]. The regularized function  $u_{G,\varepsilon}$  and the regularized criterion  $J_\varepsilon(G)$  are valid approximations by the following result, proved in [108].

**Theorem 4.1.** *Suppose that the boundary of  $\partial\Omega$  around  $\Sigma$ . The function  $u_{G,\varepsilon}$  converges to  $u_G$  strongly in  $H^1(\Omega)$ , and the following estimate holds:*

$$\|u_{G,\varepsilon} - u_G\|_{H^1(\Omega)} \leq C_s \varepsilon^s \|f\|_{L^2(\Omega)},$$

for any  $0 < s < \frac{1}{4}$ , where the constant  $C_s$  depends on  $s$ .

One can then calculate the shape derivative of the regularized criterion. The details of the proof of this calculation can be found in §4.4 of [108]. The result is presented in the following proposition.

**Proposition 4.2.** *The functional  $J_\varepsilon(G)$  is shape differentiable at  $G$ , and its shape derivative reads:*

$$J'_\varepsilon(G)(\theta) = -\frac{1}{\varepsilon^2} \int_{\Gamma \cup G} h' \left( \frac{d^\partial \Omega}{\varepsilon} \right) \theta(\pi_\Sigma(x)) \cdot n_\Sigma(\pi_\Sigma(x)) u_{G,\varepsilon}(x) p_{G,\varepsilon}(x) \, ds(x). \quad (4.9)$$

where  $\pi_\Sigma(x)$  denotes the projection of  $x$  onto  $\Sigma$  (see for instance [Definition 2.6](#), and where  $p_{G,\varepsilon} \in H^1(\Omega)$  is the solution to the adjoint problem:

$$\begin{cases} -\nabla \cdot (\gamma \nabla p_{G,\varepsilon}) = -j'(u_{G,\varepsilon}) & \text{in } \Omega, \\ \gamma \frac{\partial p_{G,\varepsilon}}{\partial n_{\partial\Omega}} + h_\varepsilon p_{G,\varepsilon} = 0 & \text{on } G \cup \Gamma. \end{cases} \quad (4.10)$$

Similarly to  $u_{G,\varepsilon}$ , we have the following result for  $J_\varepsilon(G)$ . Again, see §4.4 of [\[108\]](#) for the hypotheses and proof.

**Theorem 4.2.** *The approximate (classical) shape derivative  $J'_\varepsilon(\Omega)$  converges to its exact counterpart  $J'(\Omega)$  in the sense that:*

$$\sup_{\theta \in \Theta} |J'_\varepsilon(\Omega)(\theta) - J'(\Omega)(\theta)| = 0.$$

**Remark 4.3.** *In general, the regularization approach demonstrated in this section can be easily adapted to different contexts, such as the linearized elasticity system and the Helmholtz equation. In [Sections 4.11 to 4.14](#), we will use this approach to regularize the transition zones that exhibit weakly singular behavior.*

### 4.2.3 Approximate formulas for the numerical implementation of the regularized shape derivative

While the mathematical formulation of the derivative [\(4.9\)](#) is satisfactory from a theoretical point of view, its numerical evaluation poses certain challenges. First, one must calculate the projection  $\pi_\Sigma(y)$  onto  $\Sigma$  for each point  $y \in \Gamma \cup G$ . Secondly, [\(4.9\)](#) does not conform to the structure of [\(4.4\)](#), preventing the direct identification of a descent direction and the use of advanced optimization algorithms such as the one introduced in [\[130\]](#). Fortunately, these challenges can be addressed by making some hypotheses to derive approximate formulas, thereby simplifying their numerical computation.

**Remark 4.4.** *The derivation relies on certain principles of Riemannian geometry, including concepts like the pullback of the Riemannian metric and the use of Jacobi fields, some of which are discussed in [Chapter 2](#). While a simpler derivation could be achieved by employing an alternative version of the change of variables theorem ([Theorem 2.5](#)), we choose to present this approach as it precisely defines the nature of the approximations.*

**Theorem 4.3 (Coarea formula on Riemannian manifolds).** *Suppose  $(M, g)$  is a Riemannian manifold, and  $\phi : M \rightarrow \mathbb{R}$  is a smooth function with no critical points. Then for any measurable function  $f : M \rightarrow \mathbb{R}$  we have*

$$\int_M f(x) dV_g = \int_{\mathbb{R}} \left( \int_{\phi^{-1}(t)} \frac{f(x)}{|\nabla_M \phi(x)|} dV_{\iota_t^* g} \right) dt,$$

where  $\iota_t^* g$  denotes the induced metric on the level surface  $\phi^{-1}(t) \subset M$ , i.e. the pullback of the metric  $g$  by the inclusion map  $\iota_t : \phi^{-1}(t) \rightarrow M$ .

*Proof.* A more general statement of this proposition can be found in the textbook [\[80\]](#), or the seminar notes [\[250\]](#). ■

Since the derivative  $h'$  has compact support inside  $[-1, 1]$ , the integrand of the formula [\(4.9\)](#) has compact support inside the tubular neighborhood  $U_\varepsilon = \{x \in \partial\Omega \text{ s.t. } d^\partial \Omega(x, \Sigma) < \varepsilon\}$ . Recall that in this

tubular neighborhood,  $|\nabla_{\partial\Omega} d_G^{\partial\Omega}| = 1$ . An application of [Theorem 4.3](#) yields:

$$\begin{aligned} J'_\varepsilon(G)(\theta) &= -\frac{1}{\varepsilon^2} \int_{U_\varepsilon} h' \left( \frac{d_G^{\partial\Omega}(y)}{\varepsilon} \right) \theta(\pi_\Sigma(y)) \cdot n_\Sigma(\pi_\Sigma(y)) u_{G,\varepsilon}(y) p_{G,\varepsilon}(y) ds(y) \\ &= -\frac{1}{\varepsilon^2} \int_{-\varepsilon}^\varepsilon \int_{\{d_G^{\partial\Omega}=t\}} h' \left( \frac{t}{\varepsilon} \right) \frac{\theta(\pi_\Sigma(y)) \cdot n_\Sigma(\pi_\Sigma(y)) u_{G,\varepsilon}(y) p_{G,\varepsilon}(y) ds(y)}{|\nabla_{\partial\Omega} d_G^{\partial\Omega}(y)|} dt \\ &= -\frac{1}{\varepsilon^2} \int_{-\varepsilon}^\varepsilon \int_{\{d_G^{\partial\Omega}=t\}} h' \left( \frac{t}{\varepsilon} \right) \theta(\pi_\Sigma(y)) \cdot n_\Sigma(\pi_\Sigma(y)) u_{G,\varepsilon}(y) p_{G,\varepsilon}(y) ds(y) dt. \end{aligned}$$

Consider the map  $F_t(x) = \exp_x(tn_\Sigma(x))$ , which, according to [Theorem 2.3](#), is a diffeomorphism from  $\Sigma$  onto its image. Applying a change of variables (see [Theorem 2.5](#)), we obtain:

$$J'_\varepsilon(G)(\theta) = - \int_{-\varepsilon}^\varepsilon \int_\Sigma h' \left( \frac{t}{\varepsilon} \right) \theta(x) \cdot n_\Sigma(x) u_{G,\varepsilon}(F_t(x)) p_{G,\varepsilon}(F_t(x)) dV_{F_t^*g}(x) dt \quad (4.11)$$

The volume form of the pullbacked metric  $F_t^*g$ , is defined by:

$$(F_t^*g)_x(v_1, v_2) = d_x F_t^*(g_{F_t(x)})(v_1, v_2) = g_{F_t(x)}(d_x F_t(v_1), d_x F_t(v_2)).$$

This leads us to calculate the differential  $d_x F_t : T_x \Sigma \rightarrow T_{F_t(x)} U$ , accounting for the volume distortion resulting from spulling back the metric. Fortunately, this is a problem already addressed in Riemannian geometry (c.f. [\[322\]](#)). To address this, let  $\gamma(t) : (-\varepsilon, \varepsilon) \rightarrow U$  be the geodesic with  $\gamma(0) = x$  and  $\gamma'(0) = n_\Sigma(x)$ , then we have by [Lemma 2.2](#), that the differential  $d_x F_t : T_x \Sigma \rightarrow T_{F_t(x)} \Sigma$ , is given by:

$$\forall v \in T_x \Sigma, \quad d_x F_t(v) = J_v(t),$$

where  $J_v$  is a Jacobi field that satisfies the equation:

$$D_t^2 J_v + R(J_v, \gamma')\gamma' = 0,$$

with initial conditions:

$$J_v(0) = v, \quad \text{and} \quad D_t J_v(0) = \nabla_v n_\Sigma(x),$$

where  $R$  is the Riemann curvature tensor, the operator  $D_t$  denotes covariant differentiation along the curve  $\gamma(t)$ , and  $\nabla_v n_\Sigma$  is the covariant derivative of  $n_\Sigma$  in the direction  $v$ . we rely on two assumptions:

1. The shape derivative remains unaffected by the curvature on  $U_\varepsilon$  when traversing the path  $\gamma(t)$ . This involves setting  $R = 0$  in the equation above, rendering the Jacobi field equal to the identity map.
2. The functions  $u_{G,\varepsilon}$  and  $p_{G,\varepsilon}$  remain constant along the rays  $t \mapsto F_t(x)$ , meaning that for all  $(x, t) \in \Sigma \times (-\varepsilon, \varepsilon)$ , we have  $u_{G,\varepsilon}(x) \approx u_{G,\varepsilon}(F_t(x))$  and  $p_{G,\varepsilon}(x) \approx p_{G,\varepsilon}(F_t(x))$ . In practical terms, this equates to a zero-order approximation.

By rearranging the order of integration and utilizing these assumptions, we can express:

$$\begin{aligned} J'_\varepsilon(G)(\theta) &\approx -\frac{1}{\varepsilon^2} \int_\Sigma \theta(x) \cdot n_\Sigma(x) u_{G,\varepsilon}(x) p_{G,\varepsilon}(x) \int_{-\varepsilon}^\varepsilon h' \left( \frac{t}{\varepsilon} \right) dt d\sigma(x) \\ &= -\frac{1}{\varepsilon} \int_\Sigma \theta(x) \cdot n_\Sigma(x) u_{G,\varepsilon}(x) p_{G,\varepsilon}(x) dt d\sigma(x), \end{aligned} \quad (4.12)$$

where the second line follows from the fact that:

$$\int_{-\varepsilon}^\varepsilon h' \left( \frac{t}{\varepsilon} \right) dt = \varepsilon \int_{-1}^1 h'(t) dt = \varepsilon(h(1) - h(-1)) = \varepsilon.$$

Equation [\(4.12\)](#) is the desired approximated formula, which satisfies the typical Hadamard structure [\(4.4\)](#) and no longer features the projection  $\pi_\Sigma$  mapping.

**Remark 4.5.** *This approximation can be easily adapted to other physical contexts and boundary condition types.*

#### 4.2.4 A model optimization problem of the regions supporting boundary conditions of a thermal mechanics problem

In order to illustrate the use setting of the shape optimization algorithm [Algorithm 2](#), we consider a situation in thermal mechanics. In this example,  $\Omega$  represents a 3D mechanical part, in which cooling effects are considered. The boundary  $\partial\Omega$  is decomposed  $\partial\Omega = G \cup \Gamma \cup$ , where:

- The temperature is fixed to zero 0 on  $G$ ;
- The domain  $\Omega$  is insulated from the outside on  $\Gamma$ .

Furthermore, we assume the existence of a heat source  $f \in L^2(\Omega)$  within the medium. In this situation, the temperature  $u_G$  within  $\Omega$  is the solution to [\(4.6\)](#). We aim to identify the geometry of the region  $G$  which minimizes the average temperature within the domain  $\Omega$ , namely, we consider the shape optimization problem involving the functional  $J(G)$ :

$$J(G) = \frac{1}{|\Omega|} \int_{\Omega} u_G \, dx + \ell \, \text{Area}(G),$$

where  $\ell > 0$  serves as a penalization parameter on the area of  $G$ . The approximated shape derivative  $J_{\varepsilon}(G)$  of the resulting functional can be approximated by the following formula; see [\(4.12\)](#).

$$J'_{\varepsilon}(G)(\theta) \approx -\frac{1}{\varepsilon^2} \int_{\partial G} u_{G,\varepsilon} p_{G,\varepsilon} \theta \cdot n_{\partial G} \, d\sigma(x) + \ell \int_{\partial G} \theta \cdot n_{\partial G} \, d\sigma(x), \quad (4.13)$$

where  $u_{G,\varepsilon} \in H^1(\Omega)$  is the solution to [\(4.7\)](#) and  $p_{G,\varepsilon} \in H^1(\Omega)$  is the solution to the adjoint problem [\(4.10\)](#).

The numerical implementation of these concepts fits in the general context of the evolution of a region  $G(t)$  within a surface  $S \subset \mathbb{R}^d$  discussed in [Chapter 3](#). Utilizing the same techniques discussed in that chapter, we can give a modified version of [Algorithm 2](#) in [Algorithm 3](#). The domain  $\Omega$  is discretized by a tetrahedral mesh  $\mathcal{T}^n$ , changing every  $n$ -th iteration. The boundary surface mesh  $\mathcal{B}^n$  contains an explicit discretization of the region  $G^n$ . Introducing a pseudo-time  $t$ , one considers the evolution  $G(t)$  of the region starting from an initial guess  $G^0 \subset S$ , under the effect of the velocity field  $\theta$  inferred from [\(4.13\)](#).

---

**Algorithm 3:** Optimization of the region  $G$  bearing homogeneous Dirichlet boundary conditions.

---

**Input:** Mesh  $\mathcal{T}^0$  of  $\Omega$ , whose discretized boundary  $\mathcal{B}^0$  contains two submeshes  $\mathcal{B}_{\text{int}}^0$  of  $G^0$ , and  $\mathcal{B}_{\text{ext}}^0$  of  $\partial\Omega \setminus \overline{G^0}$ .

**for**  $n = 0, \dots, N - 1$  **do**

1. Compute the signed distance function  $d_{G^n}^{\partial\Omega}$  to  $G^n$  at the vertices of the mesh  $\mathcal{B}^n$  of  $\partial\Omega$ .

2. Compute the solutions  $u_{G,\varepsilon}$  and  $p_{G,\varepsilon}$ .

3. Infer a descent direction  $\theta^n$  of  $J_{\varepsilon}(G)$  using the expression of  $J'_{\varepsilon}(G^n)(\theta)$ .

4. For a fixed timestep  $\Delta t > 0$ , solve the advection equation

$$\begin{cases} \frac{\partial \phi}{\partial t}(t, x) + \theta^n(x) \cdot \nabla_{\partial\Omega} \phi(t, x) = 0 & \text{for } (t, x) \in (0, \Delta t) \times \partial\Omega, \\ \phi(0, x) = d_{G^n}^{\partial\Omega}(x) & \text{for } x \in \partial\Omega, \end{cases}$$

on the total mesh  $\mathcal{B}^n$  of  $\partial\Omega$ . A new level set function  $\phi^{n+1} = \phi(\Delta t, \cdot)$  is obtained for

$$G^{n+1} = \{x \in \partial\Omega, \phi^{n+1}(x) < 0\}.$$

5. From the datum of  $\phi^{n+1}$  at the vertices of  $\mathcal{B}^n$ , create a new, high-quality mesh  $\mathcal{B}^{n+1}$  of  $\partial\Omega$  made of two submeshes  $\mathcal{B}_{\text{int}}^{n+1}$  and  $\mathcal{B}_{\text{ext}}^{n+1}$  for  $G^{n+1}$  and  $\partial\Omega \setminus \overline{G^{n+1}}$ , respectively.

**end**

**Output:** Mesh  $\mathcal{T}^N$  whose discretized boundary  $\mathcal{B}^N$  contains an explicit discretization  $\mathcal{B}_{\text{int}}^N$  of  $G^N$ .

---

We conduct two experiments. For the first experiment, we choose an average mesh size of  $h = 0.08$  and  $\ell = 0.01$ . A few intermediate designs obtained in the course of the optimization process are displayed

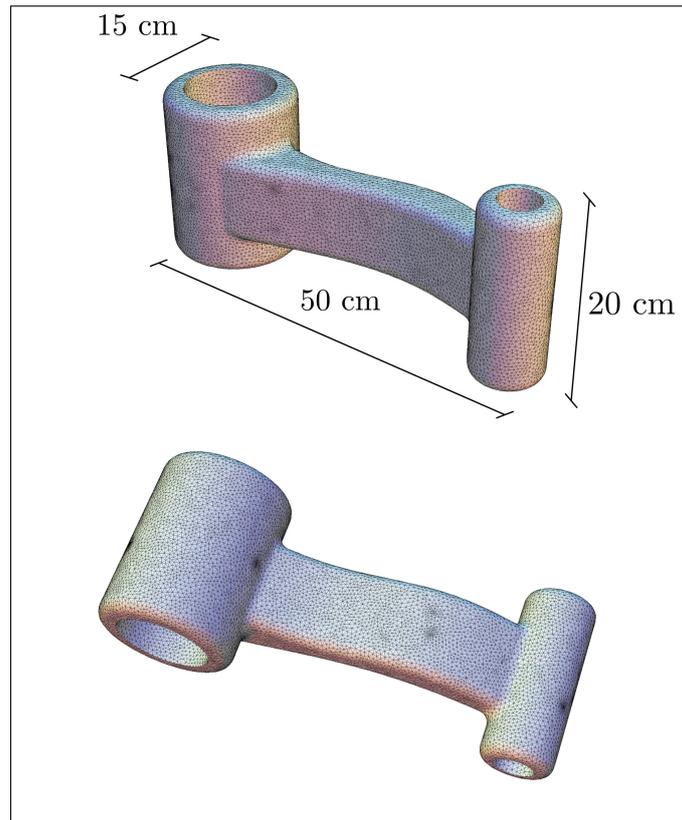


Figure 4.2: Illustration of the mechanical part  $\Omega$  where boundary conditions are optimized in [Section 4.2.4](#).

in [Fig. 4.3](#); the final design and the associated convergence history are reported in [??](#). For the second experiment, we have chosen  $h = 0.05$  and  $\ell = 0.05$ . The optimized region  $G$  resulting from this experiment and the convergence history are reported in [Fig. 4.4](#).

In both experiments, the values of the objective function  $J_\varepsilon(G)$  smoothly decrease to a local minimum. Interestingly, the optimization process starts by expanding the region  $G$  endowed with homogeneous Dirichlet boundary conditions in an attempt to minimize the mean temperature  $T_\varepsilon(G)$  within  $\Omega$ . Then, the algorithm attempts to distribute this region everywhere in  $S$  while avoiding the creation of patterns with large areas, which results in the creation of tree-like branches. This branching phenomenon agrees with typical results in the optimal design of thermal structures; which is rooted in homogenization theory [citeallaire2002shape](#), see also [\[144\]](#) and [Section 1.7](#).

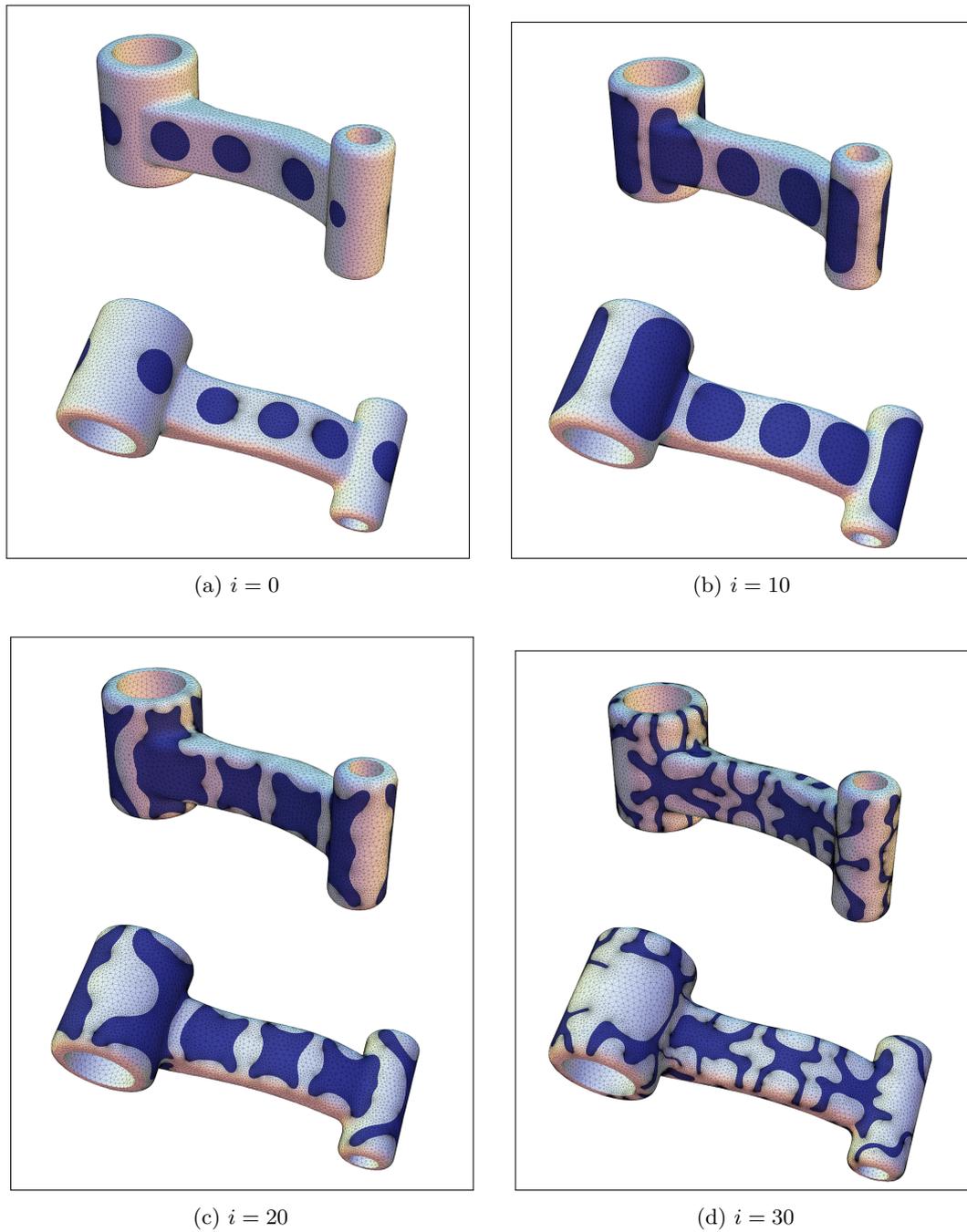
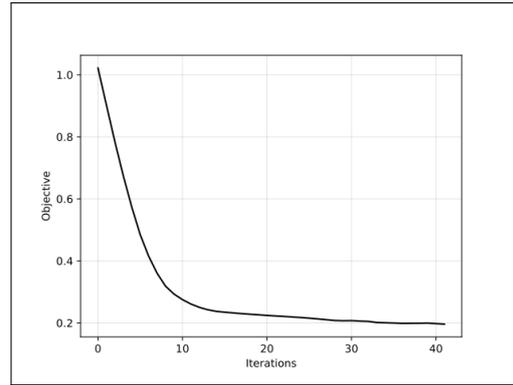


Figure 4.3: Boundary optimization process of the mechanical device considered in [Section 4.2.4](#) with penalization parameter  $\ell = 0.01$  and mesh size  $h_{\max} = 0.08$ . The optimized region  $G$  bearing homogeneous Dirichlet boundary conditions is represented in blue.



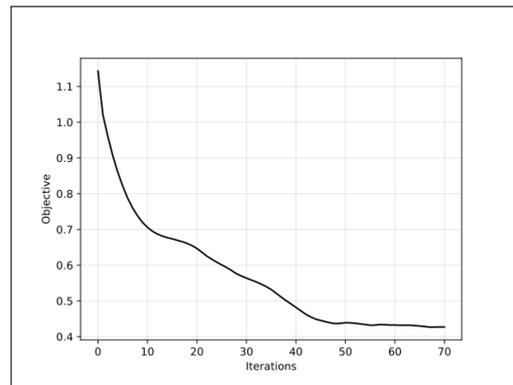
(a) Optimized design ( $n = 40$ ) in the boundary optimization process of the mechanical device considered in Section 4.2.4 with penalization parameter  $\ell = 0.01$  and mesh size  $h_{\max} = 0.08$ .



(b) Convergence history.



(c) Optimized design ( $n = 60$ ) in the boundary optimization process of the mechanical device considered in Section 4.2.4 with penalization parameter  $\ell = 0.01$  and mesh size  $h_{\max} = 0.05$ .



(d) Convergence history.

Figure 4.4: Optimized designs and convergence history.

### 4.3 Sensitivity with respect to topological variations in the support of boundary conditions

The notion of shape derivative introduced in Section 4.2 relies on Lipschitz diffeomorphic variations  $G_\theta$  of a region  $G \subset \partial\Omega$  of the form (4.1); in particular,  $G_{\tau\theta}$  and  $G$  share the same topology, see e.g. [141]. Hence, strictly speaking, the above notion of shape derivative does not capture the sensitivity of  $J(G)$  to topological changes in  $G$ . Admittedly, in practice, some topological changes may occur in the case of an algorithm such as Algorithm 3: separate parts of the boundary  $\partial G$  of the optimized region  $G$  may collide and merge by a slight abuse of the theoretical framework, but in any event, no hole can emerge inside  $G$ , and the number of connected components of  $G$  is fixed. In particular, the optimized design strongly depends on the topology of the initial guess (see [16]). Hence, to make the resolution procedure of (P) less sensitive to the existence of multiple local minimizers, we consider in this section a second, complementary means to optimize a region  $G$  of  $\partial\Omega$ . We assume that  $\Omega \subset \mathbb{R}^d$  is a smooth, fixed domain with boundary  $\partial\Omega$ . The boundary is decomposed into the disjoint union  $\partial\Omega = \bar{\Gamma} \cup \bar{G}$ , where  $\Gamma$  and  $G$ , are open, Lipschitz subsets of  $\partial\Omega$ .

#### 4.3.1 The topological derivative of replacing small regions of a set with another set

We consider the problem of computing the topological derivative of the function  $J(G)$  in the problem (P). Our objective is to gauge the sensitivity of the criterion  $J(G)$  when a small region of  $\Gamma$  is replaced with  $G$ . More precisely, let us denote by  $H$  the lower half-space of  $\mathbb{R}^d$ , defined by:

$$H := \{x = (x_1, \dots, x_d) \mid x_d < 0\}. \quad (4.14)$$

Additionally, let  $\mathbb{D}_\varepsilon$  be the  $\varepsilon$ -disk in  $\mathbb{R}^d$ , defined as:

$$\mathbb{D}_\varepsilon := \{x = (x_1, \dots, x_{d-1}, 0) \in \partial H \text{ s.t. } |x| < \varepsilon\}. \quad (4.15)$$

Let  $x_0 \in \Gamma$  be a given point; since the domain  $\Omega$  is smooth, there exists an open neighborhood  $\mathcal{O}$  of 0 in  $\mathbb{R}^d$  and a smooth diffeomorphism  $T : \mathcal{O} \rightarrow T(\mathcal{O})$  such that:

$$T(0) = x_0 \text{ and } T(H \cap \mathcal{O}) = \Omega \cap T(\mathcal{O}). \quad (4.16)$$

We then define the surfacic disk  $\omega_{x_0, \varepsilon}$  with center  $x$  and radius  $\varepsilon$  by:

$$\omega_{x_0, \varepsilon} := T(\mathbb{D}_\varepsilon), \quad (4.17)$$

and we consider the variations:

$$G_{x_0, \varepsilon} := G \cup \omega_{x_0, \varepsilon}, \quad \Gamma_{x_0, \varepsilon} := \Gamma \setminus \omega_{x_0, \varepsilon}. \quad (4.18)$$

See Fig. 4.5 for an illustration.

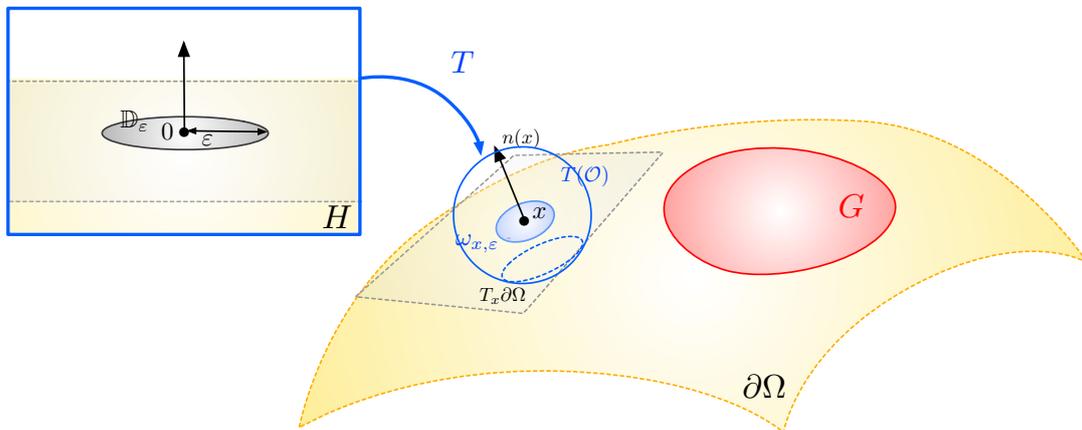


Figure 4.5: Variation  $G_{x_0, \varepsilon}$  in (4.18) of a surface region  $G$  obtained by addition of a “small” surfacic disk  $\omega_{x_0, \varepsilon}$ .

To make the analysis possible, it is necessary that  $\omega_{x_0,\varepsilon}$  lie “far” away from the interface  $\Sigma := \bar{\Gamma} \cap \bar{G}$  in the sense that, there exists a constant  $\delta > 0$  such that:

$$\text{dist}(\Sigma, \omega_{x_0,\varepsilon}) \geq \delta, \quad (4.19)$$

for all  $\varepsilon > 0$  small enough. Then we can make the following definition of topological derivative.

**Definition 4.2.** *The function  $J(G)$  has a topological derivative  $d_T(G)(x)$  at some point  $x_0 \in \partial\Omega \setminus \bar{G}$  if there exists a function  $\rho : \mathbb{R}_+ \rightarrow \mathbb{R}_+$  such that  $\rho(\varepsilon) \rightarrow 0$  as  $\varepsilon \rightarrow 0$ , and the following asymptotic expansion holds:*

$$J(G_{x_0,\varepsilon}) = J(G) + \rho(\varepsilon)d_T(G)(x_0) + o(\rho(\varepsilon)). \quad (4.20)$$

**Remark 4.6.** *At first glance, this definition may appear to depend on the choice of the local diffeomorphism  $T$  in (4.16). However, the analyses in the upcoming sections will demonstrate that this is not the case.*

**Remark 4.7.** *In general, establishing an expression for  $r_\varepsilon$  is difficult to when the boundary is not flat. The work [57] conducts an in-depth analysis assuming a curved boundary and shows that the curvature does not affect the resulting expression, for the case of the conductivity equation. To simplify the analysis, the assumptions (4.16) and (4.17) enable us to concentrate the analysis on a flat neighborhood around the point 0. See Fig. 4.6 illustrates the geometry of  $\Omega$  around  $\mathcal{O}$ . In this setting, the difference  $r_\varepsilon$  tends to 0:*

$$r_\varepsilon \xrightarrow{\varepsilon \rightarrow 0} 0 \text{ in } H^1(\Omega), \quad (4.21)$$

*an admitted fact, which is proved thanks to standard a priori estimates, see again [57]. Throughout the rest of this chapter, we denote  $\omega_\varepsilon := \omega_{0,\varepsilon}$  for simplicity.*

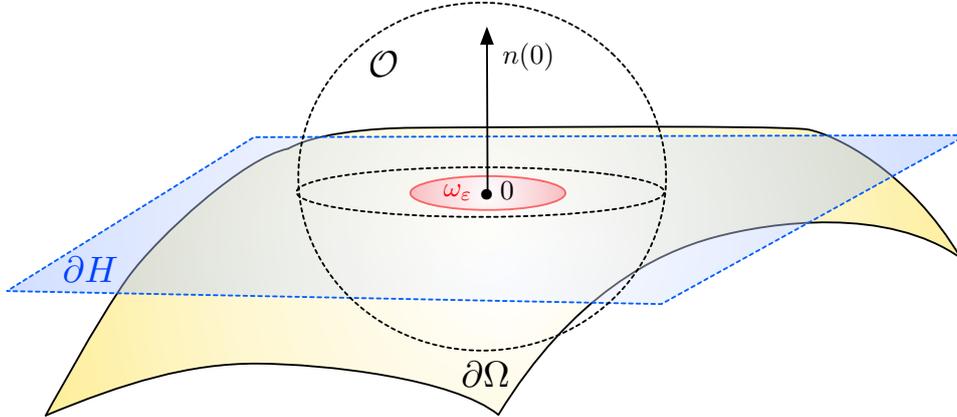


Figure 4.6: The boundary  $\partial\Omega$  is assumed to be flat in a neighborhood of the considered point 0.

### 4.3.2 Topological derivative of a criterion under the constraint of a boundary value problem

In this section, we introduce some general terminology and concepts that will be used throughout the remainder of this chapter. We assume that the domain  $\Omega$  accommodates a background solution  $u_0 \in V$ , which satisfies a boundary value problem with boundary conditions specified on  $G$  and  $\Gamma$ . Here,  $V$  denotes a Hilbert space of functions defined over  $\Omega$ , with values in  $\mathbb{R}^n$ , where  $n \in \mathbb{N}$ . We are particularly interested in shape functionals of the form:

$$J(G) = \int_{\Omega} j(u_0) \, dx, \quad (4.22)$$

where  $j \in C^2(\mathbb{R}^n, \mathbb{R})$  satisfies the growth conditions:

$$\exists K > 0, \quad \text{s.t.} \quad \forall u \in \mathbb{R}^n, \quad |j(u)| \leq K(1 + |u|^2), \quad |j'(u)| \leq K(1 + |u|), \quad |j''(u)| \leq K. \quad (4.23)$$

We can utilize the variations of  $G_{x_0, \varepsilon}$  and  $\Gamma_{x_0, \varepsilon}$  described in [Section 4.3](#) to define a perturbed solution  $u_\varepsilon \in V$  for the original boundary problem. Here, we provide a couple of examples demonstrating the perturbation.

**Example 4.1 (Poisson equation).** *Let  $n = 1$ ,  $V = H^1(\Omega)$ . Given  $f \in L^2(\Omega)$ , the solution  $u_0 \in V$  to the Poisson equation with homogeneous Dirichlet boundary conditions on  $G$  and homogeneous Neumann boundary conditions on  $\Gamma$  can be perturbed into the solution  $u_\varepsilon \in V$  characterized by the boundary value problem below:*

$$\begin{cases} -\Delta u_0 = f & \text{in } \Omega \\ u_0 = 0 & \text{on } G, \\ \partial_{n_{\partial\Omega}} u_0 = 0 & \text{on } \Gamma \end{cases}, \quad \begin{cases} -\Delta u_\varepsilon = f & \text{in } \Omega \\ u_\varepsilon = 0 & \text{on } G \cup \omega_{x_0, \varepsilon}. \\ \partial_{n_{\partial\Omega}} u_\varepsilon = 0 & \text{on } \Gamma \setminus \overline{\omega_{x_0, \varepsilon}} \end{cases}.$$

**Example 4.2 (Linear elasticity equation).** *Let  $n = d$ ,  $V = H^1(\Omega)^d$ . Given  $f \in L^2(\Omega)^d$  and  $g \in L^2(\partial\Omega)^d$ , the solution  $u_0 \in V$  to the linear elasticity equation with homogeneous Dirichlet boundary conditions on  $\Gamma_D$  and homogeneous Neumann boundary conditions on  $G$  can be perturbed into the solution  $u_\varepsilon \in V$  characterized by the boundary value problem below:*

$$\begin{cases} -\nabla \cdot (Ae(u_0)) = f & \text{in } \Omega \\ u_0 = 0 & \text{on } \Gamma_D, \\ Ae(u_0)n_{\partial\Omega} = 0 & \text{on } \Gamma \end{cases}, \quad \begin{cases} -\nabla \cdot (Ae(u_\varepsilon)) = f & \text{in } \Omega \\ u_\varepsilon = 0 & \text{on } \Gamma_D \\ Ae(u_\varepsilon)n_{\partial\Omega} = g & \text{on } \omega_{x_0, \varepsilon} \\ Ae(u_\varepsilon)n_{\partial\Omega} = 0 & \text{on } \Gamma \setminus \overline{\omega_{x_0, \varepsilon}} \end{cases}. \quad (4.24)$$

In these circumstances, we aim to determine the asymptotic behavior of the potential  $u_\varepsilon$  as  $\varepsilon \rightarrow 0$ , as well as that of the quantity of interest  $J(G)$ . Let us consider the criterion at  $G_{x_0, \varepsilon}$  for some fixed  $x \in \Gamma$ :

$$J(G_{x_0, \varepsilon}) = \int_{\Omega} j(u_\varepsilon) \, dx. \quad (4.25)$$

A simple application of the fundamental theorem of calculus yields:

$$j(u_\varepsilon) = j(u_0) + \int_0^1 j'(u_0 + t(u_\varepsilon - u_0)) \cdot (u_\varepsilon - u_0) \, dt. \quad (4.26)$$

The twice continuous differentiability of  $j$  allows us to expand the function  $t \mapsto j'(u_0 + t(u_\varepsilon - u_0))$  at  $t = 0$ :

$$j'(u_0 + t(u_\varepsilon - u_0)) = j'(u_0) + tj''(u_0)(u_\varepsilon - u_0, u_\varepsilon - u_0) + o(t^2|u_\varepsilon - u_0|). \quad (4.27)$$

Utilizing the growth conditions [\(4.23\)](#), we can rewrite the perturbed criterion [\(4.25\)](#) as:

$$J(G_{x_0, \varepsilon}) = J(G) + \int_{\Omega} \int_0^1 j'(u_0 + tr_\varepsilon) \cdot r_\varepsilon \, dt \, dx = J(G) + \int_{\Omega} j'(u_0) \cdot r_\varepsilon \, dx + o\left(\int_{\Omega} |r_\varepsilon| \, dx\right), \quad (4.28)$$

where  $r_\varepsilon := u_\varepsilon - u_0$ . This expression follows a structure very similar to that of [\(4.20\)](#). The challenge here is to find suitable  $\rho(\varepsilon)$  and  $d_T(G)(x_0)$  that satisfy:

$$\rho(\varepsilon)d_T(G)(x_0) = \int_{\Omega} j'(u_0) \cdot r_\varepsilon \, dx, \quad \forall x \in \Gamma. \quad (4.29)$$

In the following sections, this issue will be addressed by establishing an integral representation for  $r_\varepsilon$ , involving the Green's function for the background problem, and by rewriting  $J(G_{x_0, \varepsilon})(x)$  in terms of an adjoint state. In general, the analyses that we perform on the different equations ([Sections 4.6 to 4.8](#)) follow a very similar pattern, with some minor variations depending on the interior condition and type of boundary conditions considered. To this end, in [Section 4.6](#) we will perform a fairly complete analysis for the model of the conductivity equation, whose background equation is very similar to those of the Helmholtz and linear elasticity cases, for which we only will sketch the proofs.

## 4.4 Sobolev spaces on the boundary of a domain

To a large extent, the asymptotic analysis in this chapter concern functions defined on the boundary  $\partial\Omega$  of the ambient domain  $\Omega$ , which naturally belong to suitable energy spaces, whose precise definitions are recalled in the present section; we refer for instance to [172, 237] about these issues.

For any real number  $0 < s < 1$ , the fractional Sobolev space  $H^s(\partial\Omega)$  is defined by:

$$H^s(\partial\Omega) = \{u \in L^2(\partial\Omega) \text{ s.t. } \|u\|_{H^s(\partial\Omega)} < \infty\},$$

$$\text{where } \|u\|_{H^s(\partial\Omega)}^2 := \|u\|_{L^2(\partial\Omega)}^2 + \int_{\partial\Omega} \int_{\partial\Omega} \frac{|u(x) - u(y)|^2}{|x - y|^{d-1+2s}} ds(x) ds(y).$$

By convention,  $H^0(\partial\Omega) = L^2(\partial\Omega)$ , and for  $0 < -s < 1$ ,  $H^{-s}(\partial\Omega)$  is the topological dual of  $H^s(\partial\Omega)$ .

Let now  $G$  be a Lipschitz open subset of  $\partial\Omega$ ; we shall use two types of fractional Sobolev spaces of functions on  $G$ :

- For any  $0 < s < 1$ , we denote by  $H^s(G)$  the space of restrictions of functions from  $H^s(\partial\Omega)$ , that is:

$$H^s(G) = \{U|_G \text{ for some } U \in H^s(\partial\Omega)\}, \text{ equipped with the quotient norm}$$

$$\|u\|_{H^s(G)} := \inf \{\|U\|_{H^s(\partial\Omega)}, U \in H^s(\partial\Omega), U|_G = u\}.$$

- For any  $0 < s < 1$ , we denote by  $\tilde{H}^s(G)$  the subspace of elements in  $H^s(\partial\Omega)$  with compact support in  $\bar{G}$ . It is equipped with the norm  $\|u\|_{\tilde{H}^s(G)} = \|u\|_{H^s(\partial\Omega)}$  induced by that of  $H^s(\partial\Omega)$ . Equivalently,  $H^s(G)$  is the space of functions  $u \in L^2(G)$  whose extension  $\tilde{u}$  by 0 to  $\partial\Omega$  belongs to  $H^s(\partial\Omega)$ .

As for the negative versions of these spaces,

- For  $0 < s < 1$ ,  $H^{-s}(G)$  is still defined as the space of distributions on  $G$  obtained by restriction of a distribution in  $H^{-s}(\partial\Omega)$ . The space  $H^{-s}(G)$  is naturally identified with the dual space of  $\tilde{H}^s(G)$  via the following duality: for all  $u \in H^{-s}(G)$ , and  $v \in \tilde{H}^s(G)$ ,

$$\langle u, v \rangle_{H^{-s}(G), \tilde{H}^s(G)} := \langle U, \tilde{v} \rangle_{H^{-s}(\partial\Omega), \tilde{H}^s(\partial\Omega)},$$

where  $U$  is any element in  $H^{-s}(\partial\Omega)$  such that  $U|_G = u$  and  $\tilde{v}$  is the extension of  $v$  by 0 to the whole  $\partial\Omega$ .

- For  $0 < s < 1$ ,  $\tilde{H}^{-s}(G)$  is again the subspace of the distributions in  $H^{-s}(\partial\Omega)$  with compact support in  $\bar{G}$ . This space is naturally identified with the dual of  $H^s(G)$  via the following duality: for all  $u \in \tilde{H}^{-s}(G)$ ,  $v \in H^s(G)$ ,

$$\langle u, v \rangle_{\tilde{H}^{-s}(G), H^s(G)} := \langle \tilde{u}, V \rangle_{H^{-s}(\partial\Omega), \tilde{H}^s(\partial\Omega)},$$

where  $\tilde{u}$  is the extension of  $u$  by 0 to the whole  $\partial\Omega$  and  $V$  is any element in  $H^s(\partial\Omega)$  such that  $V|_G = v$ .

## 4.5 The single layer potential operator

Let us recall the definition of the single layer potential operator  $\mathcal{S}_\Omega$ , representing the potential generated in  $\mathbb{R}^d$  by a density  $\varphi : \partial\Omega \rightarrow \mathbb{R}$  of charges on  $\partial\Omega$ .

**Definition 4.3.** *The single layer potential associated to a smooth density function  $\varphi \in C^\infty(\partial\Omega)$  is the function defined by:*

$$\mathcal{S}_\Omega \varphi(x) = \int_{\partial\Omega} F(x, y) \varphi(y) ds(y), \quad x \in \mathbb{R}^d \setminus \partial\Omega.$$

s The next proposition gathers a few properties of this mapping.

**Proposition 4.3.** *The following facts hold true:*

(i) *The mapping  $S_{\partial\Omega}$  has an extension as a bounded mapping from  $H^{-1/2}(\partial\Omega)$  into  $H^{1/2}(\partial\Omega)$ .*

(ii) *Let  $\Gamma$  be a Lipschitz open subset of  $\partial\Omega$ ; then  $S_{\partial\Omega}$  induces a bounded operator  $S_\Gamma : \tilde{H}^{-1/2}(\Gamma) \rightarrow H^{1/2}(\Gamma)$  via the formula:*

$$\forall \varphi \in \tilde{H}^{-1/2}(\Gamma), \quad S_\Gamma \varphi = (S_{\partial\Omega} \tilde{\varphi})|_\Gamma,$$

*where  $\tilde{\varphi} \in H^{-1/2}(\partial\Omega)$  is the extension by 0 to  $\partial\Omega$  of an element  $\varphi \in \tilde{H}^{-1/2}(\Gamma)$ .*

(iii) *If  $d \geq 3$ , the mapping  $S_\Gamma : \tilde{H}^{-1/2}(\Gamma) \rightarrow H^{1/2}(\Gamma)$  is invertible.*

The first point is found in [237] and the second follows almost immediately from the definitions of the functional spaces  $\tilde{H}^{-1/2}(\Gamma)$  and  $H^{1/2}(\Gamma)$ . The last point is more subtle, and it is proved in [57]. Note that when  $\Gamma$  is the unit disk  $\mathbb{D}_1$ , this result also holds when  $d = 2$ . This fact is needed in the rigorous proof of the next [Theorem 4.4](#) conducted in [57], but not in the formal method provided in the sequel.

## 4.6 The case of the conductivity equation

In this section, we address the case of the conductivity equation, which is arguably the simplest case to analyze and serves as a baseline for analyzing more complex situations. Here,  $n = 1$ ,  $V = H^1(\Omega)$ . We suppose that the boundary  $\partial\Omega$  is disjointly decomposed into  $\partial\Omega = \overline{\Gamma_D} \cup \overline{\Gamma}$ . Additionally, we assume that the domain  $\Omega$  is occupied by a medium with smooth isotropic conductivity  $\gamma \in C^\infty(\overline{\Omega})$ , satisfying the following bounds:

$$\forall x \in \Omega, \quad \alpha \leq \gamma(x) \leq \beta,$$

for some fixed constants  $0 < \alpha \leq \beta$ . In the background scenario, the voltage potential  $u_0 \in H^1(\Omega)$ , in response to a smooth external source  $f \in C^\infty(\overline{\Omega})$ , is the unique solution to the following boundary value problem:

$$\begin{cases} -\nabla \cdot (\gamma \nabla u_0) = f & \text{in } \Omega, \\ u_0 = 0 & \text{on } \Gamma_D, \\ \gamma \frac{\partial u_0}{\partial n_{\partial\Omega}} = 0 & \text{on } \Gamma. \end{cases} \quad (4.30)$$

In accordance to the classical elliptic regularity theory, the voltage potential  $u_0$  is smooth except maybe at the transition zone  $\Sigma = \overline{\Gamma_D} \cap \overline{\Gamma}$ , where the boundary condition changes; see [Section 4.2.2](#) and [172]. As suggested in the previous section, we introduce the error  $r_\varepsilon := u_\varepsilon - u_0$ . The primary tool for analyzing  $r_\varepsilon$  is the concept of Green's function. We begin by recalling the expression for the fundamental solution  $F(x)$  of the operator  $-\Delta$  in free space:

$$F(x) = \begin{cases} -\frac{1}{2\pi} \log|x| & \text{if } d = 2, \\ \frac{1}{(d-2)\alpha_d} |x|^{2-d} & \text{if } d \geq 3, \end{cases} \quad (4.31)$$

where  $\alpha_d$  is the area of the unit sphere  $\mathbb{S}^{d-1} \subset \mathbb{R}^d$ . For fixed  $x$ , the function  $F$  satisfies:

$$-\Delta F = \delta_0 \text{ in the sense of distributions on } \mathbb{R}^d,$$

where  $\delta_0$  is the Dirac delta distribution at 0. Now let  $F(x, y) := F(|x - y|)$ . From this, one can construct the Green's function  $y \mapsto N(x, y)$  for the background equation (4.30). This function satisfies, for all  $x \in \Omega$ :

$$\begin{cases} -\nabla_y \cdot (\gamma(y) \nabla_y N(x, y)) = \delta_{y=x} & \text{in } \Omega, \\ N(x, y) = 0 & \text{for } y \in \Gamma_D, \\ \gamma(y) \frac{\partial N}{\partial n_{\partial\Omega}(y)}(x, y) = 0 & \text{for } y \in \Gamma. \end{cases} \quad (4.32)$$

**Remark 4.8.** The function  $N(x, y)$  is symmetric in its arguments (see [149] for a proof) and is related to  $F(x, y)$  via the relation:

$$N(x, y) = \frac{1}{\gamma(x)} F(x, y) + R(x, y),$$

where  $R(x, y)$  is the corrector term, solution to:

$$\begin{cases} -\nabla_y \cdot (\gamma(y) \nabla_y R(x, y)) = \frac{1}{\gamma(y)} \nabla \gamma(y) \cdot \nabla_y F(x, y) & \text{in } \Omega, \\ R(x, y) = -\frac{1}{\gamma(y)} F(x, y) & \text{for } y \in \Gamma_D, \\ \gamma(y) \frac{\partial R}{\partial n_{\partial\Omega}(y)}(x, y) = \frac{\gamma(y)}{\gamma(x)} \frac{\partial N}{\partial n_{\partial\Omega}(y)}(x, y) & \text{for } y \in \Gamma. \end{cases}$$

The functional characterization of the corrector term  $R(x, y)$  depends on the singularity of  $F(x, y)$ . However, it is known that  $y \mapsto R(x, y)$  belongs at least to  $H^1(\Omega)$ . Additionally, for every open subset  $U$  compactly contained in  $\mathbb{R}^d \setminus (\Sigma \cup \{x\})$ , it is of class  $C^\infty$  on  $\bar{\Omega} \cap U$ . For more details on the characterization of the corrector term and the Green's function, we refer to the standard texts [62, 164].

The key property of  $N(x, y)$  that we will utilize is that, for any function  $\varphi \in C^1(\bar{\Omega})$  such that  $\varphi = 0$  on  $\Gamma_D$ , the following holds:

$$\varphi(x) = \int_{\Omega} \gamma(y) \nabla_y N(x, y) \cdot \nabla \varphi(y) \, dy, \quad x \in \Omega. \quad (4.33)$$

In particular, one may integrate by parts to express the solution to (4.30) in terms of  $N(x, y)$  as:

$$u_0(x) = \int_{\Omega} f(y) N(x, y) \, dy. \quad (4.34)$$

We begin the analysis by introducing the Green's function  $L(x, y)$  for the version of (4.30) posed on the lower half-space  $H$  with homogeneous Neumann boundary conditions imposed on  $\partial H$ . For all  $x \in H$ ,  $y \mapsto L(x, y)$  satisfies:

$$\begin{cases} -\nabla_y \cdot (\gamma(y) \nabla_y L(x, y)) = \delta_{y=x} & \text{in } H, \\ \gamma(y) \frac{\partial L}{\partial n_{\partial\Omega}(y)}(x, y) = 0 & \text{for } y \in \partial H. \end{cases} \quad (4.35)$$

The existence of this function can be established via the so called method of images (see [194]), which yields the following construction:

$$L(x, y) = \frac{1}{\gamma} (F(x - y) + F(x + y)). \quad (4.36)$$

It is straightforward to see that the definition above satisfies (4.35). Similarly to  $N(x, y)$ , for  $\varphi \in C^1(\bar{\Omega})$ , the following formula holds:

$$\varphi(x) = \int_{\Omega} \gamma(y) \nabla_y L(x, y) \cdot \nabla \varphi(y) \, dy, \quad x \in \Omega. \quad (4.37)$$

#### 4.6.1 Replacement of homogeneous Neumann boundary conditions by homogeneous Dirichlet conditions

In the general language of Section 4.3.2, we assume that  $G = \Gamma_D$ , meaning the homogeneous Neumann boundary condition on the small disk  $\omega_\varepsilon \subset \Gamma$  is replaced by a homogeneous Dirichlet condition. In this case, the perturbed potential  $u_\varepsilon \in H_{\Gamma_D}^1(\Omega)$  is then the unique solution to:

$$\begin{cases} -\nabla \cdot (\gamma \nabla u_\varepsilon) = f & \text{in } \Omega, \\ u_\varepsilon = 0 & \text{on } \Gamma_D \cup \omega_\varepsilon, \\ \gamma \frac{\partial u_\varepsilon}{\partial n_{\partial\Omega}} = 0 & \text{on } \Gamma \setminus \bar{\omega}_\varepsilon. \end{cases} \quad (4.38)$$

Furthermore, we have that  $r_\varepsilon = u_\varepsilon - u_0$  satisfies the boundary value problem:

$$\begin{cases} -\nabla \cdot (\gamma \nabla r_\varepsilon) = 0 & \text{in } \Omega, \\ r_\varepsilon = 0 & \text{on } \Gamma_D, \\ r_\varepsilon = -u_0 & \text{on } \omega_\varepsilon, \\ \gamma \frac{\partial r_\varepsilon}{\partial n_{\partial\Omega}} = 0 & \text{on } \Gamma \setminus \overline{\omega_\varepsilon}. \end{cases} \quad (4.39)$$

The main result in this setting is the following asymptotic expansion, which was rigorously proved in [56] without the simplifying assumptions (see Remark 4.7). As mentioned in Remark 4.7, using a flat boundary does not affect the final result. For convenience, we will provide a formal sketch of the proof here.

**Theorem 4.4.** *For any point  $x \in \overline{\Omega} \setminus (\Sigma \cup \{0\})$ , the following asymptotic expansion holds:*

$$u_\varepsilon(x) = \begin{cases} u_0(x) - \frac{\pi}{|\log \varepsilon|} \gamma(0) u_0(0) N(x, 0) + o\left(\frac{1}{|\log \varepsilon|}\right) & \text{if } d = 2, \\ u_0(x) - 4\varepsilon \gamma(0) u_0(0) N(x, 0) + o(\varepsilon) & \text{if } d = 3. \end{cases}$$

*Proof.* We assume that  $\gamma$  is constant on the set  $\omega_\varepsilon$ . We now proceed in four steps to derive the asymptotic behavior of  $r_\varepsilon$ .

**Step 1.** We construct a representation formula for the values of  $r_\varepsilon$  “far” from 0 in terms of its values inside the region  $\omega_\varepsilon$ . This task starts from the integral representation of  $r_\varepsilon$  with the help of the Green’s function  $N(x, y)$  defined in (4.32). For any  $x \in \Omega$ , it holds:

$$r_\varepsilon(x) = - \int_{\Omega} \nabla_y \cdot (\gamma(y) \nabla_y N(x, y)) r_\varepsilon(y) \, dy.$$

By integrating by parts twice in the above expression, we successively obtain:

$$\begin{aligned} r_\varepsilon(x) &= - \int_{\partial\Omega} \gamma(y) \frac{\partial N}{\partial n_{\partial\Omega}(y)}(x, y) r_\varepsilon(y) \, ds(y) + \int_{\Omega} \gamma(y) \nabla_y N(x, y) \cdot \nabla r_\varepsilon(y) \, dy \\ &= - \int_{\partial\Omega} \gamma(y) \frac{\partial N}{\partial n_{\partial\Omega}(y)}(x, y) r_\varepsilon(y) \, ds(y) + \int_{\partial\Omega} \gamma(y) \frac{\partial r_\varepsilon}{\partial n_{\partial\Omega}}(y) N(x, y) \, ds(y). \end{aligned}$$

Now, considering the boundary conditions satisfied by  $r_\varepsilon$  and  $y \mapsto N(x, y)$ , the first integral on the right-hand side vanishes, as does the integrand of the second integral on  $\Gamma_D$  and  $\Gamma \setminus \overline{\omega_\varepsilon}$ . Thus, we obtain:

$$r_\varepsilon(x) = \int_{\omega_\varepsilon} \gamma(y) \frac{\partial r_\varepsilon}{\partial n_{\partial\Omega}}(y) N(x, y) \, ds(y).$$

Applying a change of variables in the above integral, we obtain:

$$r_\varepsilon(x) = \int_{\mathbb{D}_1} \varphi_\varepsilon(z) N(x, \varepsilon z) \, ds(z), \quad (4.40)$$

where we have introduced the function  $\varphi_\varepsilon(z) := \varepsilon^{d-1} \left( \gamma \frac{\partial r_\varepsilon}{\partial n_{\partial\Omega}} \right) (\varepsilon z) \in \tilde{H}^{-1/2}(\mathbb{D}_1)$ . This is the desired representation formula.

**Step 2.** We characterize the function  $\varphi_\varepsilon$  by an integral equation. To achieve this, we utilize the Green’s function  $L(x, y)$  of the lower half-space, which captures the behavior of  $N(x, y)$  near 0, but is completely. Again, integrating by parts, we obtain that for any point  $x \in \Omega$  that is far from 0:

$$r_\varepsilon(x) = - \int_{\Omega} \nabla_y \cdot (\gamma(y) \nabla_y L(x, y)) r_\varepsilon(y) \, dy \quad (4.41)$$

$$= - \int_{\partial\Omega} \gamma(y) \frac{\partial L}{\partial n_{\partial\Omega}(y)}(x, y) r_\varepsilon(y) \, ds(y) + \int_{\Omega} \gamma(y) \nabla_y L(x, y) \cdot \nabla r_\varepsilon(y) \, dy \quad (4.42)$$

$$= - \int_{\partial\Omega} \gamma(y) \frac{\partial L}{\partial n_{\partial\Omega}(y)}(x, y) r_\varepsilon(y) \, ds(y) + \int_{\partial\Omega} \gamma(y) \frac{\partial r_\varepsilon}{\partial n_{\partial\Omega}}(y) L(x, y) \, ds(y). \quad (4.43)$$

Now using the fact that  $\partial\Omega$  coincides with  $\partial H$  in a neighborhood  $\mathcal{O}$  of 0, and the boundary conditions satisfied by  $r_\varepsilon$ , we obtain:

$$r_\varepsilon(x) = \int_{\omega_\varepsilon} \gamma(y) \frac{\partial r_\varepsilon}{\partial n_{\partial\Omega}}(y) L(x, y) \, ds(y) + Kr_\varepsilon(x), \text{ where}$$

$$Kr_\varepsilon(x) := - \int_{\Gamma \setminus \overline{\mathcal{O}}} \gamma(y) \frac{\partial L}{\partial n_{\partial\Omega}}(x, y) r_\varepsilon(y) \, ds(y) + \int_{\Gamma_D} \gamma(y) \frac{\partial r_\varepsilon}{\partial n_{\partial\Omega}}(y) L(x, y) \, ds(y).$$

Changing variables in the integral above, and letting  $x$  tend to  $\omega_\varepsilon$ , with the continuity of the single layer potential (see [Proposition 4.3](#)), we obtain

$$\forall x \in \mathbb{D}_1, \quad -u_0(\varepsilon x) = \int_{\omega_\varepsilon} \varepsilon^{d-1} \gamma \frac{\partial r_\varepsilon}{\partial n_{\partial\Omega}}(\varepsilon z) L(\varepsilon x, \varepsilon z) \, ds(z) + Kr_\varepsilon(\varepsilon x).$$

Now, since  $\varepsilon x$  lies “far” from the support of the integrals featured in  $Kr_\varepsilon$  and invoking the convergence [\(4.21\)](#) of the error  $r_\varepsilon$ , the last term in the above right-hand side tends to 0 as  $\varepsilon \rightarrow 0$ . Taking advantage of the explicit form [\(4.36\)](#) of the Green’s function  $L(x, y)$ , this eventually leads to the following integral equation for the function  $\varphi_\varepsilon \in \tilde{H}^{-1/2}(\mathbb{D}_1)$ :

$$\frac{2}{\gamma} S_1 \varphi_\varepsilon = -u_0(0) + o(1), \tag{4.44}$$

where we have used the shortcut  $S_1 \varphi := S_{\mathbb{D}_1} \varphi$ , see [Proposition 4.3](#).

**Step 3.** We analyze the integral equation [\(4.44\)](#) for  $\varphi_\varepsilon$ . At first, the combination of [\(4.44\)](#) with [Proposition 4.3](#) reveals that  $\varphi_\varepsilon$  is uniformly bounded with respect to  $\varepsilon > 0$ , i.e. there exists a constant  $C > 0$  independent of  $\varepsilon$  such that:

$$\|\varphi_\varepsilon\|_{\tilde{H}^{-1/2}(\mathbb{D}_1)} \leq C. \tag{4.45}$$

We then leverage the explicit expression [\(4.31\)](#) for  $F(x, y)$ , which depends on the space dimension.

- When  $d = 2$ , we obtain the following equation:

$$- \int_{\mathbb{D}_1} \log|\varepsilon x - \varepsilon z| \varphi_\varepsilon(z) \, ds(z) = -\pi \gamma u_0(0) + o(1),$$

which rewrites:

$$|\log \varepsilon| \left( \int_{\mathbb{D}_1} \varphi_\varepsilon(z) \, ds(z) \right) - S_1 \varphi_\varepsilon(z) = -\pi \gamma u_0(0) + o(1).$$

This relation yields immediately:

$$\int_{\mathbb{D}_1} \varphi_\varepsilon(z) \, ds(z) = -\frac{1}{|\log \varepsilon|} \pi \gamma u_0(0) + o\left(\frac{1}{|\log \varepsilon|}\right).$$

- For the case  $d = 3$  the integral equation [\(4.44\)](#) reads:

$$S_1 \varphi_\varepsilon(z) = -\frac{\varepsilon}{2} \gamma u_0(0), \text{ where } S_1 \varphi = \frac{1}{4\pi} \int_{\mathbb{D}_1} \frac{1}{|x-z|} \varphi(z) \, ds(z).$$

Using the explicit knowledge of the solution to this equation (see [\[95\]](#)):

$$\forall x \in \mathbb{D}_1, \quad \varphi(x) = \frac{4}{\pi \sqrt{1-|x|^2}}, \tag{4.46}$$

we obtain, in particular:

$$\varphi_\varepsilon(z) = -\frac{\varepsilon}{2} \gamma u_0(0) S_1^{-1} 1 + o(\varepsilon), \text{ and so } \int_{\mathbb{D}_1} \varphi_\varepsilon(z) \, ds(z) = -4\varepsilon \gamma u_0(0) + o(\varepsilon).$$

**Step 4.** We pass to the limit in the representation formula [\(4.40\)](#). Performing a Taylor expansion for the function  $y \mapsto N(x, y)$  in a neighborhood of 0, and considering that  $x \notin \Sigma \cup \{0\}$ , we obtain:

$$\left| \int_{\mathbb{D}_1} \varphi_\varepsilon(z) (N(x, \varepsilon z) - N(x, 0)) \, ds(z) \right| \leq \|\varphi_\varepsilon\|_{\tilde{H}^{-1/2}(\mathbb{D}_1)} \|N(x, \varepsilon \cdot) - N(x, 0)\|_{H^{1/2}(\mathbb{D}_1)}$$

$$\leq \begin{cases} o(|\log \varepsilon|), & d = 2, \\ o(\varepsilon), & d = 3. \end{cases}$$

This leads to

$$r_\varepsilon(x) = \left( \int_{\mathbb{D}_1} \varphi_\varepsilon(z) \, ds(z) \right) N(x, 0) + \begin{cases} o(|\log \varepsilon|), & d = 2, \\ o(\varepsilon), & d = 3. \end{cases}$$

yielding the desired formulas. ■

Let us now cast this result in the context of shape and topology optimization. The optimized region  $G$  is the homogeneous Neumann boundary  $\Gamma$  and we consider an objective function of the form:

$$J(G) = \int_{\Omega} j(u_G) \, dx,$$

where  $u_G$  is the solution to (4.30) with  $G = \Gamma$ . Here,  $j : \mathbb{R} \rightarrow \mathbb{R}$  is smooth as a function, satisfying the growth conditions (4.23). Then, the following corollary yields the topological sensitivity.

**Corollary 4.1.** *The perturbed criterion  $J(G_{0,\varepsilon})$ , where  $G = \Gamma_D$ , defined by:*

$$J(G_{0,\varepsilon}) := \int_{\Omega} j(u_\varepsilon) \, dx,$$

*accounting for the replacement of the homogeneous Neumann boundary conditions on  $\omega_{0,\varepsilon} \subset \Gamma$  by homogeneous Dirichlet boundary conditions, has the following asymptotic expansion:*

$$J(G_{0,\varepsilon}) = \begin{cases} J(G) + \frac{\pi}{|\log \varepsilon|} \gamma(0) u_0(0) p_0(0) + o\left(\frac{1}{|\log \varepsilon|}\right) & \text{if } d = 2, \\ J(G) + 4\varepsilon \gamma(0) u_0(0) p_0(0) + o(\varepsilon) & \text{if } d = 3, \end{cases}$$

*where  $p_0$  is the unique solution  $H^1(\Omega)$  to the boundary value problem:*

$$\begin{cases} -\nabla \cdot (\gamma \nabla p_0) = -j'(u_0) & \text{in } \Omega, \\ p_0 = 0 & \text{on } \Gamma_D, \\ \gamma \frac{\partial p_0}{\partial n_{\partial\Omega}} = 0 & \text{on } \Gamma. \end{cases} \quad (4.47)$$

*Proof.* We show the proof for the case  $d = 3$ , the case  $d = 2$  being similar. At first, using the asymptotic expansion Theorem 4.4, we have:

$$J(G_{0,\varepsilon}) = J(G) + \int_{\Omega} j'(u_0(x)) (-4\varepsilon \gamma(0) u_0(0) N(x, 0)) \, dx + o(\varepsilon).$$

This follows from an application of the Lebesgue dominated convergence theorem whose rigorous justification is detailed in [100]. Now using the representation formula (4.34) for the adjoint state  $p_0$  in Theorem 4.4, after integrating by parts twice, we see that:

$$\begin{aligned} J(G_{0,\varepsilon}) &= J(G) - 4\varepsilon \gamma(0) u_0(0) \int_{\Omega} j'(u_0(x)) N(x, 0) \, dx + o(\varepsilon) \\ &= J(G) + 4\varepsilon \gamma(0) u_0(0) p_0(0) + o(\varepsilon), \end{aligned}$$

which is the desired result. ■

**Remark 4.9 (The case of an inhomogeneous Dirichlet boundary).** *It is possible to replace the homogeneous Neumann boundary condition in (4.30) with an inhomogeneous Dirichlet boundary condition on  $\omega_\varepsilon$ , where  $u_\varepsilon = u_{\text{in}}$  for some smooth function  $u_{\text{in}} \in C^\infty(\mathbb{R}^d)$ . The above calculations can be straightforwardly adapted to this case, and the asymptotic expansion of  $r_\varepsilon$  becomes:*

$$u_\varepsilon(x) = \begin{cases} u_0(x) + \frac{\pi}{|\log \varepsilon|} \gamma(0) (u_{\text{in}}(0) - u_0(0)) N(x, 0) + o\left(\frac{1}{|\log \varepsilon|}\right) & \text{if } d = 2, \\ u_0(x) + 4\varepsilon \gamma(0) (u_{\text{in}}(0) - u_0(0)) N(x, 0) + o(\varepsilon) & \text{if } d = 3. \end{cases}$$

*In this case, the perturbed criterion  $J(G_{0,\varepsilon})$ , accounting for the replacement of the homogeneous Neumann boundary conditions on  $\omega_\varepsilon \subset \Gamma$  by inhomogeneous Dirichlet boundary conditions, has the following asymptotic expansion:*

$$J(G_{0,\varepsilon}) = \begin{cases} J(G) - \frac{\pi}{|\log \varepsilon|} \gamma(0) (u_{\text{in}}(0) - u_0(0)) p_0(0) + o\left(\frac{1}{|\log \varepsilon|}\right) & \text{if } d = 2, \\ J(G) - 4\varepsilon \gamma(0) (u_{\text{in}}(0) - u_0(0)) p_0(0) + o(\varepsilon) & \text{if } d = 3. \end{cases}$$

#### 4.6.2 Replacement of homogeneous Neumann boundary conditions by inhomogeneous Neumann boundary conditions

We presently turn to the situation where the homogeneous Neumann boundary condition on  $\Gamma$  is replaced by an inhomogeneous Neumann condition on a small region  $\omega_\varepsilon \subset \Gamma_N$ . The boundary  $\partial\Omega$  is decomposed into three disjoint pieces, as

$$\partial\Omega = \overline{\Gamma_D} \cup \overline{\Gamma} \cup \overline{\Gamma_N},$$

where:

- The region  $\Gamma_D$  is the support of homogeneous Dirichlet boundary conditions,
- The region  $\Gamma$  is the support of a homogeneous Neumann boundary condition.
- The region  $\Gamma_N$  is the support of an inhomogeneous Neumann boundary condition.

The region  $\omega_\varepsilon$  still denotes the flat disk  $\varepsilon\mathbb{D}_1$ . The voltage potential  $u_\varepsilon$  in this perturbed situation is the solution to the boundary value problem:

$$\begin{cases} -\nabla \cdot (\gamma \nabla u_\varepsilon) = f & \text{in } \Omega, \\ u_\varepsilon = 0 & \text{on } \Gamma_D, \\ \gamma \frac{\partial u_\varepsilon}{\partial n_{\partial\Omega}} = 0 & \text{on } \Gamma \setminus \overline{\omega_\varepsilon}, \\ \gamma \frac{\partial u_\varepsilon}{\partial n_{\partial\Omega}} = g & \text{on } \Gamma_N \cup \omega_\varepsilon. \end{cases} \quad (4.48)$$

**Theorem 4.5.** *The following asymptotic expansions hold:*

$$u_\varepsilon(x) = u_0(x) + 2\varepsilon g(0) N(x, 0) + o(\varepsilon) \text{ if } d = 2,$$

and

$$u_\varepsilon(x) = u_0(x) + \varepsilon^2 \pi g(0) N(x, 0) + o(\varepsilon^2) \text{ if } d = 3.$$

*Sketch of proof.* The derivation of these formulas essentially follows the trail of the proof of [Theorem 4.4](#), in a much simpler version. Again, let  $r_\varepsilon := u_\varepsilon - u_0$  be the error between the perturbed and background potentials. This function satisfies the following boundary value problem:

$$\begin{cases} -\nabla \cdot (\gamma \nabla r_\varepsilon) = 0 & \text{in } \Omega, \\ r_\varepsilon = 0 & \text{on } \Gamma_D, \\ \gamma \frac{\partial r_\varepsilon}{\partial n_{\partial\Omega}} = 0 & \text{on } (\Gamma_N \cup \Gamma) \setminus \overline{\omega_\varepsilon}, \\ \gamma \frac{\partial r_\varepsilon}{\partial n_{\partial\Omega}} = g & \text{on } \omega_\varepsilon. \end{cases} \quad (4.49)$$

From the definition of the Green's function  $N(x, y)$  in (4.32), it holds:

$$r_\varepsilon(x) = - \int_{\Omega} \nabla_y \cdot (\gamma(y) \nabla_y N(x, y)) r_\varepsilon(y) \, dy,$$

and so, after integration by parts:

$$r_\varepsilon(x) = - \int_{\partial\Omega} \gamma(y) \frac{\partial N}{\partial n_y}(x, y) r_\varepsilon(y) \, ds(y) + \int_{\partial\Omega} \gamma(y) \frac{\partial r_\varepsilon}{\partial n_{\partial\Omega}}(y) N(x, y) \, ds(y).$$

Now using the boundary conditions satisfied by the functions  $r_\varepsilon$  and  $y \mapsto N(x, y)$ , it follows:

$$\begin{aligned} r_\varepsilon(x) &= \int_{\omega_\varepsilon} g(y) N(x, y) \, ds(y) \\ &= \varepsilon^{d-1} \int_{\mathbb{D}_1} g(\varepsilon z) N(x, \varepsilon z) \, ds(z) \end{aligned}$$

and so

$$r_\varepsilon(x) = \varepsilon^{d-1} g(0) N(x, 0) \left( \int_{\mathbb{D}_1} ds(z) \right) + o(\varepsilon^{d-1}),$$

which yields the desired result. ■

**Corollary 4.2.** *The perturbed criterion  $J(G_{0,\varepsilon})$ , defined by,*

$$J(G_{0,\varepsilon}) := \int_{\Omega} j(u_\varepsilon) \, dx,$$

where  $u_\varepsilon$  is the solution to the boundary value problem (4.48), has the following expansion:

$$J(G_{0,\varepsilon}) = J(G) - 2\varepsilon g(0) p_0(0) + o(\varepsilon) \text{ if } d = 2,$$

and

$$J(G_{0,\varepsilon}) = J(G) - \pi \varepsilon^2 g(0) p_0(0) + o(\varepsilon^2) \text{ if } d = 3,$$

where  $p_0 \in H^1(\Omega)$  is again the solution to (4.47).

The proof is omitted, as it is completely similar to that of Corollary 4.1.

## 4.7 The case of the Helmholtz equation

In this section, we slip into the physical context where the field  $u_0$  is the solution to the Helmholtz equation, as in acoustics or wave scattering. We adapt the material of the previous sections to handle applications in this setting. As we have mentioned, the calculation of exact or approximate shape derivatives of functions depending on a region  $G \subset \partial\Omega$  can be conducted by the same procedures as those described in Section 4.2 in the context of the conductivity equation. Hence, we focus on the calculation of topological derivatives, referring to Section 4.12 for examples of the use of shape derivatives in the present situation. As a great part of the analysis is similar, we essentially focus on the differences. To emphasize the parallel between both situations, we retain the same notations as in the previous Section 4.6.

### 4.7.1 Presentation of the Helmholtz model

For the simplicity of the analysis, we focus on a model interior Helmholtz problem, excerpted from [43, 119, 323]. The arguments exposed in here can be adapted to the situation of infinite propagation media, see Section 4.12 for an example in this context. Let  $\Omega$  be a smooth bounded domain in  $\mathbb{R}^d$ , whose boundary  $\partial\Omega$  is decomposed into two disjoint pieces, as

$$\partial\Omega = \bar{\Gamma} \cup \bar{\Gamma}_R,$$

where:

- The region  $\Gamma$  is the support of homogeneous Neumann boundary conditions,
- The region  $\Gamma_R$  is the support of a homogeneous Robin boundary condition.

The background equation reads:

$$\begin{cases} -\nabla \cdot (\gamma \nabla u_0) - k^2 u_0 = f & \text{in } \Omega, \\ \gamma \frac{\partial u_0}{\partial n_{\partial\Omega}} = 0 & \text{on } \Gamma, \\ \gamma \frac{\partial u_0}{\partial n_{\partial\Omega}} + ik u_0 = 0 & \text{on } \Gamma_R. \end{cases} \quad (4.50)$$

Here, we have denoted the wave number by  $k > 0$ , and we have assumed a time dependency of the form  $e^{ikt}$  (i.e.  $u_0(x)$  is the complex-valued amplitude of the time-dependent wave  $u_0(x)e^{ikt}$ ). The coefficient  $\gamma$  accounts for the physical properties of the medium, and it satisfies:

$$0 < \gamma < +\infty.$$

From the physical viewpoint, the homogeneous Neumann boundary condition on  $\Gamma$  accounts for a hard wall, where perfect reflection of the incoming wave occurs, while the Robin condition encodes a partial absorption of the wave. We refer to Chap. 35 in [137] about the well-posedness of this boundary value problem, which is essentially due to the non zero imaginary part in the Robin condition.

Let us recall that the fundamental solution  $F(x)$  in the free space  $\mathbb{R}^d$  ( $d = 2, 3$ ) for the Helmholtz operator  $u \mapsto -\Delta u - k^2 u$  is given by the following formulas:

$$F(x) = \begin{cases} \frac{-1}{4i} H_0^{(1)}(k|x|) & \text{if } d = 2, \\ \frac{e^{ik|x|}}{4\pi|x|} & \text{if } d = 3, \end{cases} \quad (4.51)$$

where  $H_0^{(1)}$  is the Hankel function of the first kind and of order 0, which is the solution to the ordinary differential equation:

$$\frac{1}{r} \frac{d}{dr} \left( r \frac{dH}{dr}(r) \right) + k^2 H(r) = 0, \text{ for } r > 0,$$

see for instance [2]. Let  $F(x, y) := F(|x - y|)$ . The Green's function  $N(x, y)$  for the background problem (4.50) can be constructed from this datum by standard means, as the solution to:

$$\begin{cases} -\nabla_y \cdot (\gamma \nabla_y N(x, y)) - k^2 N(x, y) = \delta_{y=x} & \text{in } \Omega, \\ \gamma \frac{\partial N}{\partial n_{\partial\Omega}(y)}(x, y) = 0 & \text{for } y \in \Gamma, \\ \gamma \frac{\partial N}{\partial n_{\partial\Omega}(y)}(x, y) + ik N(x, y) = 0 & \text{for } y \in \Gamma_R. \end{cases} \quad (4.52)$$

**Remark 4.10.** *In this context also, a Green's function  $L(x, y)$  for the operator in (4.50) on the lower half-space  $H$ , satisfying homogeneous Neumann boundary conditions on  $\partial H$ , can be constructed by the method of images:*

$$L(x, y) = \frac{1}{\gamma(x)} (F(x, y) + F(x, -y)). \quad (4.53)$$

## 4.7.2 Replacement of homogeneous Neumann boundary conditions by Robin boundary conditions

We consider a perturbed version of the problem (4.50) where the homogeneous Neumann boundary condition is replaced by an impedance (or Robin) boundary condition on a “small” surfacic disk  $\omega_\varepsilon \subset \Gamma$  around  $x$ , that is:

$$\begin{cases} -\nabla \cdot (\gamma \nabla u_\varepsilon) - k^2 u_\varepsilon = f & \text{in } \Omega, \\ \gamma \frac{\partial u_\varepsilon}{\partial n_{\partial\Omega}} = 0 & \text{on } \Gamma \setminus \overline{\omega_\varepsilon}, \\ \gamma \frac{\partial u_\varepsilon}{\partial n_{\partial\Omega}} + ik u_\varepsilon = 0 & \text{on } \Gamma_R \cup \omega_\varepsilon, \end{cases} \quad (4.54)$$

We rely again on the locally flat boundary assumption (see Remark 4.7) and set  $x_0 = 0$ ,  $\omega_\varepsilon = \omega_{x_0, \varepsilon}$ , which we assume  $\gamma$  is completely constant. Our main result concerning the asymptotic behavior of  $u_\varepsilon$  when  $\varepsilon \rightarrow 0$  is the following.

**Theorem 4.6.** *Let  $0 \in \Gamma$ . The following asymptotic expansion holds:*

$$u_\varepsilon(x) = \begin{cases} u_0(x) - 2\varepsilon i k u_0(0) N(x, 0) + o(\varepsilon) & \text{if } d = 2, \\ u_0(x) - \pi \varepsilon^2 i k u_0(0) N(x, 0) + o(\varepsilon^2) & \text{if } d = 3. \end{cases}$$

*Sketch of proof.* Let  $0 = 0$ . We rely on the formal argument employed in our treatment of [Theorem 4.4](#), and we only sketch the main frame for brevity. The error  $r_\varepsilon = u_\varepsilon - u_0$  is the unique solution in  $H^1(\Omega; \mathbb{C})$  to the following boundary value problem:

$$\begin{cases} -\nabla \cdot (\gamma \nabla r_\varepsilon) - k^2 r_\varepsilon = 0 & \text{in } \Omega, \\ \gamma \frac{\partial r_\varepsilon}{\partial n_{\partial\Omega}} = 0 & \text{on } \Gamma \setminus \overline{\omega_\varepsilon}, \\ \gamma \frac{\partial r_\varepsilon}{\partial n_{\partial\Omega}} + i k r_\varepsilon = 0 & \text{on } \Gamma_R, \\ \gamma \frac{\partial r_\varepsilon}{\partial n_{\partial\Omega}} + i k r_\varepsilon = -i k u_0 & \text{on } \omega_\varepsilon. \end{cases} \quad (4.55)$$

We proceed with four steps.

**Step 1.** We construct a representation formula for the values of  $r_\varepsilon$  “far” from 0 in terms of its values inside the region  $\omega_\varepsilon$ . To this end, we rely on the Green’s function  $N(x, y)$  of the background problem [\(4.50\)](#); it holds:

$$\begin{aligned} r_\varepsilon(x) &= - \int_{\Omega} (\nabla_y \cdot (\gamma \nabla_y N(x, y)) + k^2 N(x, y)) r_\varepsilon(y) \, dy \\ &= - \int_{\partial\Omega} \gamma \frac{\partial N}{\partial n_{\partial\Omega}(y)}(x, y) r_\varepsilon(y) \, ds(y) + \int_{\Omega} (\gamma \nabla_y N(x, y) \cdot \nabla r_\varepsilon(y) - k^2 N(x, y) r_\varepsilon(y)) \, dy \\ &= - \int_{\partial\Omega} \gamma \frac{\partial N}{\partial n_{\partial\Omega}(y)}(x, y) r_\varepsilon(y) \, ds(y) + \int_{\partial\Omega} \gamma \frac{\partial r_\varepsilon}{\partial n_{\partial\Omega}(y)}(y) N(x, y) \, ds(y), \end{aligned}$$

where the second and third lines follow from integration by parts. Recalling the boundary conditions satisfied by  $r_\varepsilon$  and  $N(x, \cdot)$  in [\(4.50\)](#) and [\(4.52\)](#), the above expression simplifies to:

$$r_\varepsilon(x) = \int_{\omega_\varepsilon} \gamma \frac{\partial r_\varepsilon}{\partial n_{\partial\Omega}(y)}(y) N(x, y) \, ds(y),$$

and so, after rescaling:

$$r_\varepsilon(x) = \int_{\mathbb{D}_1} \varphi_\varepsilon(z) N(x, \varepsilon z) \, ds(z), \quad \text{where } \varphi_\varepsilon(z) := \varepsilon^{d-1} \left( \gamma \frac{\partial r_\varepsilon}{\partial n_{\partial\Omega}} \right) (\varepsilon z) \in \tilde{H}^{-1/2}(\mathbb{D}_1). \quad (4.56)$$

**Step 2.** We characterize the function  $\varphi_\varepsilon$  by an integral equation. To achieve this, we essentially repeat the calculations from the first step, except that we use the explicit Green’s function for the half-space  $L(x, y)$  discussed in [Remark 4.10](#) in place of the more abstract Green’s function  $N(x, y)$  for the background equation [\(4.50\)](#). Repeating the calculation of the first step, we obtain:

$$r_\varepsilon(x) = - \int_{\partial\Omega} \gamma \frac{\partial L}{\partial n_{\partial\Omega}(y)}(x, y) r_\varepsilon(y) \, ds(y) + \int_{\partial\Omega} \gamma \frac{\partial r_\varepsilon}{\partial n_{\partial\Omega}(y)}(y) L(x, y) \, ds(y),$$

Now using the boundary conditions satisfied by both functions  $r_\varepsilon$  and  $L(x, \cdot)$ , it follows:

$$r_\varepsilon(x) = \int_{\omega_\varepsilon} \gamma \frac{\partial r_\varepsilon}{\partial n_{\partial\Omega}(y)} L(x, y) \, ds(y) + K r_\varepsilon(x),$$

where the term  $K r_\varepsilon(x)$  gathers integrals of  $r_\varepsilon$  and its derivatives whose supports are “far” from 0.

Now “letting  $x$  tend to  $\omega_\varepsilon$ ” in this formula, and inserting the resulting expression in the Robin boundary condition in [\(4.55\)](#), we obtain:

$$\forall x \in \omega_\varepsilon, \quad \gamma \frac{\partial r_\varepsilon}{\partial n_{\partial\Omega}}(x) + i k \int_{\omega_\varepsilon} \gamma \frac{\partial r_\varepsilon}{\partial n_{\partial\Omega}(y)} L(x, y) \, ds(y) + K r_\varepsilon(x) = -i k u_0(x).$$

Eventually, by rescaling the integral, we obtain the following integral equation for the function  $\varphi_\varepsilon$  in (4.56):

$$\forall x \in \mathbb{D}_1, \quad \frac{1}{\varepsilon^{d-1}} \varphi_\varepsilon(x) + ik \int_{\mathbb{D}_1} \varphi_\varepsilon(z) L(\varepsilon x, \varepsilon z) \, ds(z) = -iku_0(0) + o(1). \quad (4.57)$$

**Step 3.** We use this integral equation to glean information about the asymptotic behavior of  $\varphi_\varepsilon$ . To this end, we observe that, because of the homogeneity of  $L(\cdot, \cdot)$ , the integral operator at the left-hand side of (4.57) is of order  $|\log \varepsilon|$  if  $d = 2$ , and of order  $\varepsilon^{-(d-2)}$  if  $d \geq 3$ ; it is thus negligible with respect to the first term in the left-hand side of this equation. Hence, taking the mean value in (4.57), we immediately obtain the following relations:

$$\int_{\mathbb{D}_1} \varphi_\varepsilon(z) \, ds(z) = \begin{cases} -2ik\varepsilon u_0(0) + o(\varepsilon) & \text{if } d = 2, \\ -\pi\varepsilon^2 ik u_0(0) \pi + o(\varepsilon^2) & \text{if } d = 3, \end{cases}$$

which is the needed information for our purpose.

**Step 4.** We pass to the limit in the representation formula (4.56). The application of the Lebesgue dominated convergence theorem to the representation formula (4.56) yields:

$$r_\varepsilon(x) \approx \left( \int_{\mathbb{D}_1} \varphi_\varepsilon(z) \, ds(z) \right) N(x, 0),$$

whence the desired result. ■

We can cast this result in the context of shape and topology optimization, along the lines of Corollary 4.1. The optimized region  $G$  is the Robin boundary  $\Gamma_R$  and we consider an objective function of the form:

$$J(G) = \int_{\Omega} j(u_G) \, dx,$$

where  $u_G$  is the solution to (4.50) with  $G = \Gamma_R$ . Here,  $j : \mathbb{C} \rightarrow \mathbb{R}$  is smooth, and it satisfies the growth conditions (4.23); with a small abuse of notation, we let:

$$j'(u) := \frac{\partial j}{\partial u_1}(u) + i \frac{\partial j}{\partial u_2}(u) \in \mathbb{C},$$

where  $u_1$  represents the real part of  $j$  and  $u_2$  its imaginary part. The sensitivity of the function  $J(G)$  with respect to the addition of a small surfacic disk  $\omega_\varepsilon$  is then given by the next result.

**Corollary 4.3.** *The perturbed criterion  $J(G_{0,\varepsilon})$  has the following asymptotic expansion in terms of  $\varepsilon$ :*

$$J(G_{0,\varepsilon}) = J(G) + 2\varepsilon k \operatorname{Im} \left( \overline{u_0(0)} p_0(0) \right) + o(\varepsilon) \text{ if } d = 2,$$

and

$$J(G_{0,\varepsilon}) = J(G) + \pi\varepsilon^2 k \operatorname{Im} \left( \overline{u_0(0)} p_0(0) \right) + o(\varepsilon^2) \text{ if } d = 3,$$

where the adjoint state  $p_0 \in H^1(\Omega; \mathbb{C})$  is the unique solution to the following boundary value problem:

$$\begin{cases} -\nabla \cdot (\gamma \nabla p_0) - k^2 p_0 = -j'(u_0) & \text{in } \Omega, \\ \gamma \frac{\partial p_0}{\partial n_{\partial\Omega}} = 0 & \text{on } \Gamma, \\ \gamma \frac{\partial p_0}{\partial n_{\partial\Omega}} - ik p_0 = 0 & \text{on } \Gamma_R. \end{cases} \quad (4.58)$$

*Sketch of proof.* The proof is very similar to that of Corollary 4.1. Considering the case  $d = 3$  to set ideas, the Lebesgue dominated convergence theorem shows that:

$$J(G_{0,\varepsilon}) = J(G) - \pi\varepsilon^2 k \operatorname{Re} \left( \int_{\Omega} j'(u_0(x)) \overline{(iu_0(0)N(x,0))} \, dx \right) + o(\varepsilon^2),$$

where we have used the basic calculus rule:

$$\forall u, h \in \mathbb{C}, \quad j(u+h) = j(u) + \operatorname{Re}(j'(u)\bar{h}) + o(h), \text{ where } o(h) \xrightarrow{h \rightarrow 0} 0.$$

Now using that the complex conjugate  $\overline{N(x, y)}$  is the Green's function for the boundary value problem (4.58) (where the sign in the Robin boundary condition is changed with respect to (4.50)), the representation formula (4.34) for  $p_0$  reads:

$$p_0(0) = - \int_{\Omega} j'(u_0(x)) \overline{N(x, 0)} dx,$$

and so:

$$\begin{aligned} J(G_{0, \varepsilon}) &= J(G) - \pi \varepsilon^2 k \operatorname{Re} \left( \overline{i u_0(0)} \int_{\Omega} j'(u_0(x)) \overline{N(x, 0)} dx \right) + o(\varepsilon^2) \\ &= J(G) + \pi \varepsilon^2 k \operatorname{Im} \left( \overline{u_0(0)} p_0(0) \right) + o(\varepsilon^2), \end{aligned}$$

which is the desired formula. ■

## 4.8 The case of the linear elasticity system

In this section, we extend the previous material to address applications within the context of the linear elasticity system. We continue to use the same notations whenever applicable, and the process of calculating exact or approximate shape derivatives follows closely the approach outlined in Section 4.6, although the computations are more technically involved. Therefore, we do not dwell on these aspects but instead focus on calculating the sensitivity of a model function accounting for the insertion of a small Dirichlet region within the homogeneous Neumann zone. It is important to note that this discussion can be easily adapted to various model variations, such as those involving loads on a fixed subset of the boundary.

### 4.8.1 Presentation of the linear elasticity setting

In this section,  $\Omega$  stands for a mechanical structure, whose boundary  $\partial\Omega$  is decomposed into three disjoint pieces:

$$\partial\Omega = \overline{\Gamma_D} \cup \overline{\Gamma_N} \cup \overline{\Gamma},$$

where:

- The displacement of the shape  $\Omega$  is prevented on the region  $\Gamma_D$ ;
- The region  $\Gamma_N$  is subjected to surface loads  $g \in L^2(\partial\Omega)^d$ ;
- The remaining part  $\Gamma$  is traction free.

Assuming body forces  $f \in L^2(\mathbb{R}^d)^d$ , the displacement  $u$  of  $\Omega$  is the unique solution in the space  $H_{\Gamma_D}^1(\Omega)^d$  to the following linear elasticity system:

$$\begin{cases} -\nabla \cdot (Ae(u_0)) = f & \text{in } \Omega, \\ Ae(u_0)n_{\partial\Omega} = g & \text{on } \Gamma_N, \\ Ae(u_0)n_{\partial\Omega} = 0 & \text{on } \Gamma, \\ u_0 = 0 & \text{on } \Gamma_D. \end{cases} \quad (4.59)$$

Here,  $e(u) := \frac{1}{2}(\nabla u + \nabla u^T)$  is the strain tensor associated to a displacement field, and  $A$  is the Hooke's tensor, defined by:

$$\forall e \in \mathcal{S}_d(\mathbb{R}), \quad Ae = 2\mu e + \lambda \operatorname{tr}(e)I,$$

where  $\lambda$  and  $\mu$  are the Lamé coefficients of the elastic material. Note that these coefficients are often better expressed in terms of the more physical Young's modulus  $E$  and Poisson's ratio  $\nu$ :

$$\mu = \frac{E}{2(1 + \nu)}, \quad \text{and } \lambda = \begin{cases} \frac{E\nu}{(1+\nu)(1-2\nu)} & \text{if } d = 3, \\ \frac{E\nu}{1-\nu^2} & \text{if } d = 2 \text{ (plane stress)}. \end{cases}$$

The fundamental solution to the linear elasticity operator  $u \mapsto -\nabla \cdot (Ae(u))$  in the free space  $\mathbb{R}^d$  is provided by the so-called Kelvin matrix, defined by

$$F_{ij}(x) = \begin{cases} \frac{\alpha}{4\pi} \frac{\delta_{ij}}{|x|} + \frac{\beta}{4\pi} \frac{x_i x_j}{|x|^3}, & \text{if } d = 3, \\ -\frac{\alpha}{2\pi} \delta_{ij} \log|x| + \frac{\beta}{2\pi} \frac{x_i x_j}{|x|^2}, & \text{if } d = 2. \end{cases} \quad (4.60)$$

for  $1 \leq d \leq d$ , where the constants  $\alpha$  and  $\beta$  are given by:

$$\alpha = \frac{1}{2} \left( \frac{1}{\mu} + \frac{1}{2\mu + \lambda} \right), \text{ and } \beta = \frac{1}{2} \left( \frac{1}{\mu} - \frac{1}{2\mu + \lambda} \right).$$

The meaning is that, for  $j = 1, \dots, d$ , the  $j^{\text{th}}$  column  $F_j$  of  $F$  is the solution to the equation

$$-\nabla \cdot (Ae(F_j)) = \delta_0 e_j,$$

where  $e_j$  is the  $j^{\text{th}}$  vector of the canonical basis of  $\mathbb{R}^d$ . From the physical point of view, if  $a \in \mathbb{R}^d$  is a unit vector,  $\Gamma a$  is the displacement when a pointwise load  $a$  is imposed at 0.

**Remark 4.11.** *It is interesting to recall how this formula for the Green's function of the linear elasticity system can be derived (see [30] for details). We search for one solution, in the sense of distributions, to the equation*

$$-\mu \Delta u - (\mu + \lambda) \nabla (\nabla \cdot u) = \delta_0 e_j. \quad (4.61)$$

Taking the divergence of this equation, we are led to search for  $u_k$  such that

$$-(2\mu + \lambda) \Delta (\nabla \cdot u) = \frac{\partial}{\partial x_j} \delta_0.$$

One solution to this equation is

$$\nabla \cdot u = \frac{1}{2\mu + \lambda} \frac{\partial G}{\partial x_j},$$

where  $G(x)$  is the Green's function for the operator  $-\Delta$ . Hence, (4.61) rewrites:

$$-\mu \Delta u = \frac{\mu + \lambda}{2\mu + \lambda} \nabla \left( \frac{\partial G}{\partial x_j} \right) + \delta_0 e_j.$$

One solution to this equation can now be found under the form of a radial function, after a simple (albeit tedious) calculation.

The variational counterpart of (4.60) reads: for all sufficiently smooth function  $\varphi \in \mathcal{C}_c^\infty(\mathbb{R}^d)^d$ ,

$$\varphi_j(x) = \int_{\mathbb{R}^d} Ae(F_j) : e(\varphi) \, dx.$$

Using the fundamental solution (4.60), it is possible to construct the Green's function  $N_{ij}(x, y)$  for the background linear elasticity system (4.59): for  $j = 1, \dots, d$ , the  $j^{\text{th}}$  column  $N_j$  of  $N$  satisfies the equation

$$\begin{cases} -\nabla_y \cdot Ae_y(N_j(x, y)) = \delta_{y=x} e_j & \text{in } \Omega, \\ N_j(x, y) = 0 & \text{for } y \in \Gamma_D, \\ Ae_y(N_j(x, y)) = 0 & \text{for } y \in \Gamma \cup \Gamma_N, \end{cases} \quad (4.62)$$

that is, under variational form:

$$\text{For all } \varphi \in \mathcal{C}_c^\infty(\mathbb{R}^d)^d \text{ s.t. } \varphi = 0 \text{ on } \Gamma_D, \quad \varphi_j(x) = \int_{\Omega} Ae_y(N_j(x, y)) : e(\varphi)(y) \, dy. \quad (4.63)$$

**Remark 4.12.** *Contrary to the situations in Sections 4.6 and 4.7 involving the conductivity and Helmholtz equations, the construction of a Green's function  $L(x, y)$  for the elasticity system (4.59) in the lower half-space  $H$  cannot be easily achieved by the method of images. An expression, however, can be established. For two points  $x, y$  on  $\partial H$ , the (matrix-valued) Mindlin function  $L_{ij}(x, y)$ ,  $1 \leq i, j \leq d$ , is defined by the following.*

- In 2D, the components  $L_{ij}$  read:

$$\begin{aligned} L_{11}(x, y) &= -\frac{1-\bar{\nu}}{\pi\mu} \log|x-y| + \frac{3-4\bar{\nu}}{8\pi\mu(1-\bar{\nu})}, \\ L_{12}(x, y) &= -\frac{1}{2\pi\mu}(1-2\bar{\nu})\Theta, \quad \text{where } \Theta = \begin{cases} \frac{\pi}{2} & \text{if } x_1 > y_1, \\ -\frac{\pi}{2} & \text{otherwise,} \end{cases} \\ L_{22}(x, y) &= -\frac{1-\bar{\nu}}{\pi\mu} \log|x-y|, \end{aligned} \quad (4.64)$$

where  $\bar{\nu} = \frac{\nu}{1+\nu}$ .

- In 3D, the components  $L_{ij}$  read:

$$\begin{aligned} L_{ij}(x, y) &= \frac{1-\nu}{2\pi\mu|x-y|} \delta_{ij} + \frac{\nu}{2\pi\mu} \frac{(x_i-y_i)(x_j-y_j)}{|x-y|^3}, \quad i, j = 1, 2, \\ L_{3j}(x, y) &= -\frac{1-2\nu}{4\pi\mu} \frac{x_j-y_j}{|x-y|^2}, \quad j = 1, 2, \\ L_{33}(x, y) &= \frac{1-\nu}{2\pi\mu|x-y|}. \end{aligned} \quad (4.65)$$

We refer to [39] §2.8.2 for the 2D case and [238, 246] for the original derivation in 3D. Note that, in the latter, the formulas may look different, as the Green's function is taken for the superior half-space (which changes the sign of the components  $L_{13}$ ,  $L_{23}$ ,  $L_{31}$  and  $L_{32}$  of the tensor).

**Remark 4.13.** *In essence, the Mindlin function describes the displacement field generated in an infinite half-space when a concentrated force is applied at a point on or below the surface of the half-space. This function is named after Raymond D. Mindlin, who developed the theory in [238] to address problems in the field of elasticity, specifically in the context of half-space problems.*

## 4.8.2 Replacement of homogeneous Neumann boundary conditions by homogeneous Dirichlet conditions

We consider the perturbed version of the elasticity problem (4.59) where the homogeneous Neumann boundary conditions on  $\Gamma_N$  are replaced by homogeneous Dirichlet boundary conditions on a small subset  $\omega_\varepsilon \subset \Gamma_N$ , taking the form of a surface disk. The displacement  $u_\varepsilon$  of the considered structure in this perturbed situation is the unique solution to the following boundary value problem:

$$\begin{cases} -\nabla \cdot Ae(u_\varepsilon) = f & \text{in } \Omega, \\ u_\varepsilon = 0 & \text{on } \Gamma_D \cup \omega_\varepsilon, \\ Ae(u_\varepsilon)n_{\partial\Omega} = g & \text{on } \Gamma_N, \\ Ae(u_\varepsilon)n_{\partial\Omega} = 0 & \text{on } \Gamma \setminus \overline{\omega_\varepsilon}. \end{cases} \quad (4.66)$$

Our main result is the following.

**Theorem 4.7.** *Let  $0$  be a given point in  $\Gamma$ . Then the following asymptotic expansion holds:*

$$u_{\varepsilon,j}(x) = u_0(x) - \frac{1}{|\log \varepsilon|} \frac{\pi\mu}{1-\bar{\nu}} u_0(0) \cdot N_j(x,0) + o\left(\frac{1}{|\log \varepsilon|}\right) \text{ if } d = 2,$$

and

$$u_{\varepsilon,j}(x) = u_0(x) - \varepsilon M u_0(0) \cdot N_j(x,0) + o(\varepsilon) \text{ if } d = 3.$$

In this formula, the polarization tensor  $M$ , whose entries read:

$$M_{ij} = \int_{\mathbb{D}_1} T_L^{-1} e_j \cdot e_i \, ds, \quad i, j = 1, 2, 3; \quad (4.67)$$

this definition involves the integral operator

$$T_L : \tilde{H}^{-1/2}(\mathbb{D}_1)^d \rightarrow H^{1/2}(\mathbb{D}_1)^d, \quad T_L \varphi(x) = \int_{\mathbb{D}_1} L(x,z) \varphi(z) \, ds(z), \quad x \in \mathbb{D}_1, \quad (4.68)$$

whose kernel is the Mindlin function  $L(x,y)$  given in (4.64) and (4.65).

**Remark 4.14.**

The operator  $T_L$  has a neat physical interpretation: when  $\varphi : \mathbb{D}_1 \rightarrow \mathbb{R}^d$  represents a force applied on the unit disk  $\mathbb{D}_1$ , and  $T_L \varphi : \mathbb{D}_1 \rightarrow \mathbb{R}^d$  is the induced displacement field. Hence, for  $j = 1, 2, 3$ , the  $j^{\text{th}}$  column of the polarization tensor is the force that needs to be applied on  $\mathbb{D}_1$  to realize a unit displacement in the direction  $e_j$ . Interestingly, classical mechanical calculations in contact mechanics boil down to those of some of the entries of this tensor, see for instance the so-called “flat punch” or “indentation” problems in [45, 199, 214]. To the best of our knowledge, the complete calculation of the tensor  $M$  is not available in the literature.

**Remark 4.15.** The definition of the polarization tensor  $M$  in (4.67) implicitly assumes that the operator  $T_L$  in (4.68) is invertible. To the best of our knowledge, this fact is not known in the literature, and it is not a straightforward adaptation of the counterpart result in the setting of the conductivity equation (see Proposition 4.3), and we proceed under the assumption that this fact holds true.

*Sketch of proof.* Again, we rely on a formal calculation along the trail of Theorem 4.4, under the assumption of that  $0 = 0$  and that the boundary is locally flat near  $0$  (see Fig. 4.6). Let us introduce the error  $r_\varepsilon = u_\varepsilon - u_0 \in H^1(\Omega)^d$ , which is the unique solution to the following boundary value problem:

$$\begin{cases} -\nabla \cdot (Ae(r_\varepsilon)) = 0 & \text{in } \Omega, \\ r_\varepsilon = 0 & \text{on } \Gamma_D, \\ r_\varepsilon = -u_0 & \text{on } \omega_\varepsilon, \\ Ae(r_\varepsilon)n = 0 & \text{on } \Gamma_N \cup (\Gamma \setminus \bar{\omega}_\varepsilon). \end{cases} \quad (4.69)$$

**Step 1.** We construct a representation formula for the values  $r_\varepsilon$ .

To this end, we use the definition (4.62) of the Green’s function  $N(x,y)$  for the background elasticity problem (4.59), which yields, for each component  $j = 1, \dots, d$ :

$$\begin{aligned} r_{\varepsilon,j}(x) &= - \int_{\Omega} \nabla_y \cdot (Ae_y(N_j(x,y))) \cdot r_\varepsilon(y) \, dy \\ &= - \int_{\partial\Omega} Ae_y(N_j(x,y))n(y) \cdot r_\varepsilon(y) \, ds(y) + \int_{\Omega} Ae_y(N_j(x,y)) : e(r_\varepsilon)(y) \, dy. \end{aligned}$$

Using the boundary conditions in (4.62) and (4.69) for the functions  $N(x,\cdot)$  and  $r_\varepsilon$ , as well as another integration by parts, we obtain:

$$r_{\varepsilon,j}(x) = - \int_{\Omega} \nabla \cdot (Ae(r_\varepsilon))(y) \cdot N_j(x,y) \, dy + \int_{\partial\Omega} Ae(r_\varepsilon)(y)n(y) \cdot N_j(x,y) \, ds(y).$$

Using again the boundary conditions satisfied by  $r_\varepsilon$  in (4.69), we end up with

$$r_{\varepsilon,j}(x) = \int_{\omega_\varepsilon} Ae(r_\varepsilon)(y)n(y) \cdot N_j(x, y) \, ds(y).$$

By a change of variables in the above integral, we obtain the desired representation formula:

$$r_{\varepsilon,j}(x) = \int_{\mathbb{D}_1} \varphi_\varepsilon(z) \cdot N_j(x, \varepsilon z) \, ds(z), \quad (4.70)$$

where we have introduced the rescaled quantity:

$$\varphi_\varepsilon(z) = \varepsilon^{d-1} \left( Ae(r_\varepsilon)n \right) (\varepsilon z) \in \tilde{H}^{-1/2}(\mathbb{D}_1)^d.$$

**Step 2.** We construct an integral equation characterizing the function  $\varphi_\varepsilon(z)$ . To achieve this, we rely on the same calculations as above, except that we replace the difficult Green's function  $N(x, y)$  for the background equation (4.59) with the Green's function  $L(x, y)$  for the lower half space, with Neumann boundary conditions on  $\partial H$ , see (4.64) and (4.65). As in the proof of Theorem 4.4, this yields the integral equation:

$$\forall x \in \mathbb{D}_1, \quad \int_{\mathbb{D}_1} L(\varepsilon x, \varepsilon z) \varphi_\varepsilon(z) \, ds(z) = -u_0(\varepsilon x) + o(1). \quad (4.71)$$

*Step 3:* We use this integral equation to glean information about the function  $\varphi_\varepsilon(z)$ . In two space dimensions, using the expression (4.64) of  $L(x, y)$ , this integral equation rewrites:

$$\frac{1-\bar{\nu}}{\pi\mu} (\log \varepsilon) \int_{\mathbb{D}_1} \varphi_\varepsilon(y) \, ds(y) + K\varphi_\varepsilon = -u_0(\varepsilon x) + o(1),$$

where  $K : \tilde{H}^{-1/2}(\mathbb{D}_1)^d \rightarrow H^{1/2}(\mathbb{D}_1)^d$  is a bounded operator. We directly obtain from this equation that:

$$\int_{\mathbb{D}_1} \varphi_\varepsilon(y) \, ds(y) = \frac{1}{|\log \varepsilon|} \frac{\pi\mu}{1-\bar{\nu}} u_0(0) + o\left(\frac{1}{|\log \varepsilon|}\right), \quad (4.72)$$

which is the needed information for our purpose.

In the case of three space dimensions, using the homogeneity of the kernel  $L(x, y)$  in (4.64) and (4.65), (4.71) becomes:

$$T_L \varphi_\varepsilon = -\varepsilon u_0(0) + o(\varepsilon),$$

where the integral operator  $T_L$  is defined in (4.68). This yields immediately:

$$\varphi_\varepsilon = \sum_{j=1}^3 u_{0,j}(0) T_L^{-1} e_j,$$

and so:

$$\int_{\mathbb{D}_1} \varphi_\varepsilon \, ds = M u_0(0). \quad (4.73)$$

*Step 4:* We pass to the limit in the representation formula. As usual, applying the Lebesgue dominated convergence theorem to the representation formula (4.70), we obtain:

$$r_{\varepsilon,j}(x) \approx \left( \int_{\mathbb{D}_1} \varphi_\varepsilon(z) \, ds(z) \right) \cdot N_j(x, 0),$$

and the desired result follows from the combination of this identity with (4.72) and (4.73). ■

Let us consider again an objective function of the form:

$$J(G) = \int_{\Omega} j(u_G) \, dx,$$

for a smooth function  $j : \mathbb{R}^d \rightarrow \mathbb{R}$ , satisfying suitable growth conditions, see (4.23). Here,  $u_G$  denotes the solution to the boundary value problem (4.59) when the Dirichlet region  $\Gamma_D$  is  $G$ . The result of interest is the following.

**Corollary 4.4.** *The perturbed value  $J(G_{0,\varepsilon})$  of the functional  $J(G)$ , accounting for the replacement of the homogeneous Neumann boundary conditions on  $\omega_{0,\varepsilon} \subset \Gamma$  by homogeneous Dirichlet boundary conditions, has the following asymptotic expansion:*

$$J(G_{0,\varepsilon}) = J(G) + \frac{1}{|\log \varepsilon|} \frac{\pi\mu}{1-\bar{\nu}} u_0(0) \cdot p_0(0) + o\left(\frac{1}{|\log \varepsilon|}\right) \text{ if } d = 2,$$

and

$$J(G_{0,\varepsilon}) = J(G) + \varepsilon M u_0(0) \cdot p_0(0) + o(\varepsilon) \text{ if } d = 3.$$

Here, the polarization tensor  $M$  is defined by (4.67) and the adjoint state  $p_0$  is the unique solution in  $H^1(\Omega)^d$  to the boundary value problem:

$$\begin{cases} -\nabla \cdot A e(p_0) = -j'(u_0) & \text{in } \Omega, \\ p_0 = 0 & \text{on } \Gamma_D, \\ A e(p_0) n = 0 & \text{on } \Gamma_N \cup \Gamma. \end{cases} \quad (4.74)$$

**Remark 4.16.** *Like in the setting of Section 4.6, involving the conductivity equation, multiple variations of the present study could be considered. For instance, one may be interested in accounting for the effect of the replacement of homogeneous Neumann boundary conditions by inhomogeneous Dirichlet or Neumann conditions, etc. Numerical examples associated to such variations are presented in Sections 4.13 and 4.14.*

## 4.9 Numerical resolution of singular boundary integral equations

As we have seen in the previous sections, the identification of the expression of our topological derivatives at some point involves the resolution of a boundary integral equation, of the form:

$$\text{Search for } \varphi \in \tilde{H}^{-1/2}(\mathbb{D}_1) \quad \text{s.t.} \quad T_L \varphi(x) = f(x), \quad x \in \mathbb{D}_1, \quad (\mathcal{B})$$

where the integral operator  $T_L : \tilde{H}^{-1/2}(\mathbb{D}_1) \rightarrow H^{1/2}(\mathbb{D}_1)$  is defined by:

$$T_L \varphi(x) = \int_{\mathbb{D}_1} L(x, z) \varphi(z) dz, \quad (4.75)$$

$L$  is a homogeneous kernel of class  $-1$  (see (4.36), (4.53) and (4.65)), and  $f \in H^{1/2}(\mathbb{D}_1)$  is a given source. In fortunate cases, the solution to this equation can be computed analytically, such as the case of the conductivity (see Theorem 4.4). When this is not the case, one needs to solve it numerically, see Theorem 4.7. In this section, we aim to detail the numerical resolution of an integral equation of the type  $(\mathcal{B})$ . In the literature, this type of problem is usually known as a ‘‘screen’’ problem; the equation is posed over an open surface (with boundary) rather than the whole boundary of a domain, which is a closed surface. In general, the exact solutions to screen problems are expected to blowup at the boundary of the surface, resulting in a lower degree of global regularity (see §4.1.11 in [283]). In particular, the work [171] analyzes this exact problem, suggesting a Galerkin scheme along with numerical quadrature formulas to achieve the successful numerical resolution of the problem, when the open surface is meshed with rectangles.

The material that is presented in this section, is primarily inspired from the book [283], where the authors present a way to work around the singularities of the function  $L(x, y)$  at  $x = y$ . Despite the fact that we have the added complication of working on the open surface, we re-use their techniques at the cost of introducing numerical artifacts of the solution, which we show how to remedy numerically in Section 4.9.3.

For the interested reader, we recommend the works [197, 175] on solving boundary integral equations, which provide in-depth analysis for general cases, including those where the kernel is non-singular.

### 4.9.1 The Galerkin scheme

Following [174], our numerical solution hinges on the derivation of a Galerkin scheme for (G). Multiplying (B) with an arbitrary function  $\psi \in \tilde{H}^{-1/2}(\mathbb{D}_1)$  and integrating, we obtain:

$$\forall \psi \in \tilde{H}^{-1/2}(\mathbb{D}_1), \quad \int_{\mathbb{D}_1} T_L \varphi(x) \psi(x) \, ds(x) = \int_{\mathbb{D}_1} f(x) \psi(x) \, ds(x). \quad (\mathcal{G})$$

As it turns out, regardless of the singularity at  $x = y$  the integral:

$$\int_{\mathbb{D}_1} T_L \varphi(x) \psi(x) \, ds(x) = \int_{\mathbb{D}_1} \int_{\mathbb{D}_1} L(x, y) \varphi(x) \psi(x) \, ds(y) \, ds(x),$$

is improperly integrable and admits a principal value. The numerical resolution of (G) can be carried out by discretizing the spaces of functions  $\tilde{H}^{-1/2}(\mathbb{D}_1)$ . For some finite element space  $V_h \subset \tilde{H}^{-1/2}(\mathbb{D}_1)$ , supported on a triangulation  $\mathcal{T}$ , the operator  $T_L : V_h \rightarrow L^2(\mathbb{D}_1)$  is a continuous mapping, since it maps  $\tilde{H}^{-1/2}(\mathbb{D}_1)$  continuously into  $H^{1/2}(\mathbb{D}_1)$  [171]. This allows to decompose the integral in the left hand side of (G) as:

$$\int_{\mathbb{D}_1} T_L \varphi(x) \psi(x) \, ds(x) = \sum_{\tau \in \mathcal{T}} \int_{\tau} T_L \varphi(x) \psi(x) \, ds(x).$$

Now let  $\tau \in \mathcal{T}$ , and define, the localized version of  $T_L$ :

$$\forall x \in \mathbb{D}_1, \quad T_L^\tau \varphi(x) := \text{p.v.} \int_{\tau} L(x, y) \varphi(y) \, ds(y). \quad (4.76)$$

The following lemma guarantees that the localization of the operator is consistent; see §5.1.2 of [283].

**Lemma 4.1.** *Let  $\varphi \in L^\infty(\mathbb{D}_1)$  and  $\phi|_\tau \in C^1(\tau)$  for all  $\tau \in \mathcal{T}$ , then we have:*

1.  $\forall x \in \tau$ ,  $T_L^\tau \varphi(x)$  is finite,
2.  $\forall x \notin \bar{\tau}$ , the principal value integral in (4.76) is a classical Riemann integral.
3. For all  $x \in \mathbb{D}_1$ , it holds:

$$T_L \varphi(x) = \sum_{\tau \in \mathcal{T}} T_L^\tau \varphi(x) \, ds(x).$$

**Corollary 4.5.** *Under the hypotheses of the previous lemma; it holds:*

$$\psi \in L^2(\mathbb{D}_1), \quad \int_{\mathbb{D}_1} T_L \varphi(x) \psi(x) \, ds(x) = \sum_{\tau_1, \tau_2 \in \mathcal{T}} A_{\tau_1, \tau_2},$$

where the quantity  $A_{\tau_1, \tau_2}$  is defined by:

$$A_{\tau_1, \tau_2} = \int_{\tau_1} \psi(x) \left( \text{p.v.} \int_{\tau_2} L(x, y) \varphi(y) \, ds(y) \right) ds(x).$$

### 4.9.2 Decomposition of singular integrals through relative coordinates

Let  $\tau_1, \tau_2 \in \mathcal{T}$ . In this section we explain how to transform integrals of the form:

$$A_{\tau_1, \tau_2} = \int_{\tau_1} \psi(x) \left( \text{p.v.} \int_{\tau_2} L(x, y) \varphi(y) \, ds(y) \right) ds(x). \quad (4.77)$$

in order to achieve their numerical computation. We rely on the so called relative coordinates, which are commonly used to simplify the evaluation of singular integrals; see [283]. Singular integrals present challenges because the integrand becomes unbounded or behaves irregularly at certain points within the domain. By shifting the coordinate system to a reference system where the singularity occurs at another position, relative coordinates simplify the integrand's expression, making it easier to handle both mathematically and numerically, thus improving numerical stability and accuracy.

In our case every  $\tau \in \mathcal{T}$  is the diffeomorphic image under of the reference element  $\hat{\tau} \subset \mathbb{R}^2$  with an affine transformation  $T_\tau : \hat{\tau} \rightarrow \tau$ . We assume that on each  $\tau$  we have a set of basis functions  $b_{\tau,i}$ ,  $1 \leq i \leq n$ , so that the reference basis functions on the reference element  $\hat{\tau}$  are given by  $\hat{b}_{\tau,i} = b_{\tau,i} \circ T_\tau$  (see [136]). This allows to discretize  $\varphi$  and  $\psi$  on each  $\tau$  as:

$$\forall x \in \tau, \quad \varphi(x) = \sum_{i=1}^n w_{\tau,i} b_{\tau,i}(x), \quad \psi(x) = \sum_{i=1}^n w_{\tau,i} b_{\tau,i}(x).$$

Next, we pullback (4.77) onto the reference elements to further write:

$$\begin{aligned} A_{\tau_1, \tau_2} &= \sum_{\tau_1, \tau_2 \in \mathcal{T}} \sum_{i, j=0}^n w_{\tau_1, j} w_{\tau_2, i} \hat{A}_{\tau_1, \tau_2}^{ij} g_{\tau_1, \tau_2}(\hat{x}, \hat{y}), \\ \hat{A}_{\tau_1, \tau_2}^{ij} &:= \int_{\hat{\tau}_1} \text{p.v.} \int_{\hat{\tau}_2} k_{ij}(\hat{x}, \hat{y}) \, d\hat{y} \, d\hat{x}, \\ k_{ij}(\hat{x}, \hat{y}) &:= (L(T_{\tau_1}(\hat{x}), T_{\tau_2}(\hat{y})) \hat{b}_{\tau_2, j}(\hat{y})) \hat{b}_{\tau_1, i}(\hat{x}) \\ g_{\tau_1, \tau_2}(\hat{x}, \hat{y}) &:= \sqrt{\det(\nabla T_{\tau_1}^T(\hat{x}) \nabla T_{\tau_1}(\hat{x}))} \sqrt{\det(\nabla T_{\tau_2}^T(\hat{y}) \nabla T_{\tau_2}(\hat{y}))}. \end{aligned} \tag{4.78}$$

**Remark 4.17.** *From the implementation point of view, the quantity  $\hat{A}_{\tau_1, \tau_2}^{ij}$  represents the contribution of  $\tau_1$  and  $\tau_2$  to the  $(i, j)$ -th entry of the global stiffness matrix, while the quantity  $g_{\tau_1, \tau_2}(\hat{x}, \hat{y})$  represents the volume distortion at the points  $x, y$ .*

Now, we exploit the explicit nature of the reference element on which the integration is performed. We can distinguish between four cases when it comes to calculating (4.78):

1.  $\tau_1 = \tau_2$ ,
2.  $\tau_1$  and  $\tau_2$  share a common edge,
3.  $\tau_1$  and  $\tau_2$  share a common vertex,
4.  $\tau_1 \cap \tau_2 = \emptyset$ .

In general, deriving the regularized form of the integral  $A_{\tau_1, \tau_2}$  is quite tedious and complicated, as it involves a non-trivial decomposition of the integration domain. Since this procedure is already well documented in [283, 176], we will simply state the formulas for the case where the reference element is the unit triangle, as illustrated in Fig. 4.7, given by the parametrization:

$$\hat{\tau} = \{(x, y) \in \mathbb{R}^2 \mid 0 \leq x \leq y \leq 1\}.$$

Let us mention that, in what follows, the decompositions for  $\hat{A}_{\tau_1, \tau_2}^{ij}$  are numerically integrable via standard Gauss quadrature methods on the unit interval  $(0, 1)$ ; see, for example, [111, 283] for a general overview or [166] for the quadrature formulae calculation.

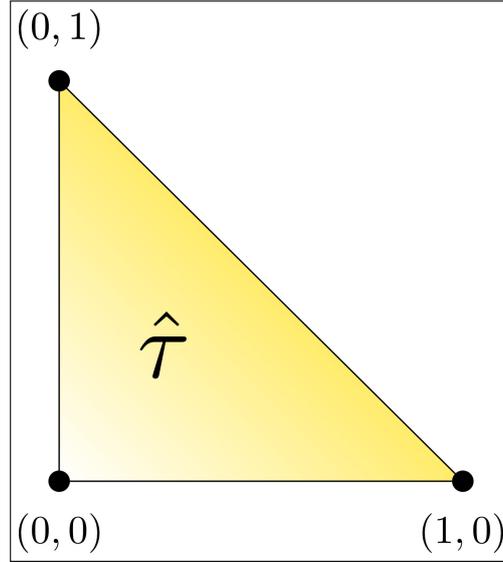


Figure 4.7: Illustration of the unit triangle considered.

**Case 1:**  $\tau_1 = \tau_2$

The integral  $\hat{A}_{\tau, \tau}$  is decomposed as follows:

$$\hat{A}_{\tau_1, \tau_2}^{ij} = \int_{(0,1)^4} \xi^3 \eta_1^2 \eta_2 \left( \sum_{l=1}^6 k_{ij}(\xi x_l, \xi y_l) \right) d\eta_1 d\eta_2 d\eta_3 d\xi,$$

where  $x_l$  and  $y_l$  are functions of  $\eta_1, \eta_2, \eta_3$ , defined by:

$$\begin{aligned} x_1 &= (1, 1 - \eta_1 + \eta_1 \eta_2), & y_1 &= (1 - \eta_1 \eta_2 \eta_3, 1 - \eta_1), \\ x_2 &= (1 - \eta_1 \eta_2 \eta_3, 1 - \eta_1), & y_2 &= (1, 1 - \eta_1 + \eta_1 \eta_2), \\ x_3 &= (1, \eta_1(1 - \eta_2 + \eta_2 \eta_3)), & y_3 &= (1 - \eta_1 \eta_2, 1 - \eta_1 \eta_2), \\ x_4 &= (1 - \eta_1 \eta_2, \eta_1(1 - \eta_2)), & y_4 &= (1, \eta_1(1 - \eta_2 + \eta_2 \eta_3)), \\ x_5 &= (1 - \eta_1 \eta_2 \eta_3, \eta_1(1 - \eta_2 \eta_3)), & y_5 &= (1, \eta_1(1 - \eta_2)), \\ x_6 &= (1, \eta_1(1 - \eta_2)), & y_6 &= (1 - \eta_1 \eta_2 \eta_3, \eta_1(1 - \eta_2 \eta_3)). \end{aligned}$$

**Case 2:**  $\tau_1$  and  $\tau_2$  share a common edge

The integral  $\hat{A}_{\tau_1, \tau_2}$  can be decomposed as follows:

$$\hat{A}_{\tau_1, \tau_2}^{ij} = \int_{(0,1)^4} \left( \xi^3 \eta_1^2 k_{ij}(\xi x_1, \xi y_1) + \xi^3 \eta_1^2 \eta_2 \left( \sum_{l=2}^5 k_{ij}(\xi x_l, \xi y_l) \right) \right) d\eta_1 d\eta_2 d\eta_3 d\xi,$$

where  $x_l$  and  $y_l$  are functions of  $\eta_1, \eta_2, \eta_3$ , defined by:

$$\begin{aligned} x_1 &= (1, \eta_1 \eta_3), & y_1 &= (1 - \eta_1 \eta_2, \eta_1(1 - \eta_2)), \\ x_2 &= (1, \eta_1), & y_2 &= (1 - \eta_1 \eta_2 \eta_3, \eta_1 \eta_2(1 - \eta_3)), \\ x_3 &= (1 - \eta_1 \eta_2, 1 - \eta_2), & y_3 &= (1, \eta_1 \eta_2 \eta_3), \\ x_4 &= (1 - \eta_1 \eta_2 \eta_3, \eta_1 \eta_2(1 - \eta_3)), & y_4 &= (1, \eta_1), \\ x_5 &= (1 - \eta_1 \eta_2 \eta_3, \eta_1(1 - \eta_2 \eta_3)), & y_5 &= (1, \eta_1 \eta_2) \end{aligned}$$

**Case 3:**  $\tau_1$  and  $\tau_2$  share a common vertex

The integral  $\hat{A}_{\tau_1, \tau_2}$  can be decomposed as follows:

$$\hat{A}_{\tau_1, \tau_2}^{ij} = \int_{(0,1)^4} \xi^3 \eta_2 (k_{ij}(\xi x_1, \xi y_2) + k_{ij}(\xi x_2, \xi y_2)) d\eta_1 d\eta_2 d\eta_3 d\xi$$

where  $x_l$  and  $y_l$  are functions of  $\eta_1, \eta_2, \eta_3$ , defined by:

$$\begin{aligned} x_1 &= (1, \eta_1), & y_1 &= (\eta_2, \eta_2\eta_3), \\ x_2 &= (\eta_2, \eta_2\eta_3), & y_2 &= (1, \eta_1). \end{aligned}$$

**Case 4:**  $\tau_1 \cap \tau_2 = \emptyset$

If  $\tau_1$  and  $\tau_2$  are disjoint, the integral can be approximated via standard Gaussian quadrature, and there is no need to employ relative coordinates.

### 4.9.3 Regularization of the variational problem

Unfortunately, the solution of the variational problem  $(\mathcal{G})$  incurs numerical artifacts, which are mainly due to the fact that the solution  $\varphi$  to  $(\mathcal{B})$  is expected to blow up near the boundary  $\partial\mathbb{D}_1$ , see [307, 306, 305] about this general behavior. To illustrate this point, let us consider the following concrete example; see [95] or [194]. Let the function  $\varphi \in \tilde{H}^{1/2}(\mathbb{D}_1)$  be defined by:

$$\forall ((x_1, x_2), 0) \in \mathbb{D}_1, \quad \varphi(x) = \frac{4}{\pi\sqrt{1-|x|^2}}.$$

As we have seen in (4.46), this function satisfies the following properties:

$$\frac{1}{4\pi} \int_{\mathbb{D}_1} \frac{1}{|x-y|} \varphi(y) \, ds(y) = 1, \text{ and } \int_{\mathbb{D}_1} \varphi(x) \, ds(x) = 8,$$

In this case, attempting to recover  $\varphi$  via the numerical resolution of the variational problem:

$$\forall \psi \in \tilde{H}^{-1/2}(\mathbb{D}_1), \quad \int_{\mathbb{D}_1} \frac{1}{4\pi} \int_{\mathbb{D}_1} \frac{1}{|x-y|} \varphi(y) \psi(x) \, ds(y) \, ds(x) = \int_{\mathbb{D}_1} \psi(x) \, ds(x)$$

yields the noisy discretized solution  $\hat{\varphi}$ , shown in Fig. 4.8b. We can compare this to the truncated graph of the exact solution Section 4.9.3, to see that the numerical solution is of poor quality.

It turns out, that a simple regularization of problem  $(\mathcal{G})$  eventually remedies this problem. Let us indeed consider the regularized problem:

$$\forall \psi \in \tilde{H}^{-1/2}(\mathbb{D}_1), \quad a_\eta(\varphi_\eta, \psi) + \int_{\mathbb{D}_1} T_L \varphi(x) \psi(x) \, ds(x) = \int_{\mathbb{D}_1} f(x) \psi(x) \, ds(x).$$

where  $\eta > 0$  is some a regularization parameter and  $a_\eta(\cdot, \cdot)$  is a regularizing bilinear form. In the particular case being analyzed, we choose:

$$a_\eta(\varphi, \psi) := \eta \int_{\mathbb{D}_1} \nabla \varphi \cdot \nabla \psi \, ds(x),$$

so that the problem reduces to:

$$\forall \psi \in \tilde{H}^{-1/2}(\mathbb{D}_1), \quad \eta \int_{\mathbb{D}_1} \nabla \varphi_\eta(x) \cdot \nabla \psi(x) + \int_{\mathbb{D}_1} \frac{1}{4\pi} \int_{\mathbb{D}_1} \frac{1}{|x-y|} \varphi_\eta(y) \psi(x) \, ds(y) \, ds(x) = \int_{\mathbb{D}_1} \psi(x) \, ds(x).$$

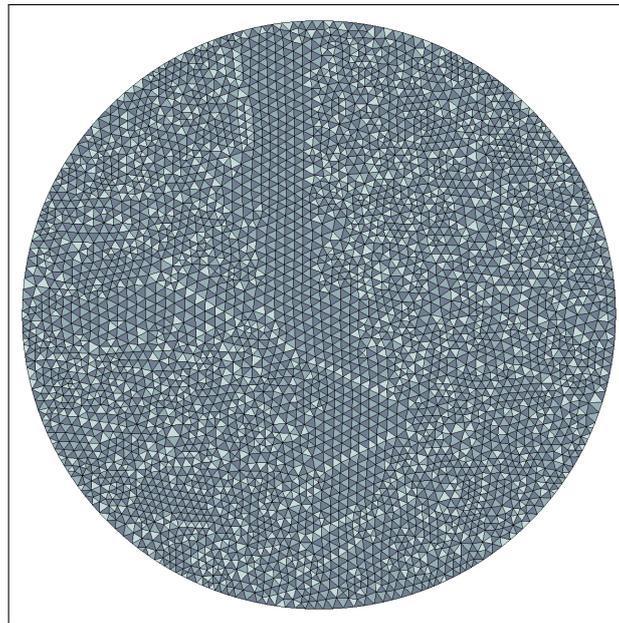
### 4.9.4 Numerical validation

To numerically validate our approach, we define three metrics that depend on the maximum mesh size  $h$  and the parameter  $\eta$ . These metrics are:

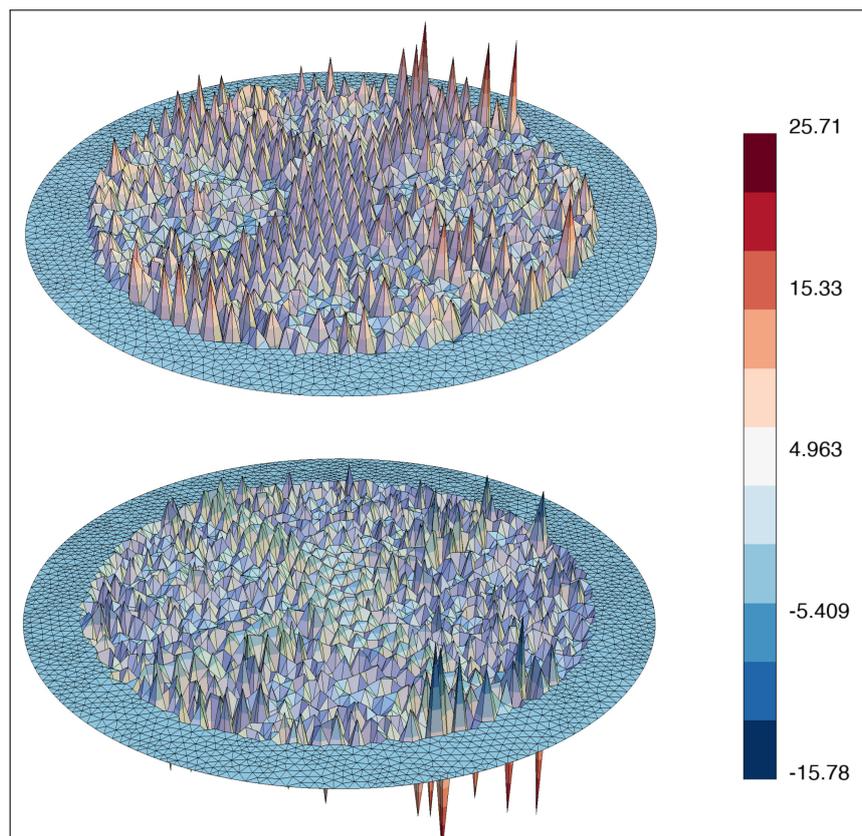
1. The squared error of the resolution.

$$\mathcal{R}(h, \eta) := \int_{\mathcal{T}} \left( \frac{1}{4\pi} \int_{\mathcal{T}} \frac{1}{|x-y|} \hat{\varphi}_\eta \, dy - 1 \right)^2 \, dx,$$

which calculates the discrepancy between the potentials generated by the numerical and exact equations.

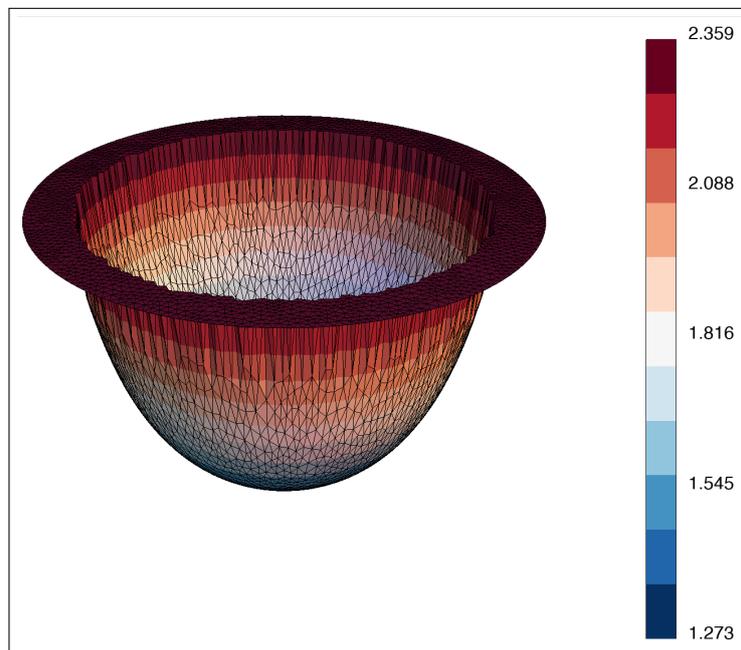


(a) Mesh  $\mathcal{T}$  representing a discretization of  $\mathbb{D}_1$ .

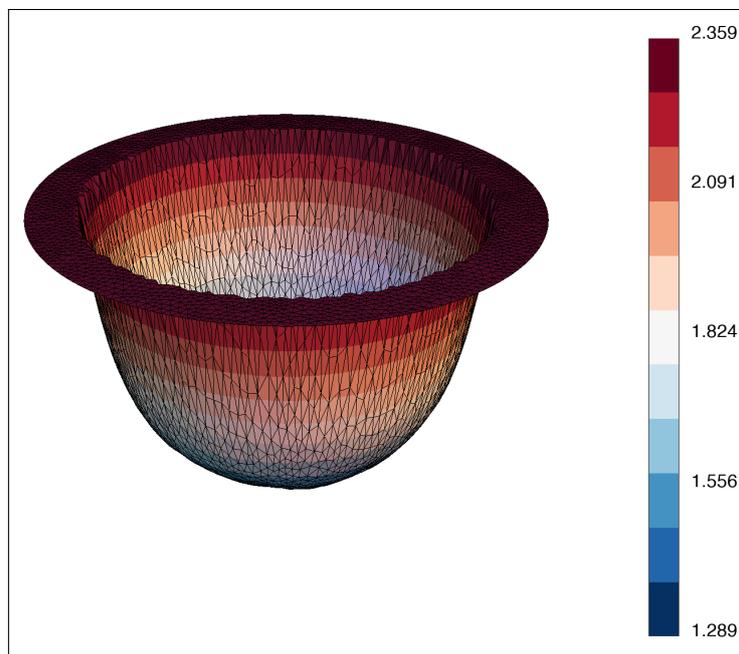


(b) Plot of the unregularized numerical solution  $\hat{\varphi}$ , truncated to zero near the boundary. The upper image shows the positive values while the bottom image shows the negative values. The colorbar on the right represents the values attained by the function on the vertices of the mesh.

Figure 4.8: Numerical discretizations of  $\mathbb{D}_1$  and numerical solution associated to the boundary integral equation problem  $(\mathcal{G})$ .



(a) Exact solution  $\varphi$  truncated to a subset smaller than the unit disk.



(b) Regularized numerical solution  $\hat{\varphi}_\eta$  truncated to a subset smaller than the unit disk.

Figure 4.9: Numerical plots of the function  $\varphi$  and the regularized solution  $\hat{\varphi}_\eta$ . Both plots are truncated to the same area to showcase their behavior on the interior, and not at the edge of the disk where both solutions blow up to infinity. Here  $\eta = 1e - 5$ ,  $h = 0.1$ .

2. The squared error of the average.

$$\mathcal{A}(h, \eta) := \left( \int_{\mathcal{T}} \hat{\varphi}_\eta dx - 8 \right)^2,$$

which measures the discrepancy between the averages of the exact and numerical solutions.

3. The squared approximation error.

$$\mathcal{E}(h, \eta) := \int_{\{|x| < 0.9, x \in \mathcal{T}\}} (\hat{\varphi}_\eta - \varphi)^2 dx,$$

which quantifies the error between the numerical and exact solutions.

The region  $|x| < 0.9$  is chosen in the third metric because the exact solution  $\varphi$  becomes singular at the boundary.

For our experiments, we use  $\mathbb{P}_1$  finite elements and we evaluate the metrics on 50 uniformly spaced values along the intervals  $0.00001 < \eta < 0.1$  and  $0.1 < h < 1$ , respectively. We plot the results on a graph as a function of both parameters in [Fig. 4.10](#). In all of the plots, as  $\eta$  and  $h$  go down to zero, all the quantities tend to 0. Note that, in [Fig. 4.10b](#) the regularization parameter  $\eta$  should not be taken too small when compared to the mesh size, to obtain a good approximation.

#### 4.9.5 Extension to the vector-valued case

The previous method can be straightforwardly adapted to an integral equation featuring a vector-valued function. As is the case in [Section 4.8](#), one seeks to solve an integral equation of the type:

$$T_L \varphi(x) = \int_{\mathbb{D}_1} L(x, y) \varphi(y) ds(y),$$

where this time the kernel function  $L(x, y)$  represents a matrix operator and the sought function  $\varphi$  is now vector-valued. A similar Galerkin method as that in [Section 4.9.1](#) is used to decompose the integrals on the reference elements. Employing similar arguments,  $T_L$  is decomposed into:

$$\int_{\mathbb{D}_1} T_L \varphi(x) \psi(x) ds(x) = \sum_{\tau_1, \tau_2 \in \mathcal{T}} A_{\tau_1, \tau_2},$$

where the vector-valued quantity  $A_{\tau_1, \tau_2}$  is defined by:

$$A_{\tau_1, \tau_2} = \int_{\tau_1} \psi(x) \left( \text{p.v.} \int_{\tau_2} L(x, y) \varphi(y) ds(y) \right) ds(x).$$

In this case the corresponding functions,  $k_{ij} : \mathbb{D}_1 \times \mathbb{D}_1 \rightarrow \mathbb{R}^d$  defined in [\(4.78\)](#) are written analogously in terms of the matrix product:

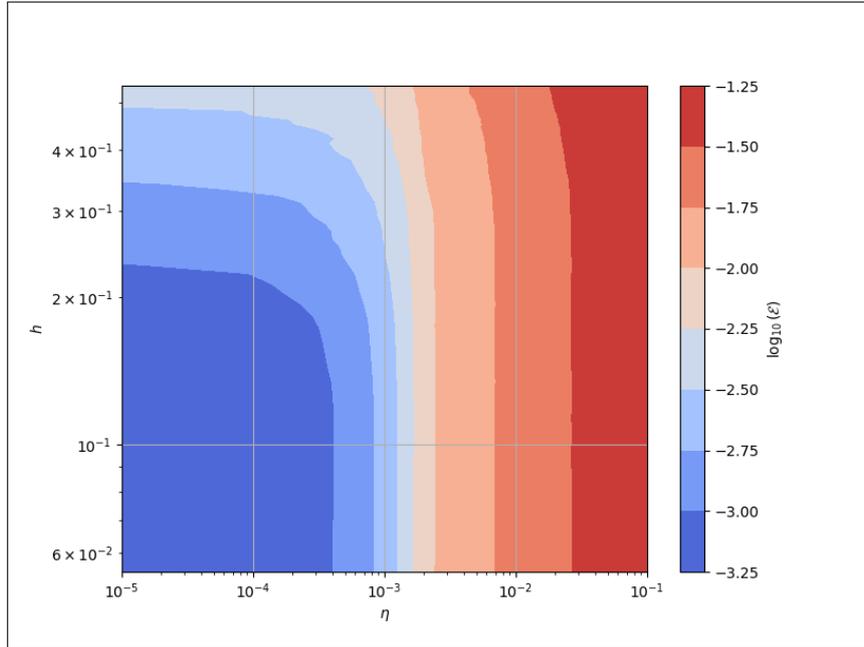
$$k_{ij}(x, y) = \left[ L(T_{\tau_1}(\hat{x}), T_{\tau_2}(\hat{y})) \hat{b}_{\tau_2, j}(\hat{y}) \right] \cdot \hat{b}_{\tau_1, i}(\hat{x}) g_{\tau_1, \tau_2}(\hat{x}, \hat{y}),$$

so that we can re-utilize the same integration rules described in the end of [Section 4.9.2](#). Similar to the previous section, we validate our approach utilizing the Mindlin kernel in 3D [\(4.65\)](#) from [Section 4.8](#). We only look at the resolution error  $\mathcal{R}(h, \eta)$ , since neither the value for the average or a exact solution is known analytically, to the best of our knowledge. This time we seek to solve the regularized vector valued problem:

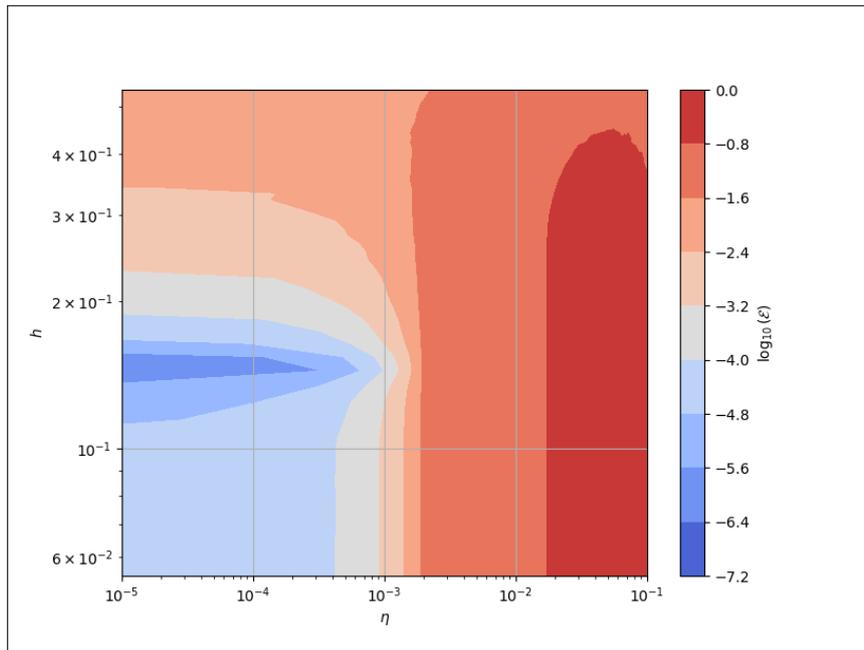
$$\forall \psi \in H^{1/2}(\mathbb{D}_1)^d, \quad a_\eta(\varphi_\eta, \psi) + \int_{\mathbb{D}_1} T_L \varphi_\eta(x) \cdot \psi(x) ds(x) = \int_{\mathbb{D}_1} f(x) \cdot \psi(x) ds(x).$$

In general, we have found that choosing the bilinear form for the elasticity equation as the regularizing term, yielded the best numerical results.

$$a_\eta(\varphi, \psi) := \eta \int_{\mathbb{D}_1} Ae(\varphi) : e(\psi) ds(x)$$

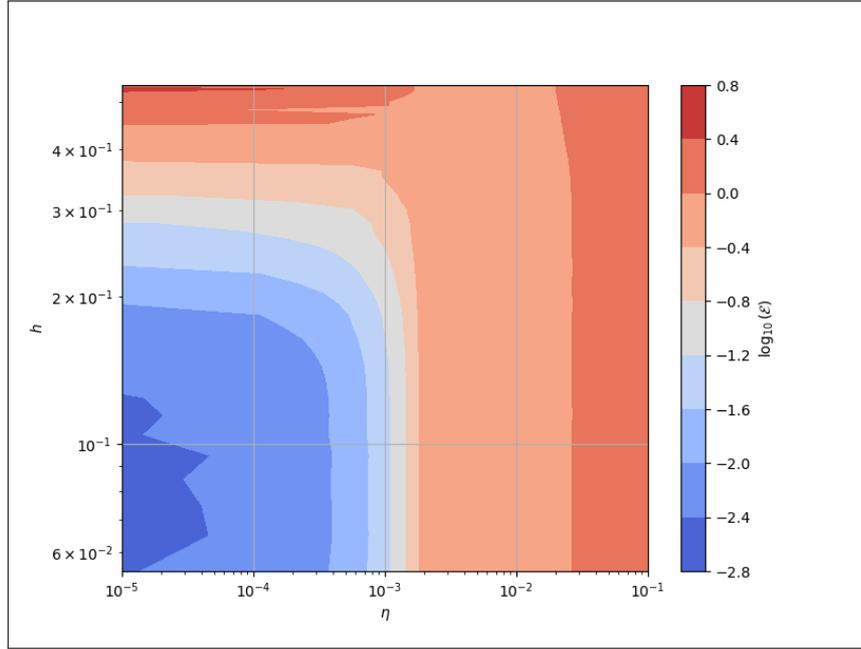


(a) Plot of the numerical resolution error  $\mathcal{R}(h, \eta)$ .



(b) Plot of the numerical error of the average  $\mathcal{A}(h, \eta)$ .

Figure 4.10: Plots of the numerical errors defined in Section 4.9.4. In each plot, the  $h$ ,  $\eta$  and error values have been scaled logarithmically.

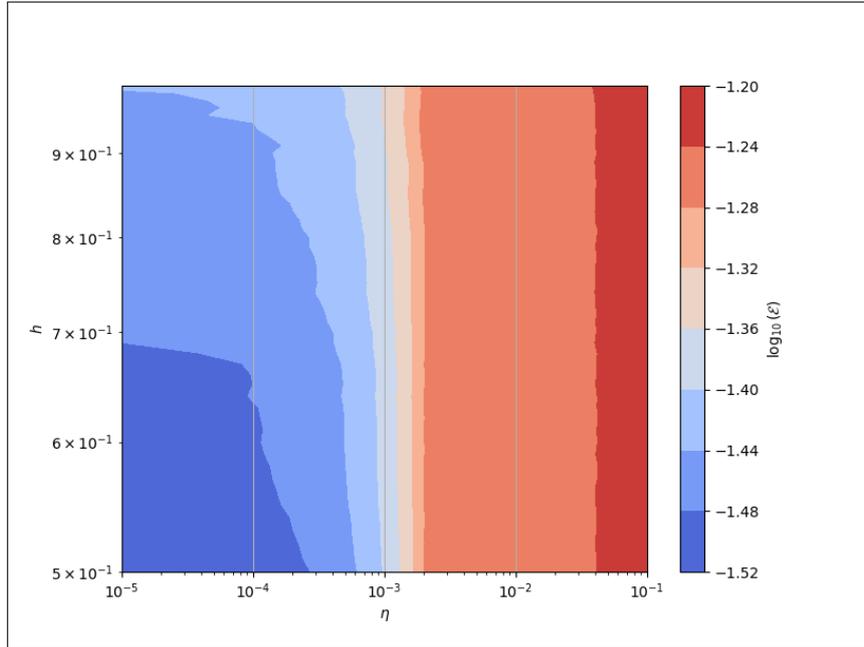


(c) Plot of the numerical squared approximation error  $\mathcal{E}(h, \eta)$ .

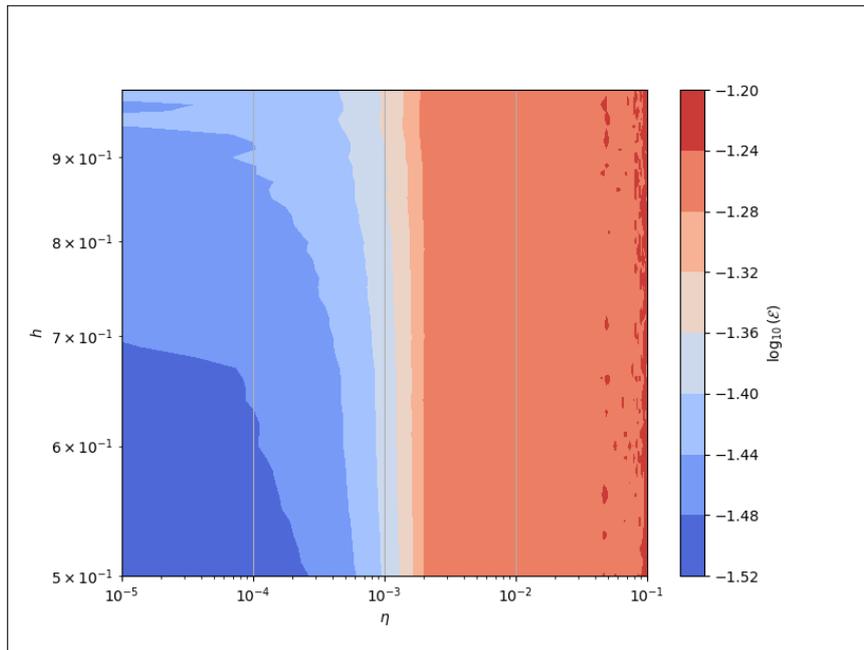
Figure 4.10: (cont.) Plots of the numerical errors defined in Section 4.9.4. In each plot, the  $h$ ,  $\eta$  and error values have been scaled logarithmically.

For our experiments, we have divided the intervals  $0.5 < h < 1$  and  $0.00001 < \eta < 0.1$  into 50 uniformly spaced points. Furthermore, we have chosen the physical parameters  $\mu = 67.5676$  and  $\nu = 0.48$ . We conduct three experiments, for  $f = e_i$ ,  $i = 1, 2, 3$ , representing the canonical directions of  $\mathbb{R}^d$ . The error plots are presented in Fig. 4.12 while the plots of the solutions when  $h = 0.5$ ,  $\eta = 0.00001$  is shown in Fig. 4.11.

The error behaves like in the previous section. As both parameters  $h$  and  $\eta$  decrease, so does the error. Analyzing the solution plots, one can see that the solution  $\phi_\eta e_1$  and  $e_2$  are the same, but rotated by a 90 degree angle. On the other hand, the solution plot for  $e_3$  differs from the previous two and furthermore it is totally symmetric, just like the solution plot of the scalar case in Section 4.9.3. This is to be expected, since the kernel  $L$  is anisotropic in the third row of its entries.

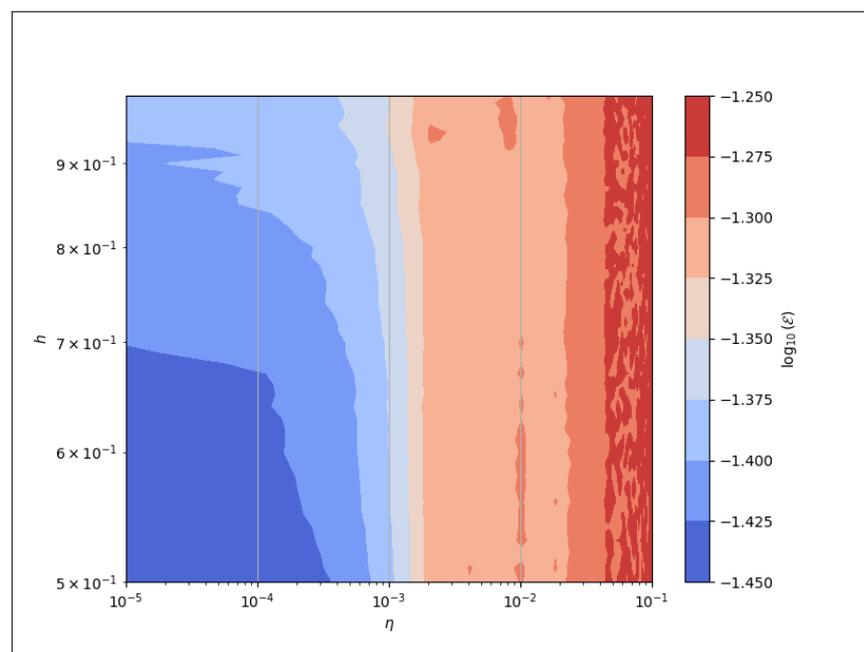


(a) Plot of the numerical resolution error  $\mathcal{R}(h, \eta)$  when solving for  $e_1$ .



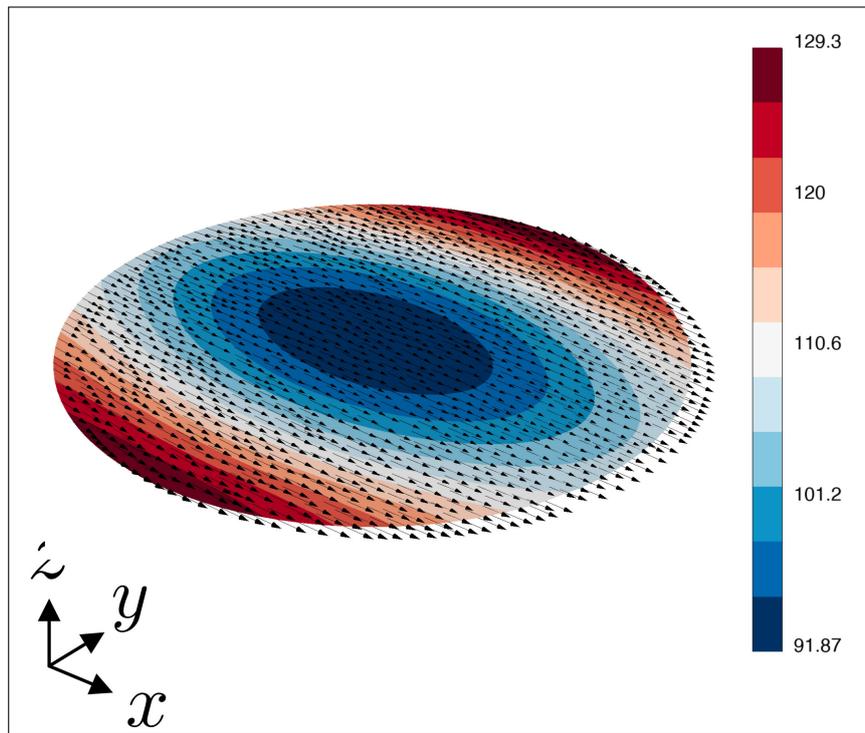
(b) Plot of the numerical resolution error  $\mathcal{R}(h, \eta)$  when solving for  $e_2$ .

Figure 4.11: Plots of the resolution error  $\mathcal{R}(h, \eta)$  for different right hand sides. In each plot, the  $h$ ,  $\eta$  and error values have been scaled logarithmically.

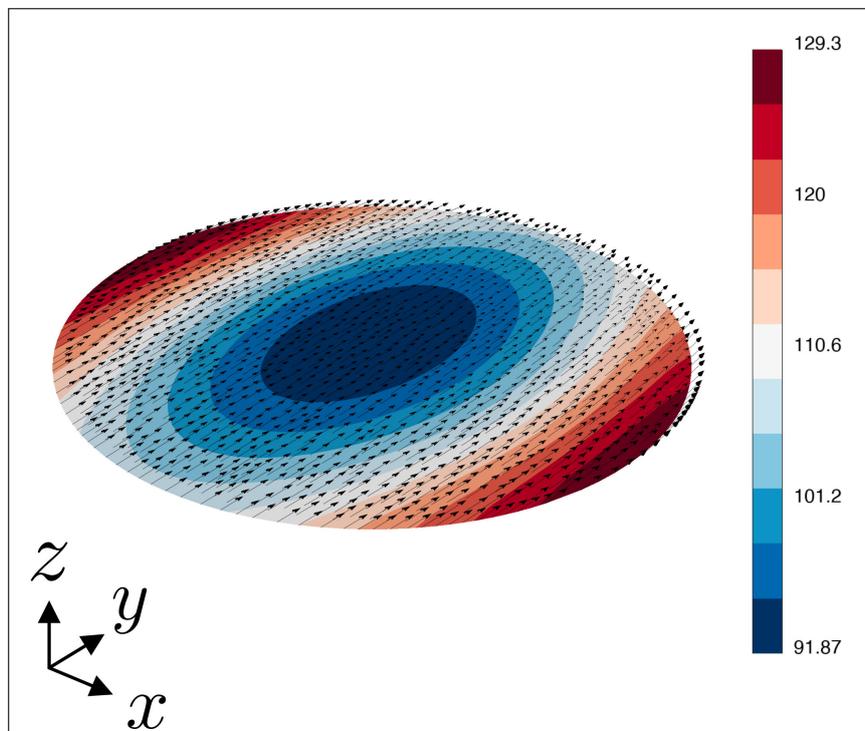


(c) Plot of the numerical resolution error  $\mathcal{R}(h, \eta)$  when solving for  $e_3$ .

Figure 4.11: (cont.) Plots of the resolution error  $\mathcal{R}(h, \eta)$  for different right hand sides. In each plot, the  $h$ ,  $\eta$  and error values have been scaled logarithmically.

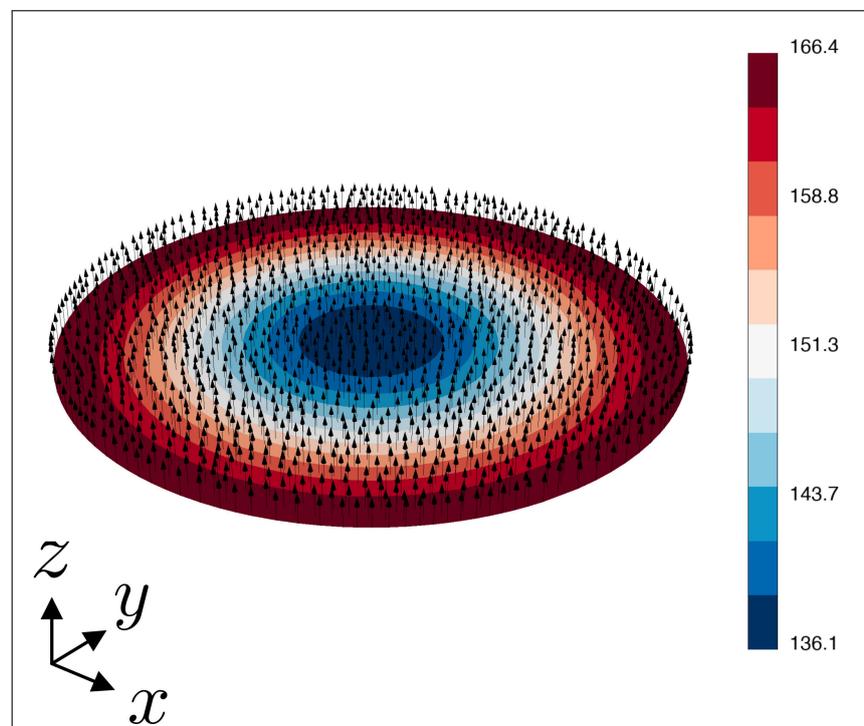


(a) Plot of the solution when  $f = e_1$ .



(b) Plot of the solution when  $f = e_2$ .

Figure 4.12: Plots of the approximated solutions for different values of  $f$ .



(c) Plot of the solution when  $f = e_3$ .

Figure 4.12: (cont.) Plots of the approximated solutions for different values of  $f$ .

## 4.10 Numerical method

Following the previous sections, we can see the results follow exactly the structure of (4.20):

$$J(G_{x_0, \varepsilon}) = J(G) + \rho(\varepsilon)d_T(G)(x_0) + o(\rho(\varepsilon)).$$

for a combination of  $\rho(\varepsilon)$  and  $d_T(G)(x_0)$ . Consistent with the methods discussed in Chapter 3, we can add a topological step to Algorithm 3 to account for the topological derivative  $d_T$  into the boundary optimization process.

Practically speaking, if we suppose we have a finite element space discretization over a mesh  $\mathcal{T}$ , we can loop over all the nodes associated with the degrees of freedom to interpolate this function. Then we select the most negative value. In particular, for continuous  $\mathbb{P}_1$  elements, we just need to loop over the mesh vertices and find the greatest negative value. At this negative value, we would nucleate a small hole of radius  $\varepsilon > 0$ . If there are no negative values, then we know that we are at a local minimum, and we don't add a new region.

The full shape and topology optimization algorithm is detailed in Algorithm 4.

---

**Algorithm 4:** Shape and topology optimization of the region  $G \subset \partial\Omega$ .

---

**Input:** Mesh  $\mathcal{T}^0$  of  $\Omega$ , whose discretized boundary  $\mathcal{B}^0$  contains two submeshes  $\mathcal{B}_{\text{int}}^0$  of  $G^0$ , and  $\mathcal{B}_{\text{ext}}^0$  of  $\partial\Omega \setminus \overline{G^0}$ .

**for**  $n = 0, \dots, N - 1$  **do**

1. Compute the signed distance function  $d_{G^n}^{\partial\Omega}$  to  $G^n$  at the vertices of the mesh  $\mathcal{B}^n$  of  $\partial\Omega$ .
2. Infer a descent direction from  $\theta^n$  using the approximated shape derivative expression  $J'_\varepsilon(G^n)(\theta)$ , solving the state equation and adjoint equation if necessary.
3. Find the point  $c_0$  of greatest topological decrease:

$$c_0 = \arg \max_{x_0 \in \partial\Omega} |d_T(G)(x_0)| \quad \text{s.t.} \quad d_T(G)(x_0) < 0.$$

and nucleate a small hole  $\omega_{c_0, r}$  of radius  $r > 0$  by setting the level set function  $d_{G^n}^{\partial\Omega}(x) \leftarrow 0$  for  $x \in \omega_{c_0, r}$ .

4. Solve the advection equation for a fixed time-step  $\Delta t > 0$ :

$$\begin{cases} \frac{\partial \phi}{\partial t}(t, x) + \theta^n(x) \cdot \nabla_{\partial\Omega} \phi(t, x) = 0 & \text{for } (t, x) \in (0, \Delta t) \times \partial\Omega, \\ \phi(0, x) = d_{G^n}^{\partial\Omega}(x) & \text{for } x \in \partial\Omega, \end{cases}$$

on the total mesh  $\mathcal{B}^n$  of  $\partial\Omega$ . A new level set function  $\phi^{n+1} = \phi(\Delta t, \cdot)$  is obtained for

$$G^{n+1} = \{x \in \partial\Omega, \phi^{n+1}(x) < 0\}.$$

5. From the datum of  $\phi^{n+1}$  at the vertices of  $\mathcal{B}^n$ , create a new, high-quality mesh  $\mathcal{B}^{n+1}$  of  $\partial\Omega$  made of two submeshes  $\mathcal{B}_{\text{int}}^{n+1}$  and  $\mathcal{B}_{\text{ext}}^{n+1}$  for  $G^{n+1}$  and  $\partial\Omega \setminus \overline{G^{n+1}}$ , respectively.

**end**

**Output:** Mesh  $\mathcal{T}^N$  whose discretized boundary  $\mathcal{B}^N$  contains an explicit discretization  $\mathcal{B}_{\text{int}}^N$  of  $G^N$ .

---

## 4.11 Optimization of the repartition of cathode-anode regions for a direct current electroosmotic mixer

Our first applicative instance of optimal design problems featuring boundary condition regions is motivated by the field of microfluidics, which investigates the handling of very small volumes of fluid, ranging between  $10^{-18}$  to  $10^{-9}$  L. This practice has aroused a tremendous enthusiasm over the last 20 years, heralding decisive advances in analytical chemistry, molecular biology and biomedical engineering (design of biochips and DNA micro-arrays, electrophoresis and liquid chromatography for proteins and DNA),

optics (design of microlens arrays), etc. We refer to e.g. [249, 308] for comprehensive introductions to this subject and its challenges.

Among the various operations of interest in this context, that of fluid mixing in microchannel is particularly crucial, and one method of choice in this perspective is electroosmosis [81, 217, 269, 282, 344]. This process consists in applying an electric field on a liquid-filled channel – thus leveraging the Coulomb force to electrically actuate the charged particles and ions within the fluid and triggering the pumping of electrolytic fluid through drag forces. The configuration and placement of electrode pairs, whose induced charges interact with the fluid, play a pivotal role in determining the electric field and the resulting fluid-dynamic velocity distribution.

In this section, we seek to optimize the design of an electroosmotic mixer (EMM). This device mixes two fluids by leveraging the electroosmosis phenomenon. It consists of three parts: the inflow compartment, the mixing chamber, and the outflow compartment. See the schematic in Fig. 4.13. Additionally, the surface of this device can be designed so that certain regions are positively charged (anode) and others are negatively charged (cathode). Our goal is to optimize the placement and geometry of these regions to enhance the mixing efficiency.

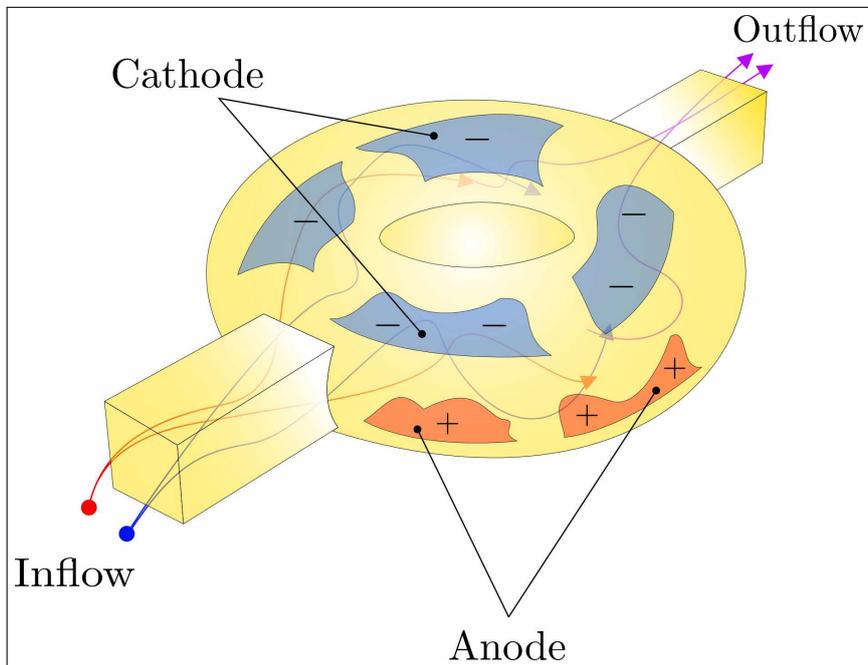


Figure 4.13: Illustration of an electroosmotic mixer. Two fluids (represented by blue/red arrows) go in via the inflow compartment, they get mixed in the mixing chamber which has a specific boundary repartition of anode and cathode regions. The final fluid (represented by purple arrows) goes out via the outflow compartment.

Our study relies on a simplified version of the model presented in [118]. Briefly, a stronger electrical potential results in a higher electrokinetic force within the channel, inducing more efficient disturbances and chaotic advection in the microflow. We thus aim to optimize the placement of electrodes to maximize the electric field  $|\nabla u|^2$  on the surface of the mixer. This approach differs from [118], which focuses on integrating the Navier-Stokes equations into the physical model and optimizing the “quality” of the final fluid.

#### 4.11.1 The optimization problem

We rephrase the problem above into a shape and topology optimization problem in the mathematical setting. In this context, the EMM is represented by a bounded Lipschitz domain  $\Omega \subset \mathbb{R}^3$ , whose boundary is decomposed into three disjoint pieces:  $\partial\Omega = \Gamma_A \cup \Gamma_C \cup \Gamma$ , where:

- The region  $\Gamma_C$  is the cathode, where the voltage potential is set to 0 zero;
- The region  $\Gamma_A$  is the anode, where a value  $u_{\text{in}} \neq 0$  is imposed;

- The device is perfectly insulated from the remaining region  $\Gamma$ .

We assume that the  $\Gamma_C$  and  $\Gamma_A$  are far away from each other in the sense that:

$$\forall \varepsilon > 0, \text{dist}(\Gamma_C, \Gamma_A) > 0.$$

For generality, we consider the minimization of the shape functional:

$$\min_{\Gamma_C, \Gamma_A \subset \partial\Omega} J(\Gamma_C, \Gamma_A), \text{ where } J(\Gamma_C, \Gamma_A) = \int_{\Omega} j(\nabla u_{A,C}) \, dx, \quad (4.79)$$

whose integrand  $j \in C^1(\mathbb{R}^d, \mathbb{R})$  satisfies appropriate growth conditions:

$$\exists C > 0, \forall \xi \in \mathbb{R}^d, \quad |j(\xi)| \leq C(1 + t^2), \quad |j'(\xi)| \leq C(1 + |\xi|), \text{ and } |j''(\xi)| \leq C.$$

The criterion (4.79) depends on the gradient  $\nabla u$  of the solution  $u$  to the boundary value problem:

$$\begin{cases} -\nabla \cdot (\gamma \nabla u_{A,C}) = 0 & \text{in } \Omega, \\ u_{A,C} = 0 & \text{on } \Gamma_C, \\ u_{A,C} = u_{\text{in}} & \text{on } \Gamma_A, \\ \gamma \frac{\partial u_{A,C}}{\partial n_{\partial\Omega}} = 0 & \text{on } \Gamma, \end{cases} \quad (4.80)$$

for a given input voltage  $u_{\text{in}} \in L^2(\partial\Omega)$ .

### 4.11.2 The shape derivative

It turns out that the boundary value problem (4.80) is weakly singular at the transition zones  $\Sigma_C = \overline{\partial\Gamma_C} \cap \overline{\partial\Gamma}$  and  $\Sigma_A = \overline{\partial\Gamma_A} \cap \overline{\partial\Gamma}$ , where the boundary conditions change from the homogeneous Neumann condition to a Dirichlet condition. As described in Section 4.2, we can smooth out these transition zones by using a Robin boundary condition. The exact problem can then be reformulated into an approximate version:

$$\min_{\Gamma_C, \Gamma_A \subset \partial\Omega} J_{\varepsilon}(\Gamma_C, \Gamma_A), \text{ where } J_{\varepsilon}(\Gamma_C, \Gamma_A) = \int_{\Omega} j(\nabla u_{A,C,\varepsilon}) \, dx \quad (4.81)$$

subject to:

$$\begin{cases} -\nabla \cdot (\gamma \nabla u_{A,C,\varepsilon}) = 0 & \text{in } \Omega, \\ \gamma \frac{\partial u_{A,C,\varepsilon}}{\partial n_{\partial\Omega}} + (h_{C,\varepsilon} + h_{A,\varepsilon}) u_{A,C,\varepsilon} - h_{A,\varepsilon} u_{\text{in}} = 0 & \text{on } \partial\Omega. \end{cases} \quad (4.82)$$

In this formulation, the cutoff function  $h_{A,\varepsilon} : \partial\Omega \rightarrow \mathbb{R}$  is defined by:

$$\forall x \in \partial\Omega, \quad h_{A,\varepsilon}(x) = \frac{1}{\varepsilon} h \left( \frac{d_{\Gamma_A}^{\partial\Omega}(x)}{\varepsilon} \right),$$

where  $d_{\Gamma_A}^{\partial\Omega}$  denotes the signed distance function to  $\Gamma_A$  on  $\partial\Omega$ , and  $h \in C^{\infty}(\mathbb{R})$  satisfies (4.8). The case for the cutoff function  $h_{C,\varepsilon}$  is defined analogously. Additionally, the weak formulation for the solution  $u_{A,C,\varepsilon} \in H^1(\Omega)$  to (4.82) is:

$$\forall v \in H^1(\Omega), \quad \int_{\Omega} \gamma \nabla u_{A,C,\varepsilon} \cdot \nabla v \, dx + \int_{\partial\Omega} (h_{C,\varepsilon} + h_{A,\varepsilon}) u_{A,C,\varepsilon} v \, ds = \int_{\partial\Omega} h_{A,\varepsilon} u_{\text{in}} v \, ds.$$

We can then compute the shape derivative of the approximated criterion  $J_{\varepsilon}(\Gamma_C, \Gamma_A)$ , along with the simplified formulas derived similarly to those in Section 4.2. The proof can be found in Appendix A.2.1.

**Proposition 4.4.** *The criterion  $J_\varepsilon(\Gamma_C, \Gamma_A)$  is shape differentiable at  $\theta = 0$  (with  $\theta \cdot n_{\partial\Omega} = 0$ ), and its shape derivative reads:*

$$J'_\varepsilon(\Gamma_C, \Gamma_A)(\theta) = \frac{1}{\varepsilon^2} \int_{\partial\Omega} h' \left( \frac{d_{\Gamma_A}^{\partial\Omega}}{\varepsilon} \right) \theta(\pi_{\Sigma_A}) \cdot n_{\Sigma_A}(\pi_{\Sigma_A})(u_{\text{in}} - u_{A,C,\varepsilon}) p_{A,C,\varepsilon} \, ds \\ - \frac{1}{\varepsilon^2} \int_{\partial\Omega} h' \left( \frac{d_{\Gamma_C}^{\partial\Omega}}{\varepsilon} \right) \theta(\pi_{\Sigma_C}) \cdot n_{\Sigma_C}(\pi_{\Sigma_C}) u_{A,C,\varepsilon} p_{A,C,\varepsilon} \, ds,$$

where  $p_{A,C,\varepsilon} \in H^1(\Omega)$  is the weak solution to the following equation:

$$\begin{cases} -\nabla \cdot (\gamma \nabla p_{A,C,\varepsilon}) = \nabla \cdot j'(\nabla u_{A,C,\varepsilon}) & \text{in } \Omega, \\ \gamma \frac{\partial p_{A,C,\varepsilon}}{\partial n_{\partial\Omega}} + (h_{C,\varepsilon} + h_{A,\varepsilon}) p_{A,C,\varepsilon} = -j'(\nabla u_{A,C,\varepsilon}) \cdot n_{\partial\Omega} & \text{on } \partial\Omega. \end{cases}$$

Using the same procedure as outlined in [Section 4.2.3](#), we can derive approximate formulas that conform to the typical Hadamard structure described in [\(4.4\)](#). Assuming that the regions around  $\Sigma_C$  and  $\Sigma_A$  are flat, and that the solutions  $u_{A,C,\varepsilon}$  and  $p_{A,C,\varepsilon}$  remain constant along the rays in these regions, we can obtain the following approximate formula:

$$J'_\varepsilon(\Gamma_C, \Gamma_A)(\theta) = \frac{1}{\varepsilon} \int_{\Sigma_A} (u_{\text{in}} - u_{A,C,\varepsilon}) p_{A,C,\varepsilon} \theta \cdot n_{\Sigma_A} \, ds \, ds - \frac{1}{\varepsilon} \int_{\Sigma_C} u_{A,C,\varepsilon} p_{A,C,\varepsilon} \theta \cdot n_{\Sigma_C} \, ds.$$

### 4.11.3 The topological derivative

The criterion considered in this section is slightly different from the criterion [Section 4.3](#), since it depends on the gradient of the solution to [\(4.79\)](#). This poses no problem as we are about to see that it is straightforward to adapt the analysis to the case where  $j$  is a function whose domain is  $\mathbb{R}^d$ . In this context, the solution  $u_{A,C}$  to [\(4.80\)](#) is the background solution. We consider two perturbations  $u_{C,\varepsilon}, u_{A,\varepsilon} \in H^1(\Omega)$  solutions to:

$$\begin{cases} -\nabla \cdot (\gamma \nabla u_{C,\varepsilon}) = 0 & \text{in } \Omega, \\ u_{C,\varepsilon} = 0 & \text{on } \Gamma_C \cup \omega_\varepsilon, \\ u_{C,\varepsilon} = u_{\text{in}} & \text{on } \Gamma_A, \\ \gamma \frac{\partial u_{C,\varepsilon}}{\partial n_{\partial\Omega}} = 0 & \text{on } \Gamma \setminus \overline{\omega_\varepsilon}, \end{cases} \quad \begin{cases} -\nabla \cdot (\gamma \nabla u_{A,\varepsilon}) = 0 & \text{in } \Omega, \\ u_{A,\varepsilon} = 0 & \text{on } \Gamma_C, \\ u_{A,\varepsilon} = u_{\text{in}} & \text{on } \Gamma_A \cup \omega_\varepsilon, \\ \gamma \frac{\partial u_{A,\varepsilon}}{\partial n_{\partial\Omega}} = 0 & \text{on } \Gamma \setminus \overline{\omega_\varepsilon}. \end{cases}$$

The solution  $u_{C,\varepsilon}$  corresponds to the topological perturbation of the cathode area, and  $u_{A,\varepsilon}$  corresponds to the perturbation of the anode area. Let us perform the analysis for the case where we seek to replace  $\omega_\varepsilon \subset \Gamma$  with  $\Gamma_A$ . The case for the cathode region  $\Gamma_C$  is analogous. As in the analysis performed in [Section 4.3](#), applying the fundamental theorem of calculus yields:

$$j(\nabla u_{A,\varepsilon}) = j(u_{A,C}) + \int_0^1 j'(\nabla u_{A,C} + t(\nabla u_{A,\varepsilon} - \nabla u_{A,C})) \cdot \nabla(u_{A,\varepsilon} - u_{A,C}) \, dt.$$

The twice continuous differentiability of  $j$  allows us to expand the function  $t \mapsto j'(\nabla u_{A,C} + t\nabla(u_{A,\varepsilon} - u_{A,C}))$  at  $t = 0$ :

$$j'(\nabla u_{A,C} + t\nabla(u_{A,\varepsilon} - u_{A,C})) = j'(\nabla u_{A,C}) + t j''(\nabla u_{A,C}) \cdot \nabla(u_{A,\varepsilon} - u_{A,C}) + o(t|\nabla u_{A,\varepsilon} - u_{A,C}|).$$

If we define  $r_{A,\varepsilon} := u_{A,\varepsilon} - u_{A,C}$ , we look for an asymptotic of the following quantity:

$$\int_{\Omega} j'(\nabla u_{A,C}) \cdot \nabla r_{A,\varepsilon} \, dx = \int_{\partial\Omega} r_{A,\varepsilon} j'(\nabla u_{A,C}) \cdot n_{\partial\Omega} \, ds(x) - \int_{\Omega} r_{A,\varepsilon} \nabla \cdot j'(\nabla u_{A,C}) \, dx.$$

In this manner, we have the following proposition for both cases where  $\Gamma$  is replaced with either  $\Gamma_C$  or  $\Gamma_A$ . The proof is very similar to that of [\(4.47\)](#), with only the adjoint solution  $p$  having a different characterization, and thus it is omitted.

**Proposition 4.5.** *Let  $x_0 \in \Gamma$  be a given point. Then,*

(i) *The perturbed criterion  $J((\Gamma_C)_{x_0,\varepsilon}, \Gamma_A)$ , accounting for the addition of the surfacic disk  $\omega_{x_0,\varepsilon} \subset \Gamma$  to  $\Gamma_C$ , has the following asymptotic expansion:*

$$J((\Gamma_C)_{x_0,\varepsilon}, \Gamma_A) = \begin{cases} J(\Gamma_C, \Gamma_A) - \frac{\pi}{|\log \varepsilon|} \gamma(x_0) u_{A,C}(x_0) p_{A,C}(x_0) + o\left(\frac{1}{|\log \varepsilon|}\right) & \text{if } d = 2, \\ J(\Gamma_C, \Gamma_A) - 4\varepsilon \gamma(x_0) u_{A,C}(x_0) p_{A,C}(x_0) + o(\varepsilon) & \text{if } d = 3. \end{cases}$$

(ii) *The perturbed criterion  $J(\Gamma_C, (\Gamma_A)_{x_0,\varepsilon})$ , accounting for the addition of  $\omega_{x_0,\varepsilon} \subset \Gamma$  to  $\Gamma_A$ , has the following asymptotic expansion:*

$$J(\Gamma_C, (\Gamma_A)_{x_0,\varepsilon}) = \begin{cases} J(\Gamma_C, \Gamma_A) + \frac{\pi}{|\log \varepsilon|} \gamma(x_0) (u_{\text{in}} - u_{A,C}(x_0)) p_{A,C}(x_0) + o\left(\frac{1}{|\log \varepsilon|}\right) & \text{if } d = 2, \\ J(\Gamma_C, \Gamma_A) + 4\varepsilon \gamma(x_0) (u_{\text{in}} - u_{A,C}(x_0)) p_{A,C}(x_0) + o(\varepsilon) & \text{if } d = 3. \end{cases}$$

In the above, the adjoint state  $p_{A,C} \in H^1(\Omega)$  denote the solution to the following boundary value problem:

$$\begin{cases} -\nabla \cdot (\gamma \nabla p_{A,C}) = \nabla \cdot (j'(\nabla u_{A,C})) & \text{in } \Omega, \\ \gamma \frac{\partial p_{A,C}}{\partial n} = -j'(\nabla u_{A,C}) \cdot n & \text{on } \partial\Omega. \end{cases}$$

#### 4.11.4 Experiment setup

We focus on maximizing the voltage potential, as motivated at the beginning of this section. The shape optimization problem is formulated as follows:

$$\max_{\Gamma_A, \Gamma_C \subset \partial\Omega} \int_{\Omega} |\gamma \nabla u_{A,C}|^2 dx - \ell_1 \text{Area}(\Gamma_A) - \ell_2 \text{Area}(\Gamma_C),$$

where  $u_{A,C}$  is the solution to (4.82), and  $\ell_1, \ell_2 > 0$  are weak penalization parameters for the surface area of the anode and cathode regions, respectively.

We also consider the case where the contours of both regions  $\Gamma_A$  and  $\Gamma_C$  are penalized:

$$\max_{\Gamma_A, \Gamma_C \subset \partial\Omega} \int_{\Omega} |\gamma \nabla u_{A,C}|^2 dx - \ell_1 \text{Area}(\Gamma_A) - \ell_2 \text{Area}(\Gamma_C) - \ell_3 \text{Cont}(\Gamma_A) - \ell_4 \text{Cont}(\Gamma_C),$$

where  $\ell_3 > 0$  and  $\ell_4 > 0$  are penalization parameters for the contours of the anode and cathode regions, respectively.

The domain  $\Omega$  is represented by a tetrahedral mesh  $\mathcal{T}$ , consisting of 29k vertices and 135k tetrahedrons, as illustrated in Fig. 4.14. This mesh includes a composite structure formed by the union of a torus, with a major radius  $R = 10u$  and a minor radius  $r = 5u$ , along with inflow/outflow rectangular compartments. Here,  $u$  denotes a spatial unit of length. In the broader literature [81, 347, 195], this configuration is often referred to as a “ring” EMM due to the torus at its center. This design is widely adopted in the field, which is why we selected it. Additionally, its non-flat geometry makes it an interesting example for testing our boundary optimization technology. The optimization is carried through Algorithm 4, utilizing the shape derivative of  $J'(\Gamma_A, \Gamma_C)$ , Area, and Cont. The topological derivative is that one inferred from Proposition 4.5.

For our tests, we fixed the anode region on the upper part of the inflow/outflow chambers, the cathode region on the lower part, and the inflow/outflow regions. These regions were not subject to optimization, meaning no topological or geometrical changes were made to them. In contrast, the remaining boundary was subject to the nucleation and geometrical optimization of anode/cathode regions. Throughout the process, we alternated between the evolution of  $\Gamma_A$  and  $\Gamma_C$ . Specifically, in each iteration, either the domain  $\Gamma_A$  or  $\Gamma_C$  (but not both) was deformed according to the procedure described in Chapter 3. We conducted three experiments, with the numerical parameter values shown in Table 4.1. In all experiments, new topological regions are added every 10 iterations, continuing until 100 iterations are reached, after which only geometric optimization is performed.

#### 4.11.5 Analysis of results

The snapshots (Fig. 4.15) from the first experiment (without contour penalization) reveal an intriguing behavior. Topologically, the method tends to create opposing regions on opposite sides of the mixer.

Parameter	Value
$\varepsilon$	0.001
$\ell_1$	0.0001
$\ell_2$	0.0001
$\ell_3$	0
$\ell_4$	0
$\alpha$	4
$h_{max}$	0.05
$h_{min}$	0.005

(a) Experiment 1.

Parameter	Value
$\varepsilon$	0.001
$\ell_1$	0.001
$\ell_2$	0.001
$\ell_3$	0
$\ell_4$	0
$\alpha$	4
$h_{max}$	0.05
$h_{min}$	0.005

(b) Experiment 2.

Parameter	Value
$\varepsilon$	0.001
$\ell_1$	0.001
$\ell_2$	0.001
$\ell_3$	0.001
$\ell_4$	0.001
$\alpha$	4
$h_{max}$	0.05
$h_{min}$	0.005

(c) Experiment 2.

Table 4.1: Values of the parameters used in the optimization process.

Initially, it forms an anode region on the upper side and a corresponding cathode region on the lower side, positioned directly across the middle. As the algorithm progresses, it stops introducing new regions and instead focuses on geometrical optimization. This leads to the expansion of the existing regions, which gradually attempt to cover the entire upper and lower sides of the mixer while maintaining a separation between them. Eventually, the anode and cathode regions nearly encompass the upper and lower sides, respectively. At this stage, the challenge shifts to homogenizing the design, resulting in a maze-like structure. The homogenization of the design is not a new phenomenon; it is a common occurrence in these types of scenarios, especially when dealing with the Poisson equation in the context of thermal cooling (see, for instance, [Section 4.2.4](#)). For this experiment, the final mesh consists of approximately 5 million tetrahedrons.

In the second experiment ([Fig. 4.16](#)), we use a larger area penalization parameter. This results in disjoint structures that remain separate throughout the process. Similar to the previous experiment, branches appear, resembling the structures discussed in [Section 4.2.4](#), which is expected given the similarity of the problems being modeled. However, the branches are more separated due to the higher penalization parameter. Additionally, as in the previous experiment, the top and bottom regions of the mixer do not converge, as maintaining their separation proves advantageous. The final mesh consists of 1.8 million tetrahedron, less than the first experiment due to the less intricate patterns that need meshing.

For the third experiment ([Fig. 4.17](#)), we utilize a contour penalization on both regions, while utilizing the same area penalization as that of the second experiment. Instantly, one can see that the patterns are way less simpler and present fewer branches, due to the penalization. The final mesh also consists of 1.8 million tetrahedron, similar to the first experiment.

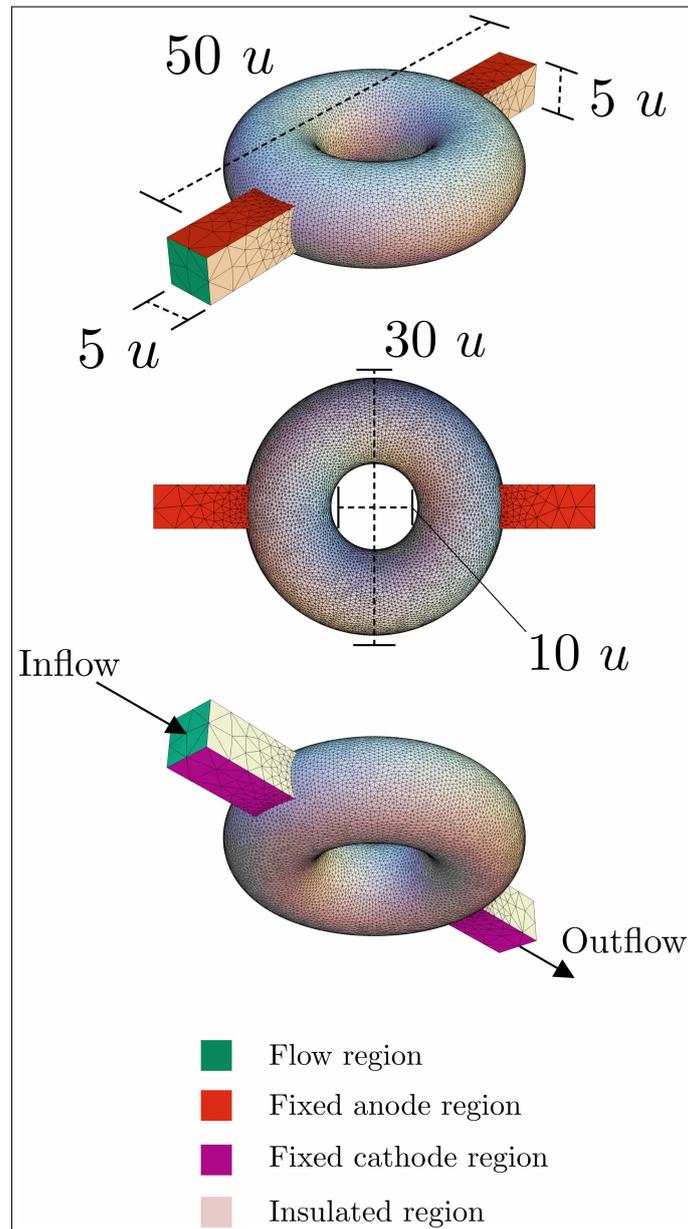


Figure 4.14: Different view angles of the tetrahedral mesh  $\mathcal{T}$  employed in the simulations. The red and pink regions correspond to the anode and cathode regions, respectively, which are fixed and not subject to optimization. The green region represents the flow region, which is also fixed and not subject to optimization. The remaining region corresponds to the insulated region (homogeneous Neumann boundary), which will be subject to optimization.

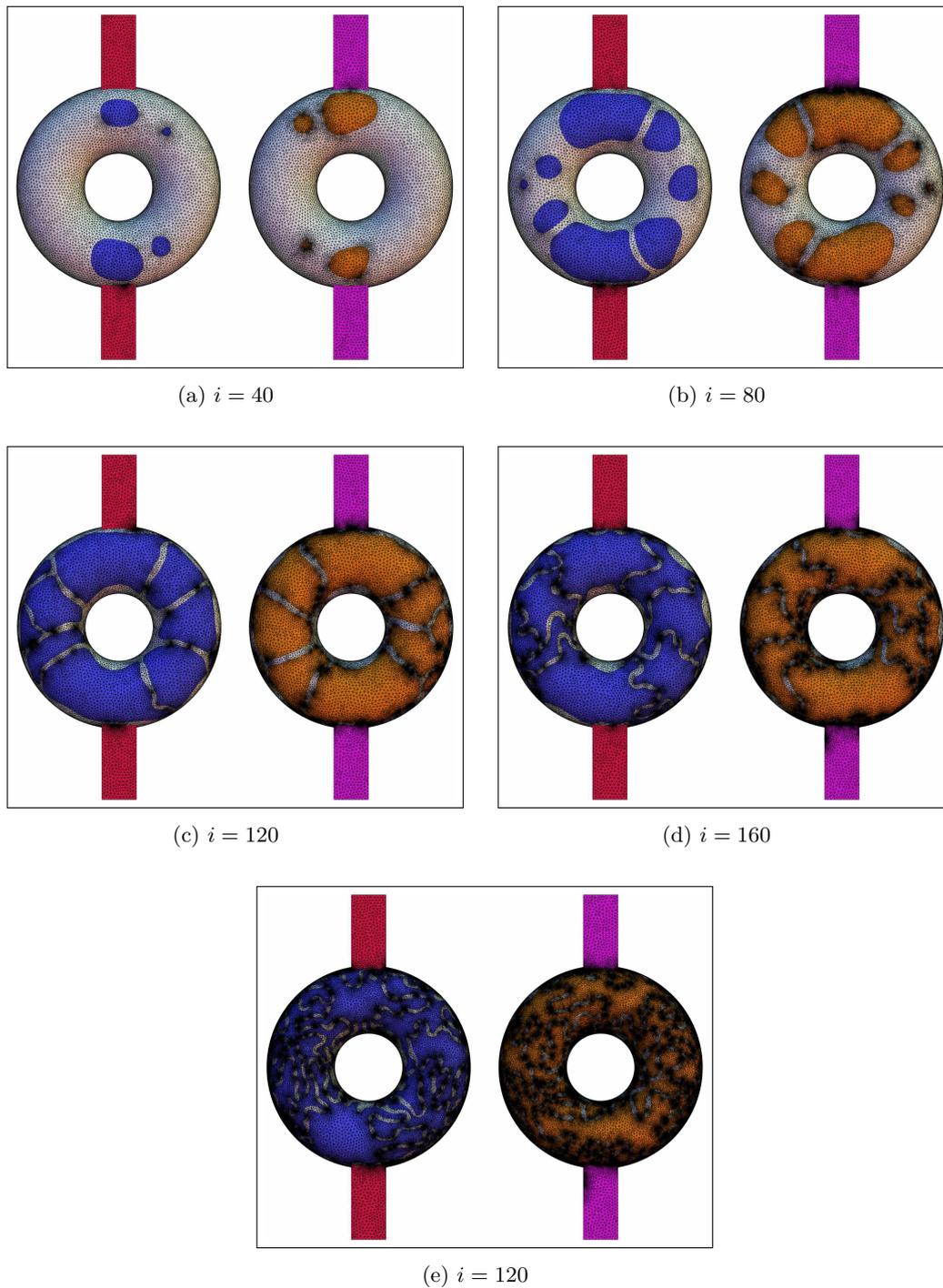
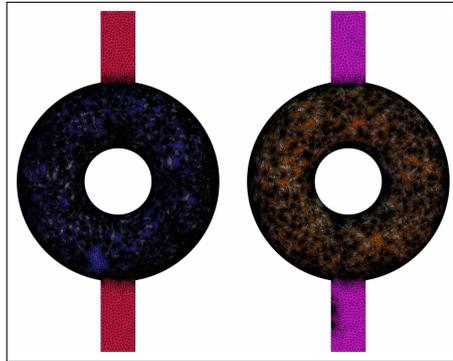
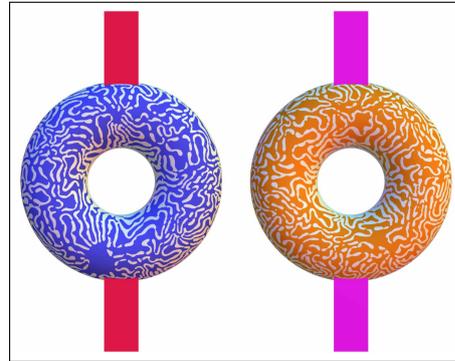


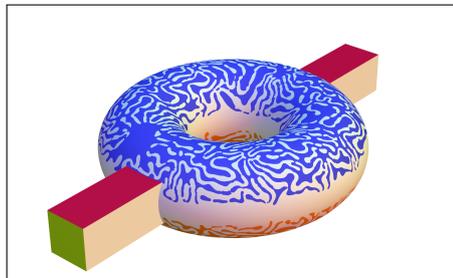
Figure 4.15: Snapshots of the optimization process of the boundary repartition. Here  $i$  indicates the number of iterations that have passed. The blue color corresponds to the cathode region and the orange color the anode region.



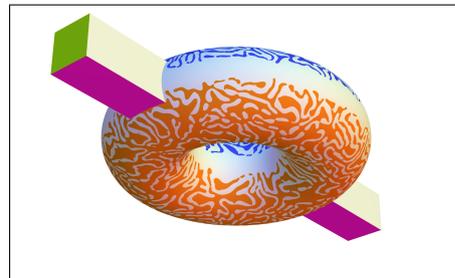
(f)  $i = 245$



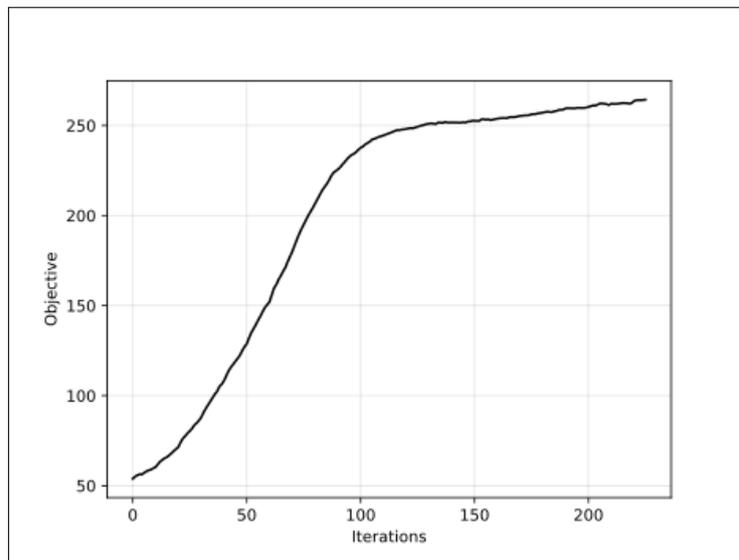
(g)  $i = 245$  (no mesh lines)



(h) Perspective top view of the EMM.



(i) Perspective bottom view of the EMM.



(j) Evolution of the objective criterion  $J(\Omega)$

Figure 4.15: (cont.) Snapshots of the optimization process of the boundary repartition. In figures (g) and (h) the perspective of the design is shown in order to better appreciate the repartition of the boundary. In figure (i), the evolution of the criterion as a function of each iteration.

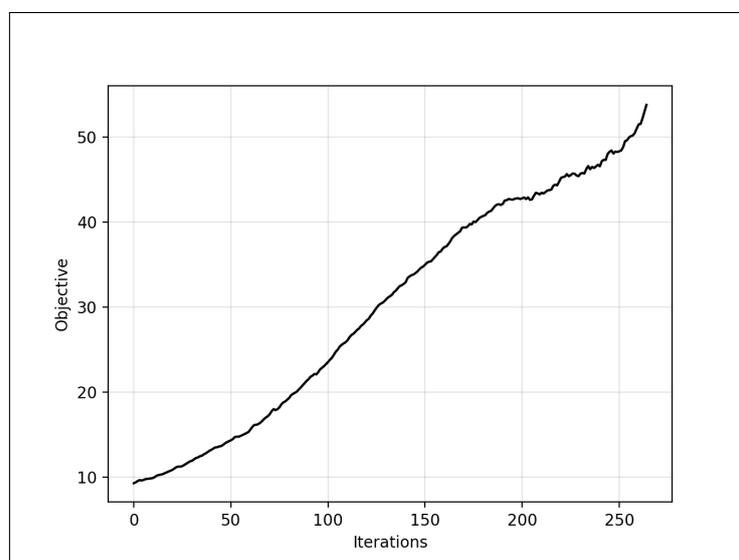


Figure 4.16: Results of the optimization process for the second experiment.

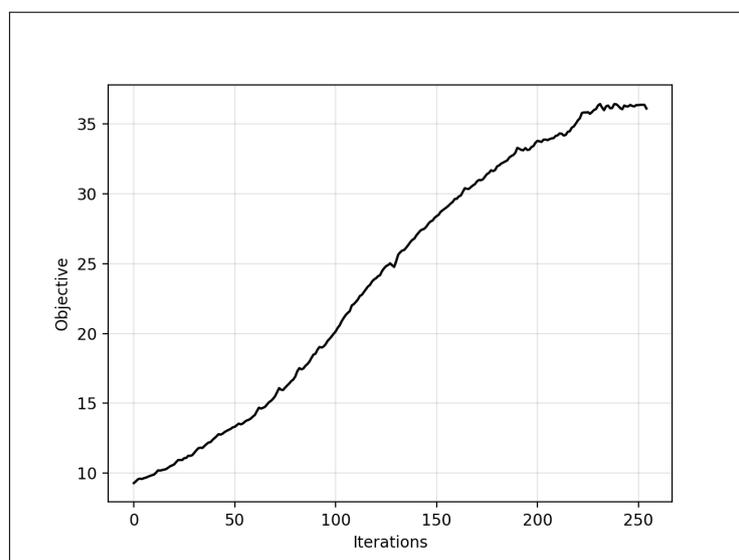
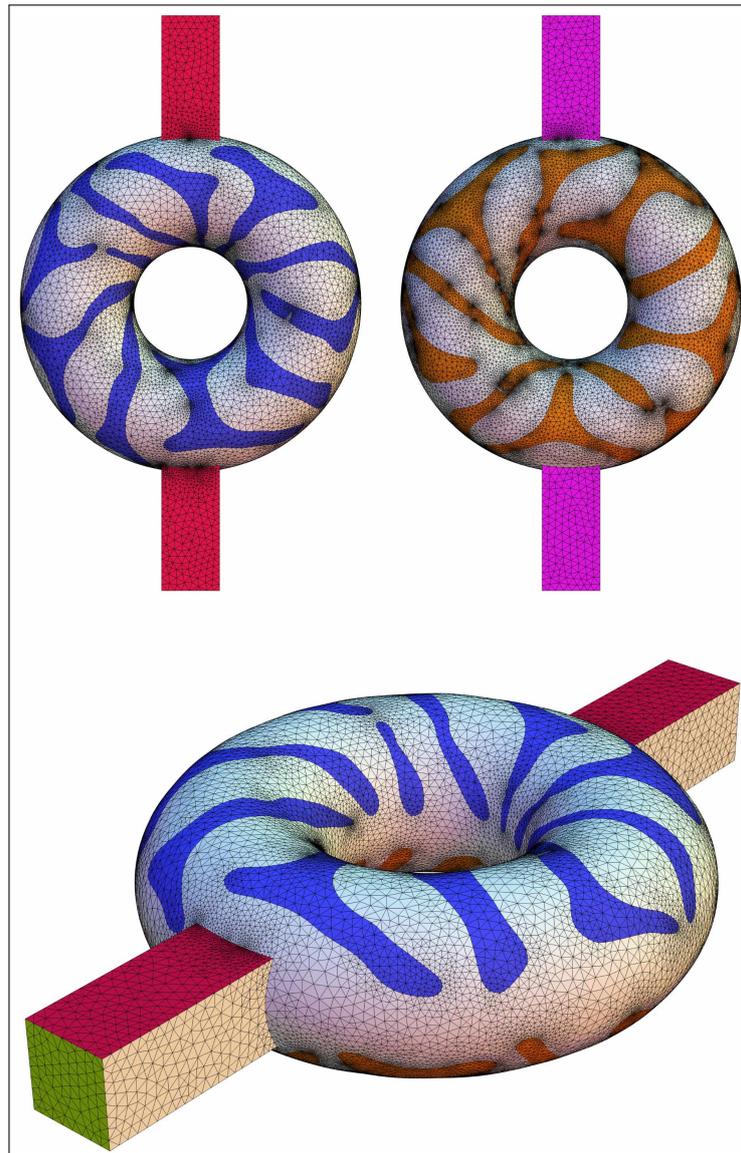


Figure 4.17: Results of the optimization process for the third experiment.

## 4.12 Optimization of sound-soft regions on the surface of a sound-hard obstacle for acoustic cloaking

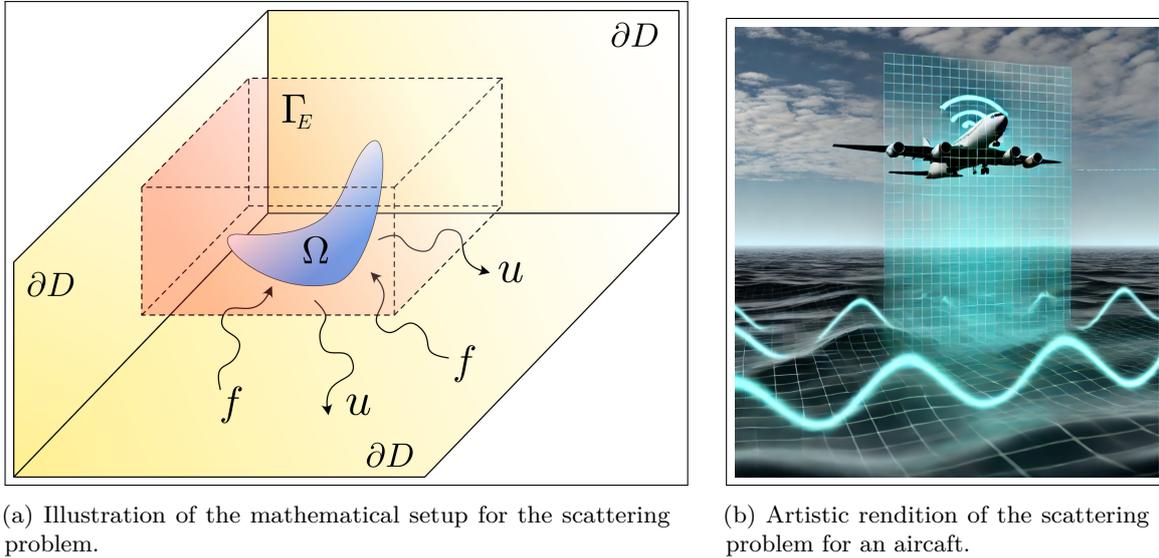


Figure 4.18: In the figure, the obstacle  $\Omega$  is shown immersed within the medium  $D$ . The fictional boundary  $\Gamma$  is where the Sommerfeld radiation condition is imposed, and it depicts the incident wave  $f$  and the scattered wave  $u$ .

In acoustics, the scattering effect occurs when an incident wave (e.g. a sound wave) hits an obstacle or a discontinuity in the propagation medium. This interaction causes the wave to be scattered, i.e. deflected, absorbed, or transmitted in various directions. Depending on the application, this effect is either considered fortunate or undesirable. In particular, it is testament of the presence of an obstacle, which one would rather try to hide, e.g. in military applications. Acoustic cloaking is a technique that creates a shell around an object to manipulate sound waves, allowing them to pass through and around the cloak without scattering, thus rendering both the cloak and the object invisible to acoustic detection. Various types of acoustic cloaking exist, most of which depend on developing metamaterials with specific physical properties, see e.g. the review article [252] about this subject.

Passive acoustic cloaking has received significant attention in recent years, see e.g. [167, 168, 210, 170, 65]. This strategy essentially aims to shrink the object and cloak from the observer's viewpoint, making the object appear vanishingly small. From an optimal design perspective, the preliminary works [331, 265] sought to optimize the parameters of the various acoustic metamaterial layers that compose an acoustic cloak, rather than the structure of the cloak itself. These approaches achieved better results with less complexity compared to the traditional schemes such as those described in [252]. In a physical context similar to ours, recent studies such as [337, 157, 229] have tackled this problem through density optimization, focusing on optimizing the structure surrounding the object for a specific material. These methods typically aim to minimize the norm of the sound pressure of the scattered wave, often using the Helmholtz equation or its variants.

In the present section, we propose to optimizing the constituent material of (the boundary of) the obstacle itself to make it invisible to detection, rather than the surrounding region. To achieve this, we adapt the model presented in [94]. In that study, the authors address a two-dimensional acoustic scattering problem in an inhomogeneous medium, with water as the background fluid and a cloaking region modeled as an inhomogeneous yet isotropic equivalent fluid. Their goal is to minimize the mean squared amplitude of the acoustic scattered pressure field. In the following sections, we present the physical model and specify the objective criterion.

### 4.12.1 The scattering problem

We consider an obstacle  $\Omega \subset \mathbb{R}^n$  with a  $C^2$  boundary  $\partial\Omega$  immersed in a fluid. When the system is subjected to time harmonic waves, the steady-state acoustic pressure  $P(x, t)$  can be expressed as

$\operatorname{Re}(a(x)e^{ikt})$ , where the complex amplitude  $a(x) \in \mathbb{C}$  satisfies the Helmholtz equation in free space for inhomogeneous media:

$$-\nabla \cdot (\gamma(x)\nabla a(x)) = \omega^2 a(x), \quad x \in \mathbb{R}^n, \quad (4.83)$$

where  $\omega \in \mathbb{C}$  is the angular frequency of the wave and  $\gamma \in C^0(\mathbb{R}^n)$  indicates the specific volume of the fluid.

**Remark 4.18.** *In terms of the angular wavenumber, the angular frequency can be decomposed into:*

$$\omega = \frac{2\pi}{\lambda} v$$

where  $\lambda > 0$  denotes the wavelength and  $v$  indicates the phase velocity.

Given an incident wave field  $f$ , we suppose that the pressure field is the superposition of two pressure fields away from the obstacle [94, 25]:

$$a(x) = f(x) + u(x), \quad x \in \mathbb{R}^n \setminus \bar{\Omega} \quad (4.84)$$

where  $f$  is the incident field and  $u$  is the scattered field. The incident field  $f$  is the solution of the Helmholtz equation obtained when no obstacles exist in the medium:

$$-\nabla \cdot (\gamma(x)\nabla f(x)) = \omega^2 f(x), \quad x \in \mathbb{R}^n.$$

To derive an equation for the scattered field, we substitute (4.84) into (4.83) to obtain:

$$-\nabla \cdot (\gamma(x)\nabla u(x)) - \omega^2 u(x) = 0, \quad x \in \mathbb{R}^n \setminus \bar{\Omega}$$

In the numerical setting, it is necessary to approximate the free space using a bounded computational domain. To this end, we consider a rectangular domain  $D \subset \mathbb{R}^3$  that contains the obstacle  $\Omega \subset D$ . This type of approximation requires us to ensure that the scattered wave is outgoing via the Sommerfeld radiation condition:

$$\lim_{r \rightarrow \infty} r \left( \frac{\partial u(x)}{\partial r} - i\omega u(x) \right) = 0, \quad (4.85)$$

where  $r = \|x\|$ . By introducing an artificial boundary  $\Gamma_E$  around the obstacle  $\Omega$ , one can derive approximations for (4.85) using asymptotic expansions of the solution at large distances from the origin. The accuracy of these approximations improves as the distance between the artificial boundary  $\Gamma$  and the obstacle  $\Omega$  increases. Therefore, achieving higher accuracy requires enlarging the computational domain, which results in increased computational cost. In our case, we utilize a first order approximation as presented in [295]:

$$\gamma \frac{\partial u}{\partial n_{\Gamma_E}} - \left( i\omega - \frac{1}{R} \right) u = 0 \text{ on } \Gamma_E, \quad (4.86)$$

where  $n_{\Gamma_E}$  denotes the normal vector to  $\Gamma$ , and  $R$  is the distance between  $\Gamma_E$  and  $\partial D$ . In our work, we consider that the boundary of the obstacle  $\partial\Omega$  is decomposed into  $\partial\Omega = \Gamma_R \cup \Gamma$ , such that:

- The region  $\Gamma$  consists of sound-hard material, supporting an inhomogeneous Neumann condition:

$$\gamma \frac{\partial u}{\partial n_{\partial\Omega}} = -\gamma \frac{\partial f}{\partial n_{\partial\Omega}}, \quad x \in \Gamma,$$

which means that the incident wave is entirely reflected on this region.

- The region  $\Gamma_R$  consists of the sound-soft material which supports a Robin boundary condition:

$$\gamma \frac{\partial u}{\partial n_{\partial\Omega}} + \frac{i\omega}{z} u = -\gamma \frac{\partial f}{\partial n_{\partial\Omega}} - \frac{i\omega}{z} f, \quad x \in \Gamma_R,$$

which means that the wave is partially absorbed on this region, according to the acoustic impedance  $z > 0$ .

Collecting all these conditions, we can characterize the scattered field as the unique, complex valued, solution  $u_{\Gamma_R} \in H^1(D \setminus \bar{\Omega}; \mathbb{C})$  to the boundary value problem:

$$\begin{cases} -\nabla \cdot (\gamma \nabla u_{\Gamma_R}) - \omega^2 u_{\Gamma_R} = 0 & \text{in } D \setminus \bar{\Omega} \\ \gamma \frac{\partial u_{\Gamma_R}}{\partial n_{\partial\Omega}} = -\gamma \frac{\partial f}{\partial n_{\partial\Omega}} & \text{on } \Gamma, \\ \gamma \frac{\partial u_{\Gamma_R}}{\partial n_{\partial\Omega}} + \frac{i\omega}{z} u_{\Gamma_R} = -\gamma \frac{\partial f}{\partial n_{\partial\Omega}} - \frac{i\omega}{z} f & \text{on } \Gamma_R, \\ \gamma \frac{\partial u_{\Gamma_R}}{\partial n_{\Gamma}} - \left(i\omega - \frac{1}{R}\right) u_{\Gamma_R} = 0 & \text{on } \Gamma_E, \end{cases} \quad (4.87)$$

where we have employed the subscript  $\Gamma_R$  to indicated the dependence on the sound-soft boundary. See Fig. 4.18 for an illustration of the setup. Finally, let us remark that the incident field  $f$  and the acoustic impedance  $z$  are given as the data of the problem. The weak formulation for the solution  $u_{\Gamma_R} \in H^1(D \setminus \bar{\Omega}; \mathbb{C})$  to (4.87) is:

$$\begin{aligned} \forall v \in H^1(D \setminus \bar{\Omega}; \mathbb{C}), \\ \int_{D \setminus \bar{\Omega}} \gamma \nabla u_{\Gamma_R} \cdot \bar{\nabla} v \, dx - \omega^2 \int_{D \setminus \bar{\Omega}} u_{\Gamma_R} \bar{v} \, dx - \left(i\omega - \frac{1}{R}\right) \int_{\Gamma_E} u_{\Gamma_R} \bar{v} \, ds - \frac{i\omega}{z} \int_{\Gamma_R} u_{\Gamma_R} \bar{v} \, ds = \\ \frac{i\omega}{z} \int_{\Gamma_R} f \bar{v} \, ds + \int_{\partial\Omega} \gamma \frac{\partial f}{\partial n_{\partial\Omega}} \bar{v} \, ds, \end{aligned} \quad (4.88)$$

where  $\bar{v}$  is the conjugate of  $v$ , and  $\bar{\nabla} v$  is the conjugate of its gradient. Note that in this case, the normal vector to the boundary of  $\Omega$  is the negative of the normal to the boundary of  $D \setminus \bar{\Omega}$ , i.e.:

$$n_{\partial\Omega}(x) = -n_{\partial(D \setminus \bar{\Omega})}(x), \quad x \in \partial\Omega.$$

See Fig. 4.19 for an illustration.

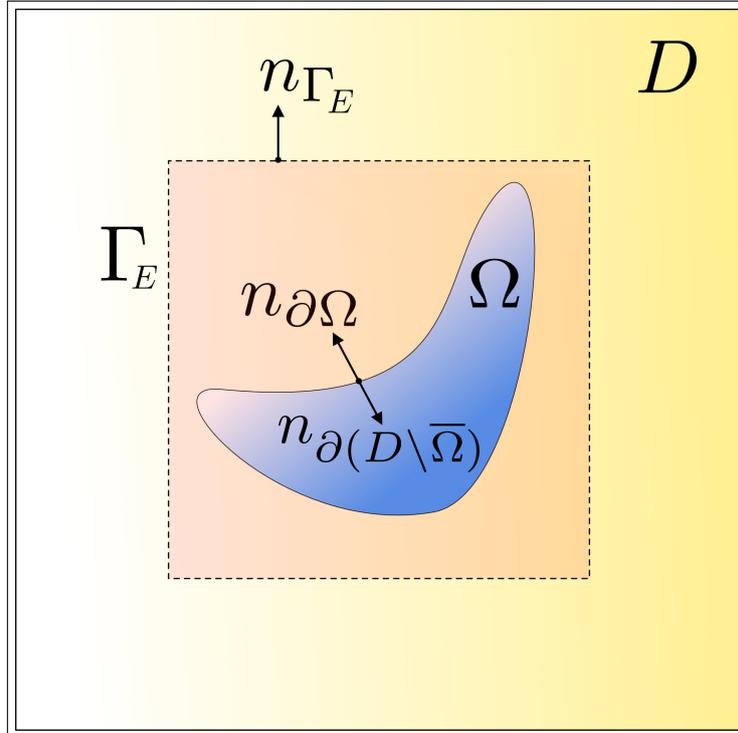


Figure 4.19: Illustration of the boundary normal setup for the scattering problem.

**Remark 4.19.** *One particular fact about (4.88), is that despite its ellipticity, the Lax-Milgram lemma cannot be used to establish well-posedness since the sesquilinear form is not coercive. The appropriate approach to address the Helmholtz problem is to apply the so-called Banach–Nečas–Babuška theorem (see Chapters 25 and 35 in [137]). In this case the imposition of the Robin boundary conditions in (4.87) yields the well-posedness which is essentially due to the non zero imaginary part in the condition.*

### 4.12.2 The optimization problem

We aim to tailor the optimization problem presented in [94] to suit our specific context. As discussed in that study, and illustrated by the decomposition in (4.84), the obstacle can be effectively cloaked by minimizing the squared amplitude of the scattered field. For a given obstacle  $\Omega$ , we would like to replace the sound-hard material with the sound-soft material on the surface  $\partial\Omega$ . Hence, the optimization problem is:

$$\min_{\Gamma_R \subset \partial\Omega} J(\Gamma_R), \quad (4.89)$$

where

$$J(\Gamma_R) = \frac{1}{2\text{Vol}(\Omega)} \int_{D \setminus \bar{\Omega}} |u_{\Gamma_R}|^2 dx \quad (4.90)$$

$u_{\Gamma_R}$  is the scattered pressure field solution to (4.87) and  $|u|^2 := \bar{u}u$ ,  $u \in \mathbb{C}$ .

### 4.12.3 The shape derivative

In this section, we state the shape derivative of (4.89). The proof can be found in Appendix A.2.2.

**Proposition 4.6.** *The criterion  $J(\Gamma_R)$  is shape differentiable at  $\theta = 0$  (with  $\theta \cdot n_{\partial\Omega} = 0$ ), and its shape derivative reads:*

$$J'(\Gamma_R)(\theta) = -\text{Im} \left( \int_{\partial\Gamma_R} \frac{\omega}{z} \overline{(u_{\Gamma_R} + f)} p_{\Gamma_R} \theta \cdot n_{\partial\Gamma_R} d\sigma \right),$$

where the adjoint state  $p_{\Gamma_R} \in H^1(\Omega; \mathbb{C})$  is the unique solution to the following boundary value problem:

$$\left\{ \begin{array}{ll} -\nabla \cdot (\gamma \nabla p_{\Gamma_R}) - \omega^2 p_{\Gamma_R} = -\frac{u_{\Gamma_R}}{\text{Vol}(\Omega)} & \text{in } D \setminus \bar{\Omega} \\ \gamma \frac{\partial p_{\Gamma_R}}{\partial n_{\partial\Omega}} = 0 & \text{on } \Gamma, \\ \gamma \frac{\partial p_{\Gamma_R}}{\partial n_{\partial\Omega}} - \frac{i\omega}{z} p_{\Gamma_R} = 0 & \text{on } \Gamma_R, \\ \gamma \frac{\partial p_{\Gamma_R}}{\partial n_{\Gamma}} - \left( -i\omega - \frac{1}{R} \right) p_{\Gamma_R} = 0 & \text{on } \Gamma_E, \end{array} \right.$$

### 4.12.4 The topological derivative

It is important to note that our setting differs slightly from that of Section 4.7, with the key difference being the absence of a homogeneous Dirichlet region and the Robin boundary conditions. However, this variation does not pose a problem, as the proof is analogous. Consequently, we can still derive an expression for the topological sensitivity when introducing a sound-soft region into a sound-hard region.

**Proposition 4.7.** *The perturbed criterion  $J(G_{x_0,\varepsilon})$  has the following asymptotic expansion in terms of  $\varepsilon$ :*

$$J(G_{x_0,\varepsilon}) = J(G) + \pi\varepsilon^2 k \operatorname{Im} \left( \overline{u_0(x_0) + f(x_0)} p_0(x_0) \right) + o(\varepsilon^2) \text{ if } d = 3,$$

where the adjoint state  $p_{\Gamma_R} \in H^1(\Omega; \mathbb{C})$  is the unique solution to the following boundary value problem:

$$\left\{ \begin{array}{ll} -\nabla \cdot (\gamma \nabla p_{\Gamma_R}) - \omega^2 p_{\Gamma_R} = -\frac{u_{\Gamma_R}}{\operatorname{Vol}(\Omega)} & \text{in } D \setminus \bar{\Omega} \\ \gamma \frac{\partial p_{\Gamma_R}}{\partial n_{\partial\Omega}} = 0 & \text{on } \Gamma, \\ \gamma \frac{\partial p_{\Gamma_R}}{\partial n_{\partial\Omega}} - \frac{i\omega}{z} p_{\Gamma_R} = 0 & \text{on } \Gamma_R, \\ \gamma \frac{\partial p_{\Gamma_R}}{\partial n_{\Gamma}} - \left( -i\omega - \frac{1}{R} \right) p_{\Gamma_R} = 0 & \text{on } \Gamma_E, \end{array} \right.$$

#### 4.12.5 Experiment setup

We consider the minimization of the norm of pressure field while also minimizing occupied surface area:

$$\min_{\Gamma_R \subset \partial\Omega} \frac{1}{2\operatorname{Vol}(\Omega)} \int_{D \setminus \bar{\Omega}} |u_{\Gamma_R}|^2 dx + \ell \operatorname{Area}(\Gamma_R), \quad (4.91)$$

where  $\ell > 0$  is a penalization parameter. We assume that the computational domain  $D$  has an associated tetrahedral mesh  $\mathcal{T}$  of the computational domain  $D$ , such that it contains a submesh  $\mathcal{K}$  representing the obstacle  $\Omega$ . For our examples, we have settled on the model of an aircraft, depicted in Fig. 4.21. In our experiments, the computational domain, is a box containing an inner box, which itself contains the aircraft. This is depicted in Fig. 4.22. For simplicity, we have assumed that  $\gamma = 1$ , everywhere, so that the incident wave is given by the plane wave function:

$$\forall x \in \mathbb{R}^d, \quad f(x) = e^{i\omega\xi \cdot x}$$

travelling vertically in the direction  $\xi = (0, 0, 1) \in \mathbb{R}^n$ . The computational mesh consists of 334k vertices, with 1.97m tetrahedra. The aircraft itself consists of 464k tetrahedra and 93k vertices. To solve the problem, we construct a finite element space on the mesh  $\mathcal{T} \setminus \mathcal{K}$ . The optimization process starts with 6 iterations of topological changes. After the 10th iteration, we continue applying topological updates every 10 iterations, up to iteration 100. Between the topological iterations, we perform geometrical optimization, and let the algorithm run for 142 iterations. The parameters chosen are detailed in Section 4.12.5.

Parameter	Value
$\ell$	$1e-7$
$\alpha$	4
$\omega$	20
$h_{max}$	$\omega/3$
$h_{min}$	$\omega/32$

Table 4.2: Values of the parameters used in the optimization process.

#### 4.12.6 Analysis of results

The snapshots of the evolution process are shown in Fig. 4.23, and the convergence history is depicted in Fig. 4.20. The final design is illustrated in Figs. 4.24 to 4.26. The whole iterative process took a total of approximately 12.5h for 142 iterations. The first snapshot ( $i = 6$ ) displays the design after introducing 6 sound-soft zones. The topological derivative seems to prioritize the wings and the center of the aircraft. By iteration 10, the two central holes have vanished, and the holes near the wings are modified through geometrical optimization. This process continues, with no new areas forming, as geometrical optimization appears to suppress the nucleation of new zones. The design gradually becomes homogenized, similar to the outcome in Section 4.11, where new regions are formed by the geometrical optimization “cutting off regions” rather than inserting new ones via the topological step.

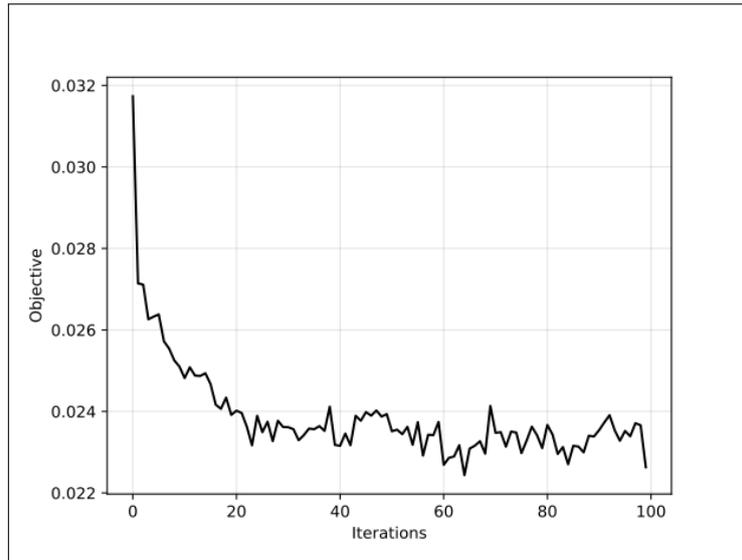


Figure 4.20: Convergence history of the criterion.

The primary focus of the final design is clear: the wings are the areas that require prioritization to effectively conceal the aircraft. While this may not be immediately apparent, a closer analysis shows that the wings are positioned perpendicular to the direction  $\xi = (0, 0, 1)$ , which causes them to reflect the wave more strongly and generate the highest pressure zones. This is evidenced by examining the magnitude of the scattered pressure field around these regions, as shown in Fig. 4.26. From these figures, it becomes evident that the wings, along with the tail, should be the primary focus. In the final design, the pressure field around the wings has been diffused through the application of a Robin boundary, making the wings appear "less visible." Additionally, the plots suggest that the tail should be the next priority for optimization. However, we believe that no material was added to the tail due to area penalization, which placed greater emphasis on the wings.

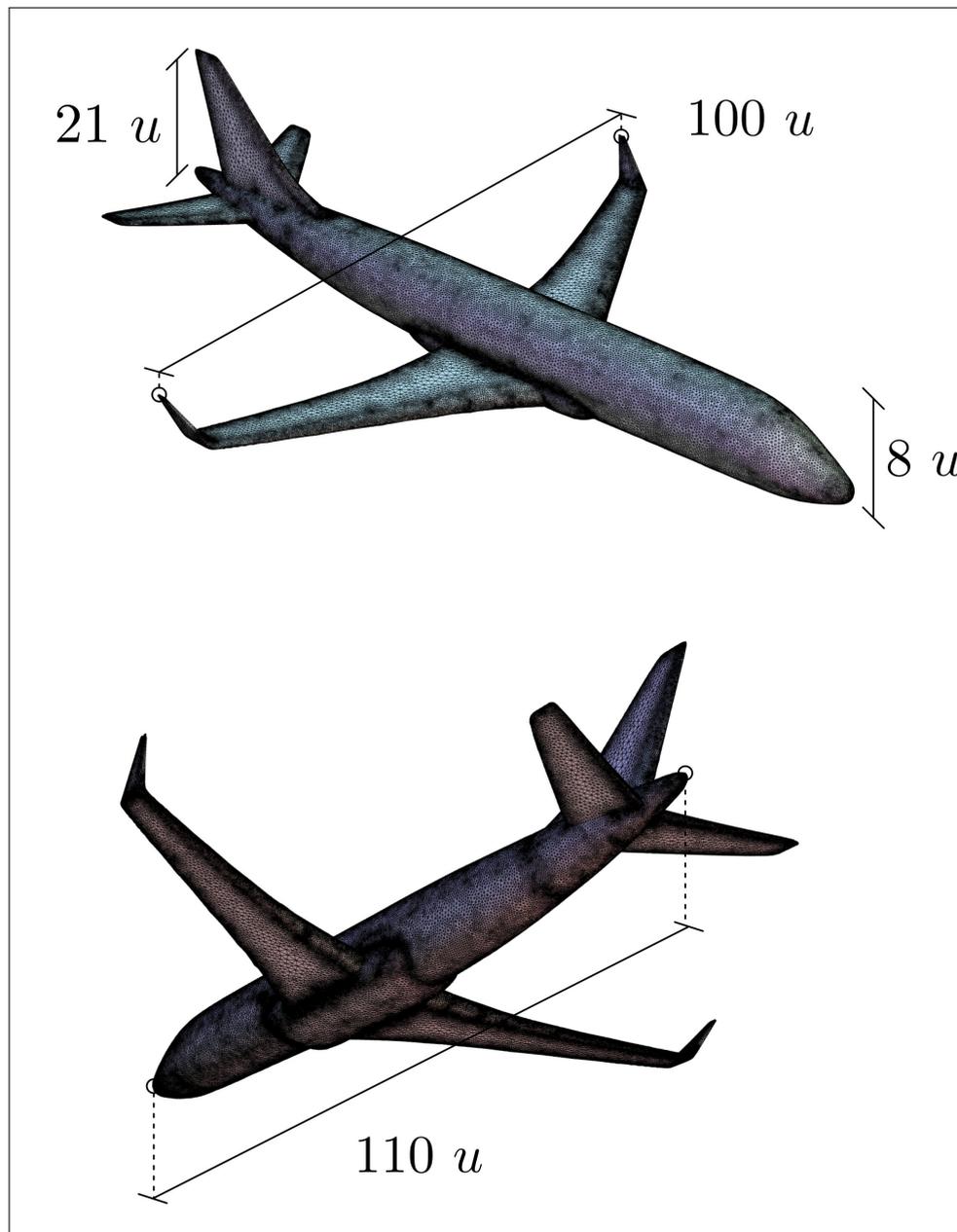


Figure 4.21: Different view angles of the tetrahedral mesh  $\mathcal{T}$  which serves as the obstacle in the medium  $D$ . The wingspan of the aircraft is 100 units wide, the rear wing is 21 units high, the cockpit is 8 units high and the aircraft itself is 110 units long. The boundary of the obstacle consists of only sound-hard material.

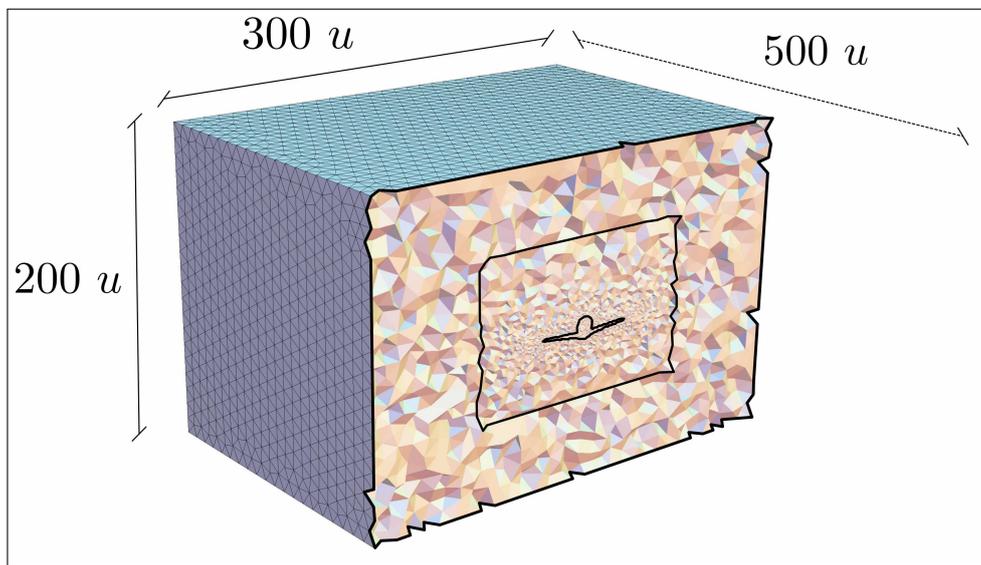
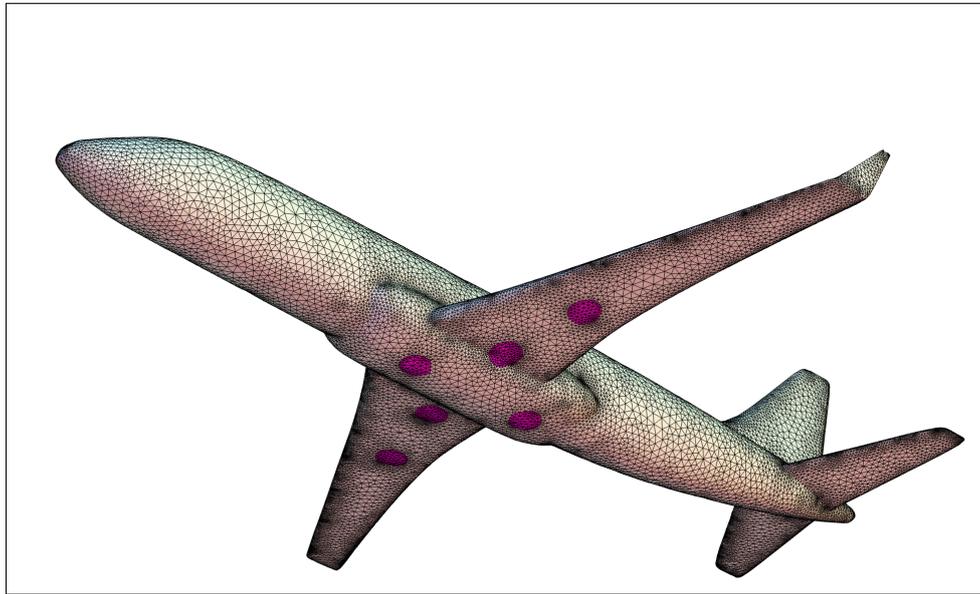
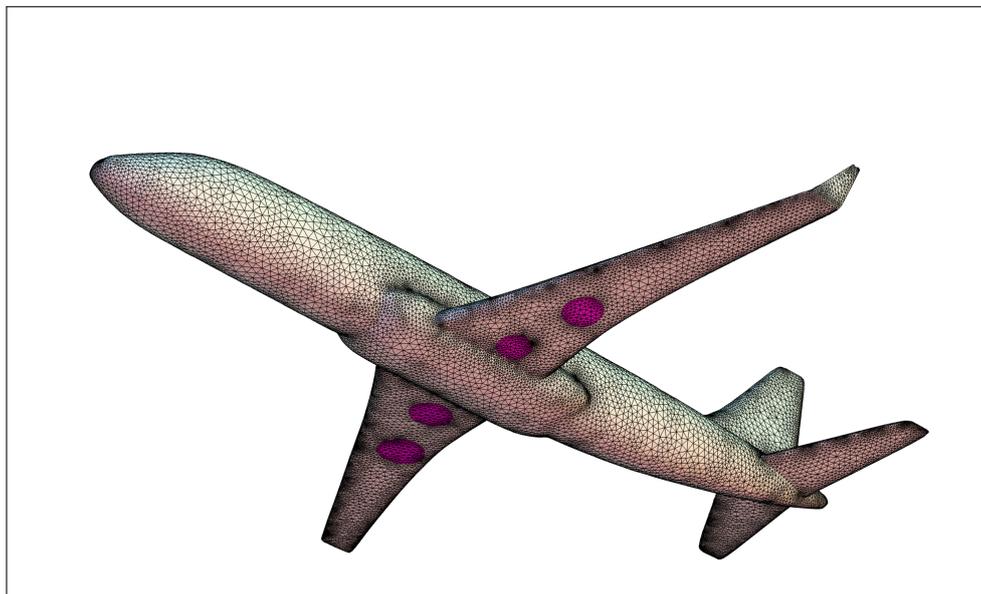


Figure 4.22: Different view angles of the tetrahedral mesh  $\mathcal{T}$  which serves as the computational domain  $D$ . The outermost layer  $\partial D$ , is the boundary of the domain. The middle layer acts as the artificial boundary  $\Gamma_E$ , where the Sommerfield radiation condition is imposed. The innermost layer is just the boundary  $\partial\Omega = \Gamma \cup \Gamma_R$  of the obstacle  $\Omega$ . The outer box is 200 units high, 300 wide, and 500 units long. The inner box is 100 units high, 150 units wide, and 250 units long.

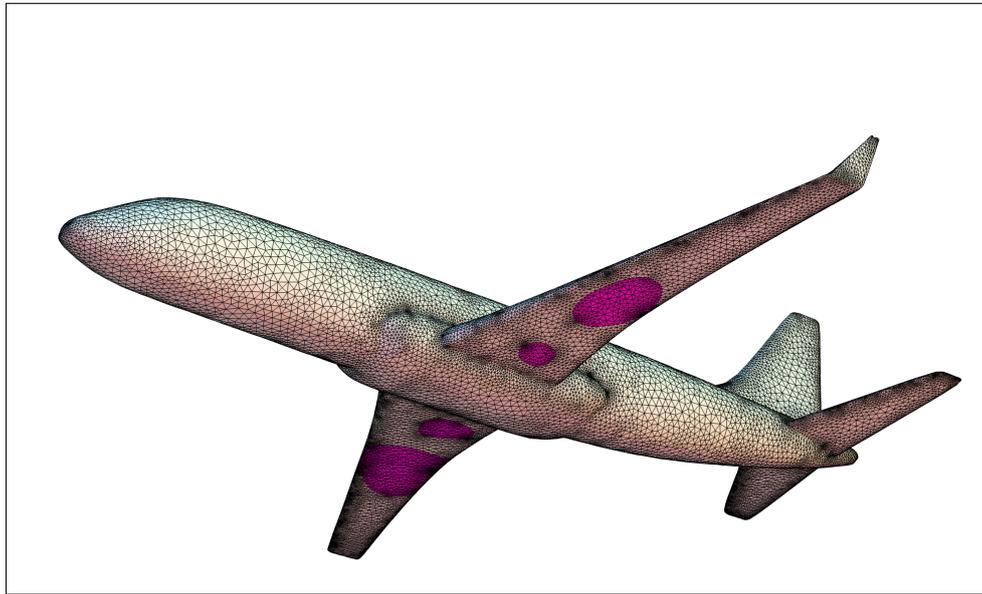


(a)  $i = 6$

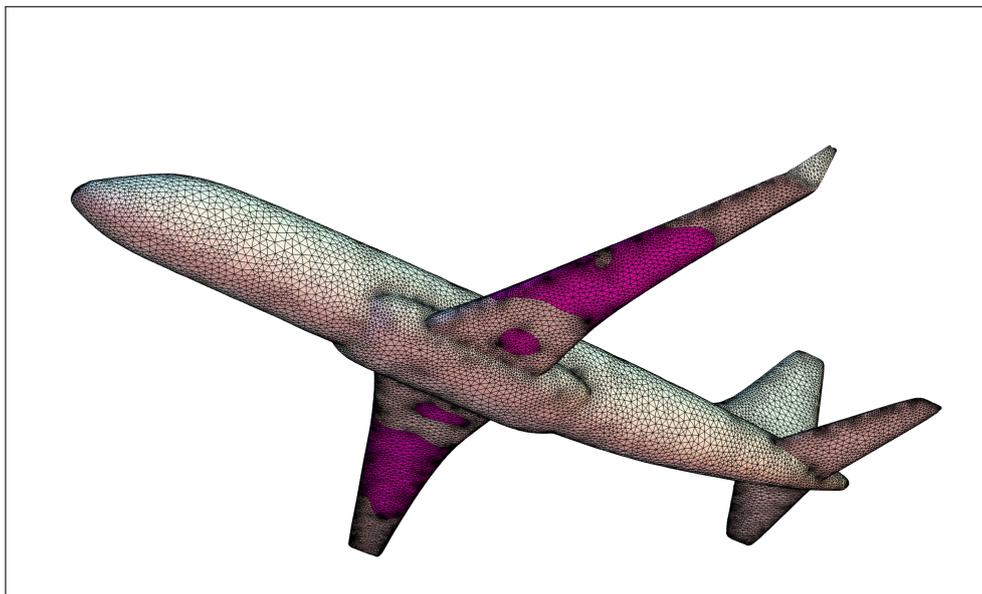


(b)  $i = 10$

Figure 4.23: Snapshots of the optimization process of the sound-soft optimization process. Here  $i$  indicates the number of iterations that have passed. The pink color corresponds to the sound-soft boundary, while the rest corresponds to the sound-hard boundary.

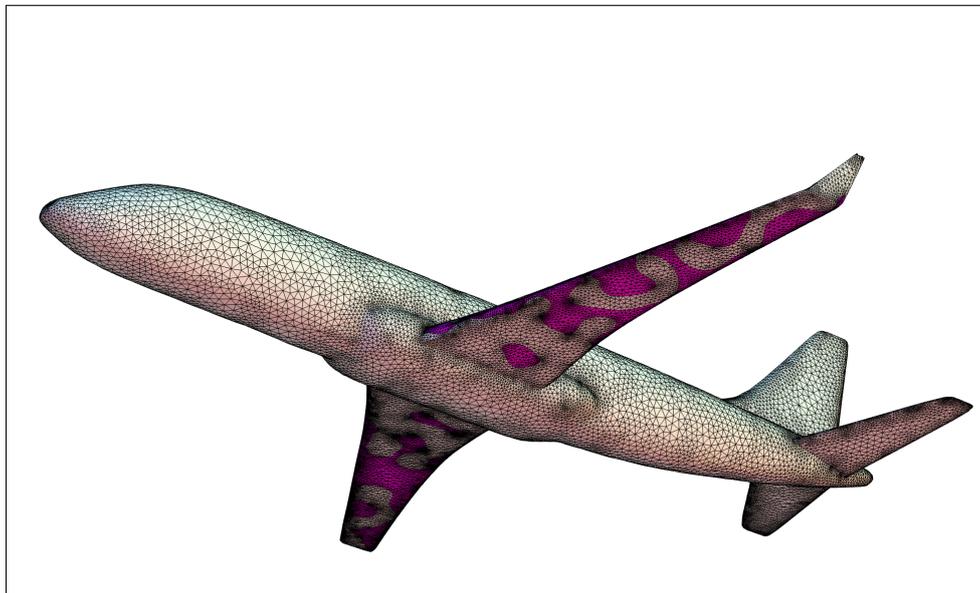


(c)  $i = 20$

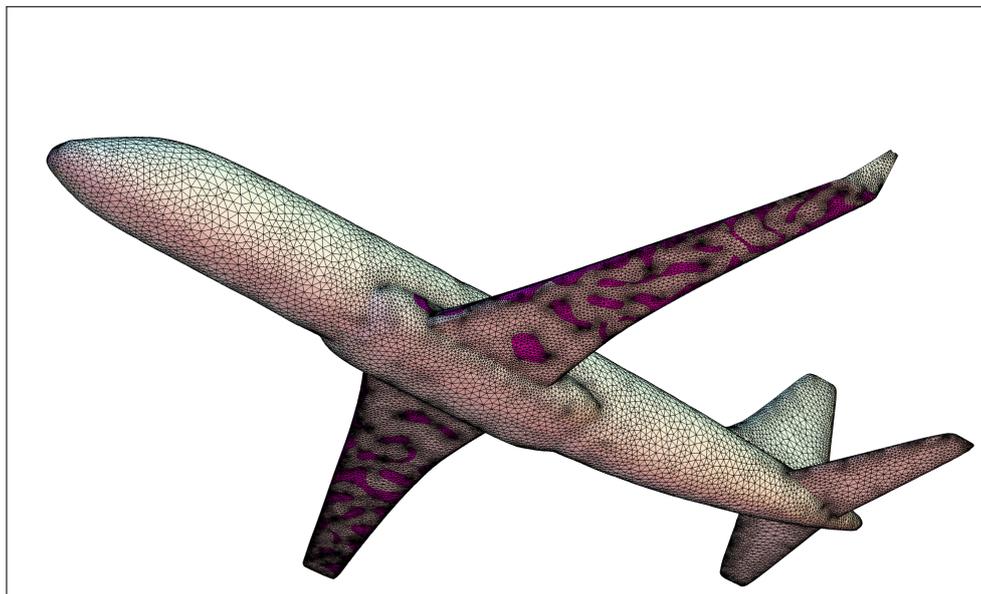


(d)  $i = 40$

Figure 4.23: Snapshots of the optimization process of the sound-soft optimization process. Here  $i$  indicates the number of iterations that have passed. The pink color corresponds to the sound-soft boundary, while the rest corresponds to the sound-hard boundary.



(e)  $i = 80$



(f)  $i = 120$

Figure 4.23: Snapshots of the optimization process of the sound-soft optimization process. Here  $i$  indicates the number of iterations that have passed. The pink color corresponds to the sound-soft boundary, while the rest corresponds to the sound-hard boundary.

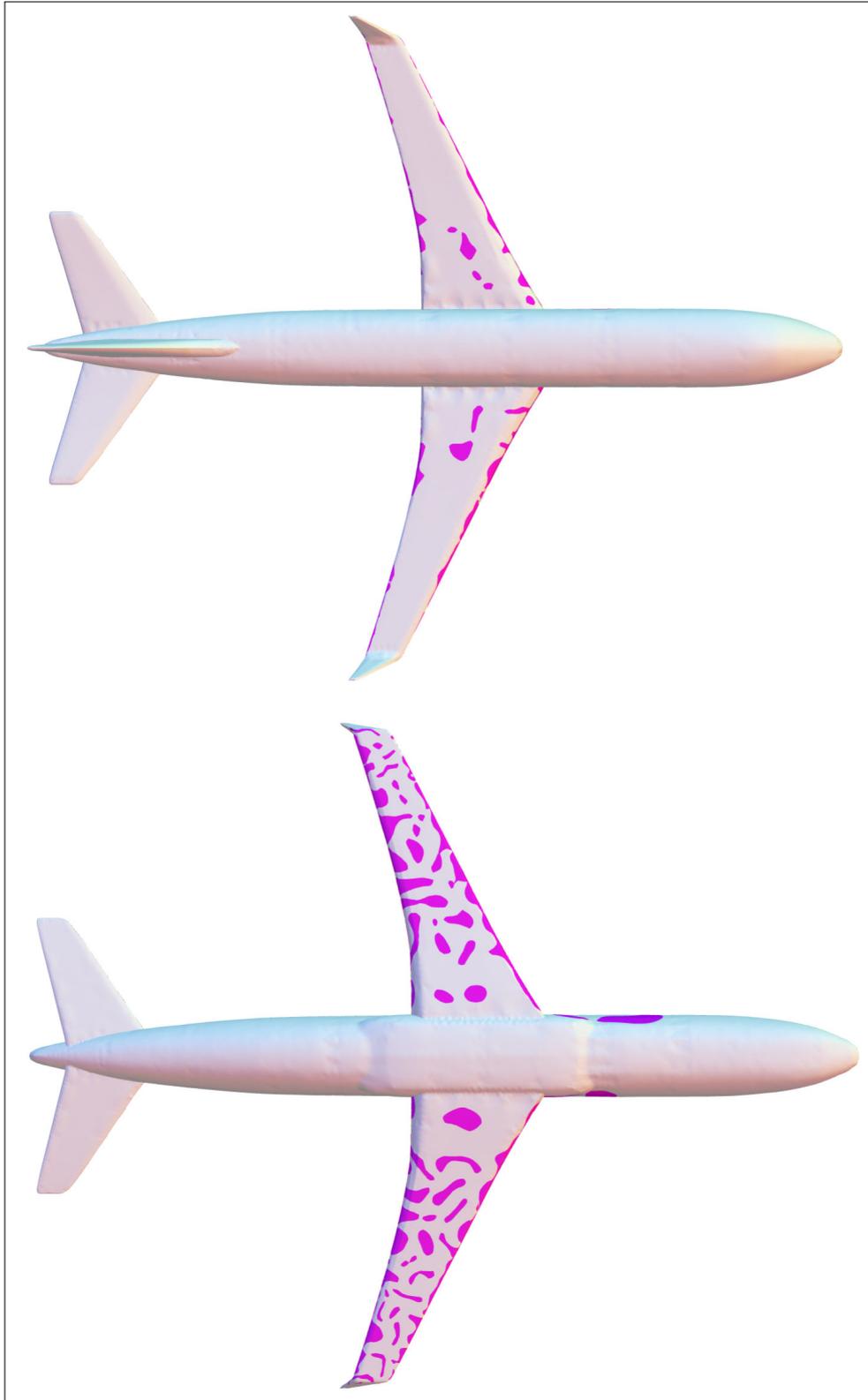


Figure 4.24: Top and bottom views of the final result ( $i = 142$ ) of the experiment in [Section 4.12](#).

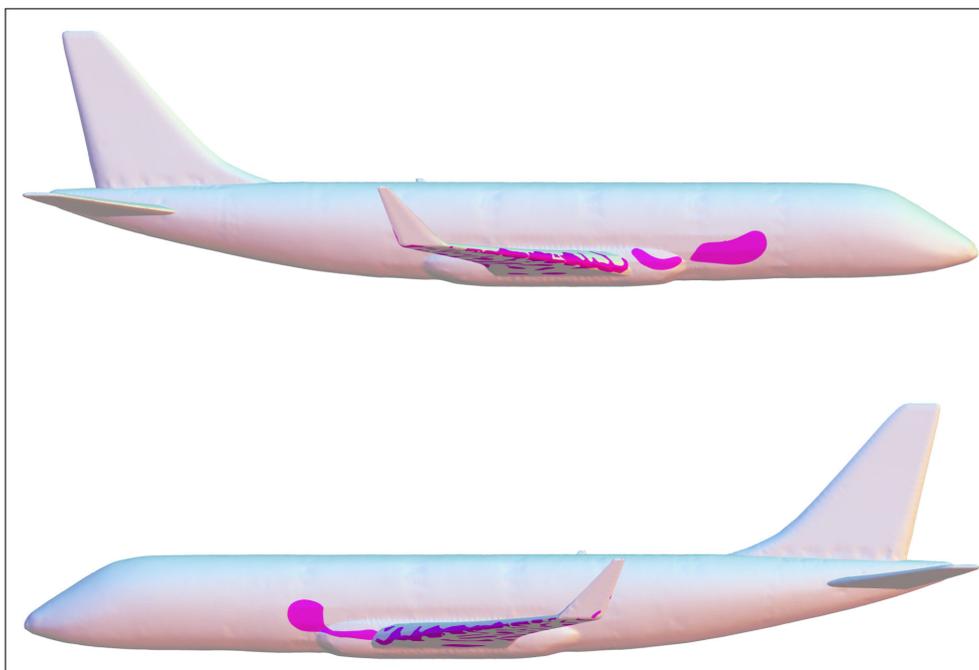


Figure 4.25: Side views of the final result ( $i = 142$ ) of the experiment in [Section 4.12](#).

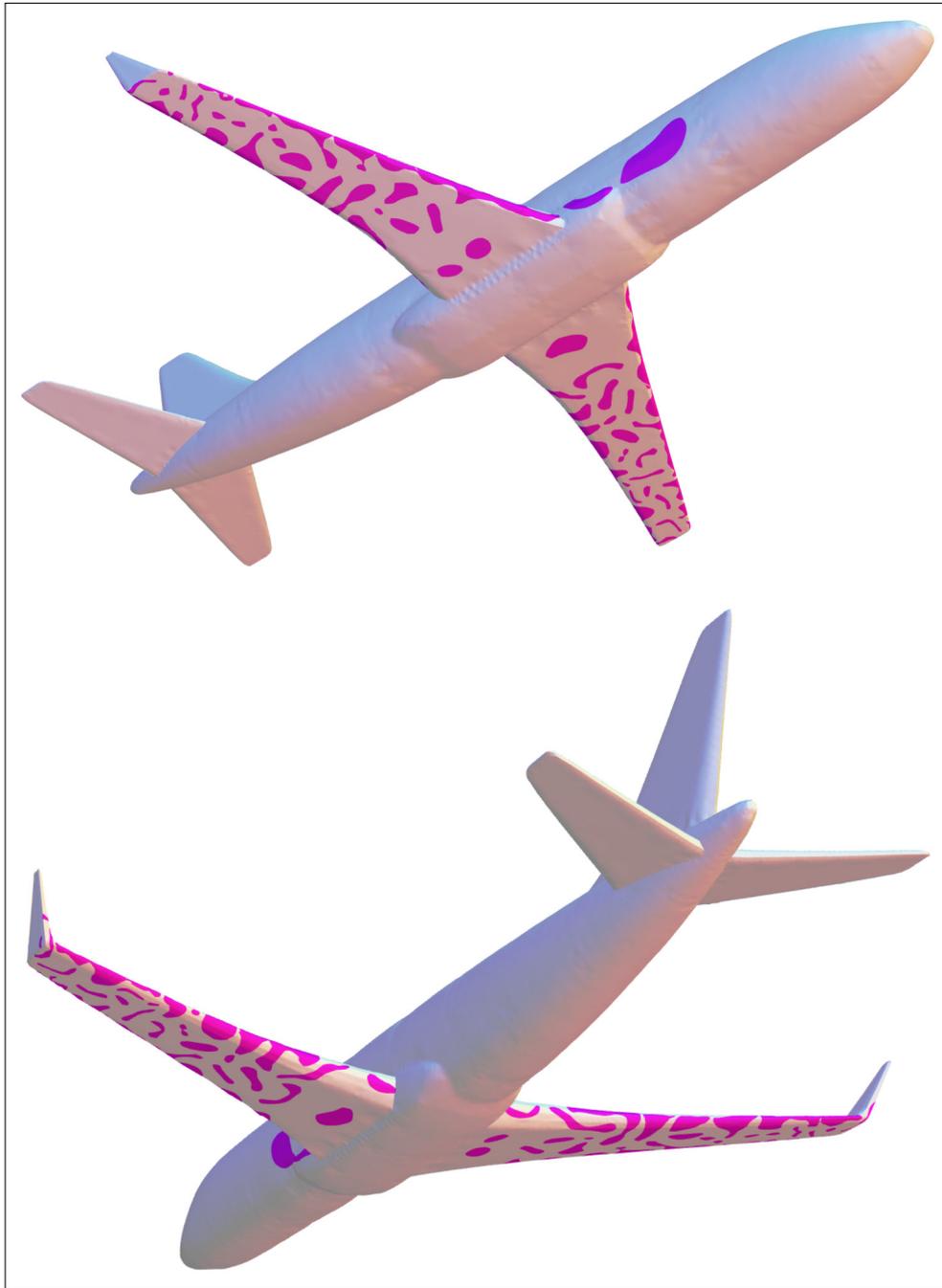
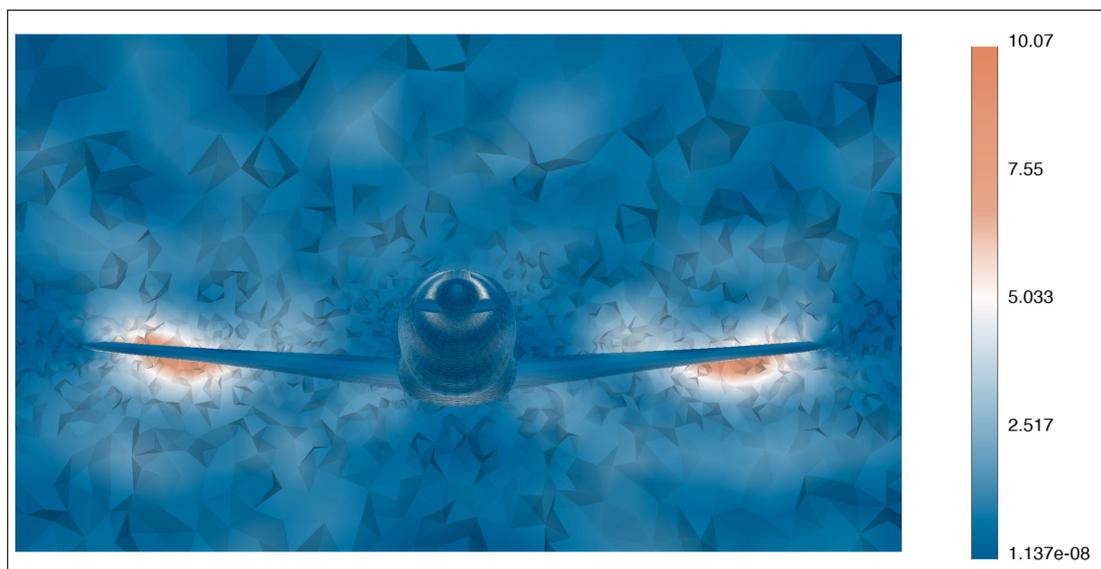
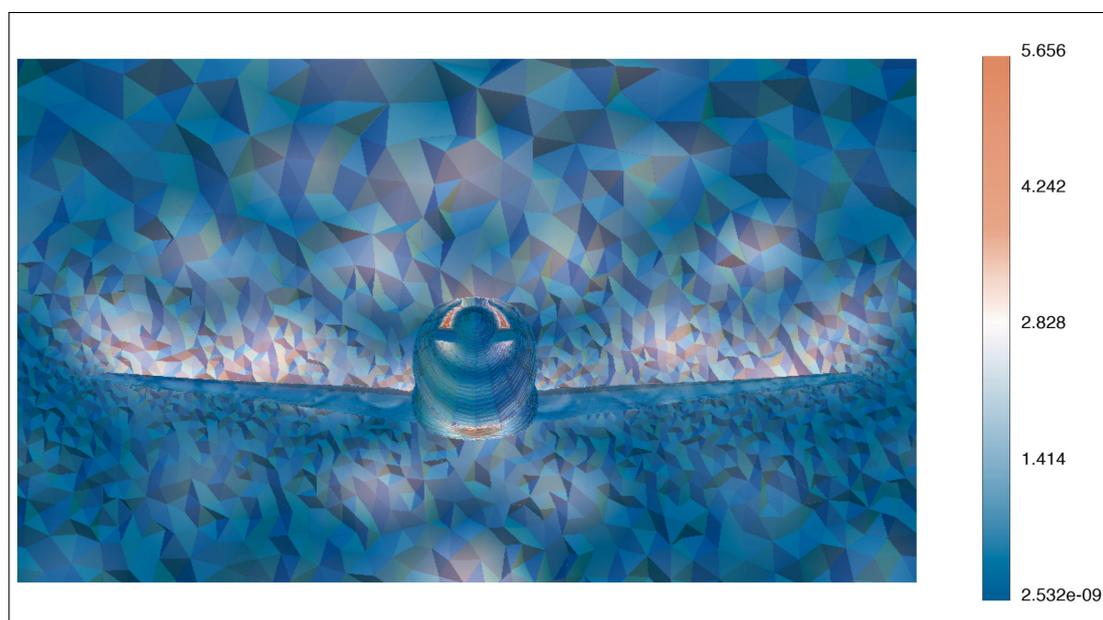


Figure 4.26: Perspective views of the final result ( $i = 142$ ) of the experiment in [Section 4.12](#).

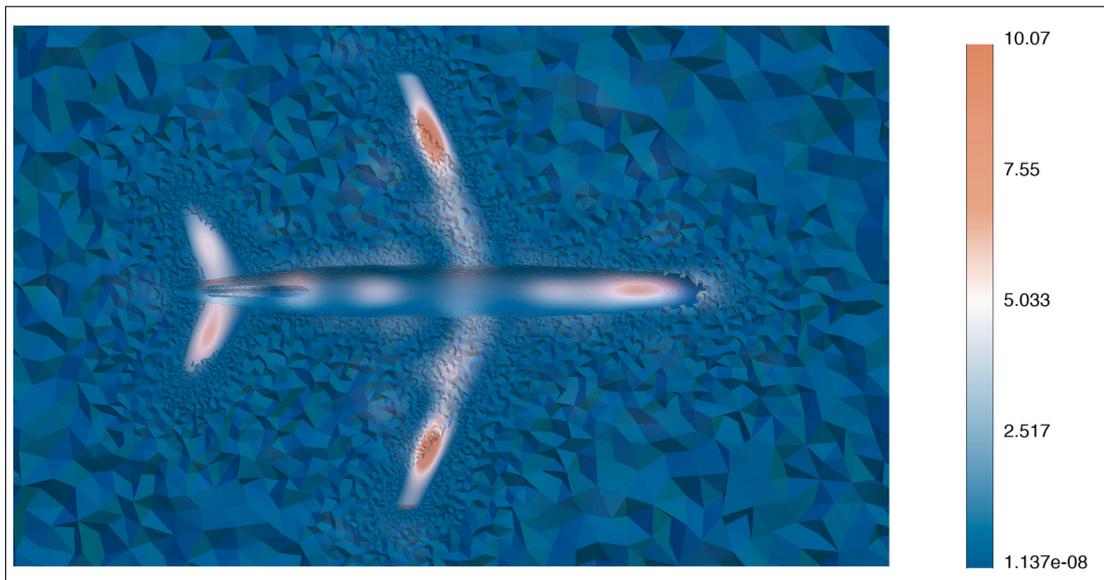


(a) Magnitude of the scattered pressure field, with only a sound-hard boundary.

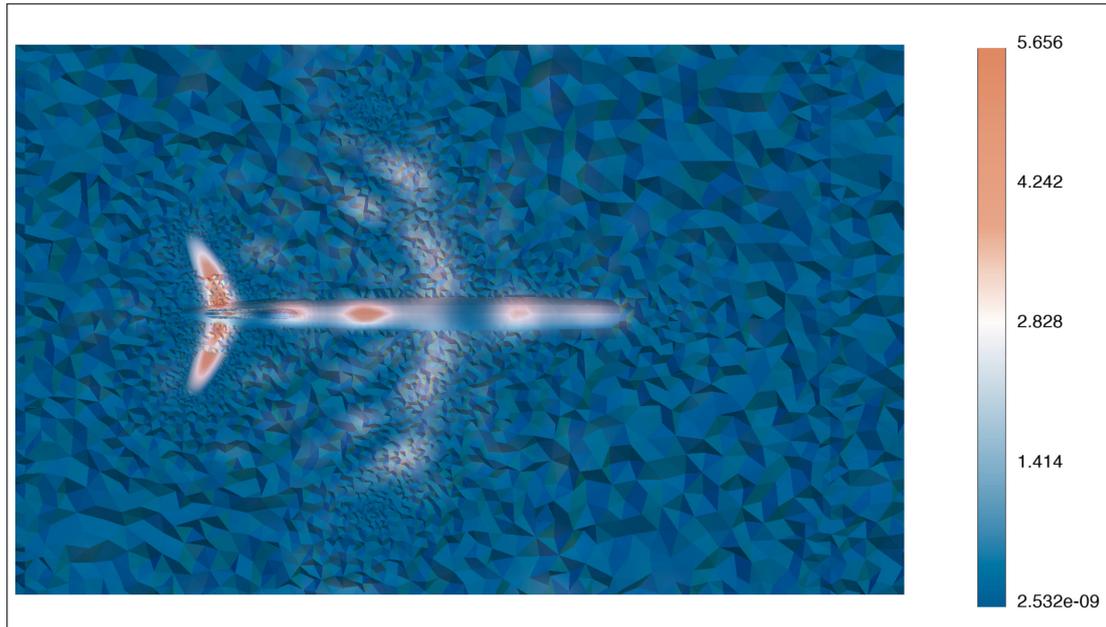


(b)  $i = 120$

Figure 4.26: Magnitude of the scattered pressure field, with the repartition of sound-soft boundary found by the algorithm.



(c) Magnitude of the scattered pressure field, with only a sound-hard boundary.



(d) Magnitude of the scattered pressure field, with the repartition of sound-soft boundary found by the algorithm.

Figure 4.26: Bottom cross-section view of the magnitude of the scattered pressure field  $u_{\Gamma_R}$ .

### 4.13 Optimization of the repartition of structural supports for a water tank

Structural optimization involves a suite of computational techniques designed to enhance the performance, efficiency, and cost-effectiveness of structures by systematically modifying their geometry, material properties, or layout. This optimization can be achieved through various methods, including structural and sensitivity analyses, computer-aided geometric design, mathematical optimization, and interactive graphics.

Shape optimization, a key aspect of structural optimization, refines the geometry of structures to improve load distribution, reduce stress concentrations, and enhance overall durability. This approach also boosts aerodynamic or hydrodynamic performance, making it essential for applications in aerospace [353, 215], automotive [286, 336], and marine industries [342]. Additionally, shape optimization leads to significant weight reduction by eliminating unnecessary material, thus saving costs and increasing efficiency. This lightweight design is particularly advantageous in industries where fuel savings and increased payload capacity are crucial. Furthermore, it fosters innovation and sustainability by enabling creative, resource-efficient designs that reduce environmental impact [34]. For a technical overview of shape and topology optimization techniques applied to structural optimization, we refer the reader to [15], and to [274] for a qualitative comparison of various methods applied to structural optimization.

In this section, we analyze the case of optimizing the supporting regions of a water tank structure, as illustrated in Fig. 4.27. Our goal is to minimize the total displacement of the water tank, thereby enhancing its ability to handle forces and reducing the risk of structural failure. We consider that certain regions of the tank's boundary can be fixed, experiencing no displacement, and our aim is to optimize the placement of these fixed regions. In practice, these fixed regions correspond to where the tank is riveted or attached to a concrete structure.

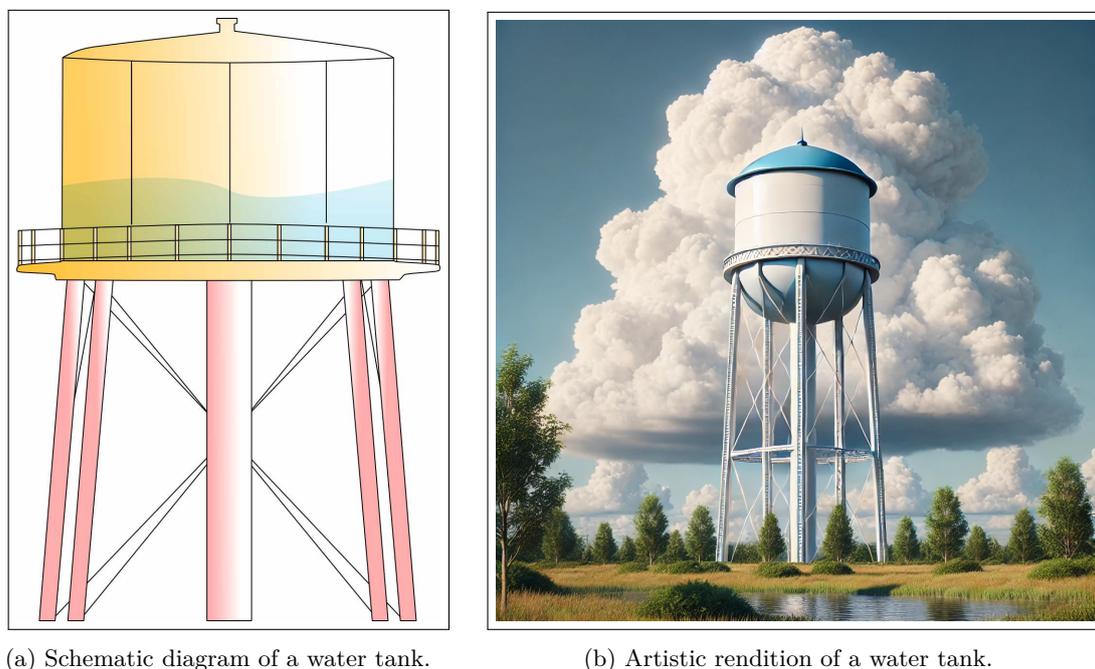


Figure 4.27: A supported water tank typically features a cylindrical body made of materials like steel, concrete, or polyethylene, with a flat or slightly curved bottom, and a domed or flat top equipped with an access hatch or manway. The tank is elevated on legs made of, typically made steel or reinforced concrete, whose number and thickness depend on the tank's size and load requirements. To enhance stability, cross braces may connect the legs, forming a lattice structure, and the legs are anchored to a solid base plate or foundation to distribute the load evenly and prevent sinking or tilting.

### 4.13.1 The optimization problem

We consider that the water tank is represented by a bounded Lipschitz domain  $\Omega \subset \mathbb{R}^3$ . The boundary  $\partial\Omega$  is decomposed into the following parts:

- The region  $\Gamma_D$  is that where the water tank is fixed;
- The region  $\Gamma_F$  is the upper part of the tank; this traction-free region is non-optimizable, as it cannot be riveted because it usually serves as the lid of the tank;
- The remaining region  $\Gamma$  is traction-free.

The water tank  $\Omega$  is subject to body forces such as gravity  $f \in L^2(\Omega)^d$ . In this situation, the displacement is modeled as the unique solution  $u \in H^1(\Omega)^d$  to the linearized elasticity system:

$$\begin{cases} -\nabla \cdot (Ae(u_{\Gamma_D})) = f & \text{in } \Omega, \\ Ae(u_{\Gamma_D})n_{\partial\Omega} = 0 & \text{on } \Gamma_F \cup \Gamma, \\ u_{\Gamma_D} = 0 & \text{on } \Gamma_D. \end{cases} \quad (4.92)$$

We aim to minimize the mean displacement using the least amount of material while also penalizing the boundary of the region  $\Gamma_D$ . This objective is formulated as follows:

$$\min_{\Gamma_D \subset \partial\Omega} J(G) = \frac{1}{2\text{Vol}(\Omega)} \int_{\Omega} |u_{\Gamma_D}|^2 dx. \quad (4.93)$$

In the following sections, we will present both the topological and shape derivatives.

### 4.13.2 The shape derivative

Similarly to [Section 4.2.4](#), we trade the exact problem for the approximated problem:

$$\min_{\Gamma_D \subset \partial\Omega} J_{\varepsilon}(\Gamma_D) = \frac{1}{2\text{Vol}(\Omega)} \int_{\Omega} |u_{\Gamma_D, \varepsilon}|^2 dx + \ell_1 \int_{\Gamma_D} dx + \ell_2 \int_{\partial\Gamma_D} d\sigma(x),$$

where  $u_{\Gamma_D, \varepsilon} \in H^1(\Omega)$  is the unique solution to the following boundary value problem:

$$\begin{cases} -\nabla \cdot (Ae(u_{\Gamma_D, \varepsilon})) = f & \text{in } \Omega, \\ Ae(u_{\Gamma_D, \varepsilon})n_{\partial\Omega} + h_{\Gamma_D, \varepsilon}u_{\Gamma_D, \varepsilon} = 0 & \text{on } \partial\Omega. \end{cases}$$

where the function  $h_{\Gamma_D, \varepsilon} : \partial\Omega \rightarrow \mathbb{R}$  is defined by:

$$\forall x \in \partial\Omega, \quad h_{\Gamma_D, \varepsilon}(x) = h \left( \frac{d_{\Gamma_D}^{\partial\Omega}(x)}{\varepsilon} \right)$$

where  $d_{\Gamma_D}^{\partial\Omega}$  is the signed distance function to  $\Gamma_D$  on  $\partial\Omega$  and  $h \in C^\infty(\mathbb{R})$  satisfies [\(4.8\)](#). We then calculate the shape derivative of the approximated shape functional. We omit the proof since it is very similar to the proof of [Proposition 4.4](#).

**Proposition 4.8.** *The criterion  $J_{\varepsilon}(\Gamma_D)$  is shape differentiable at  $\theta = 0$  (with  $\theta \cdot n_{\partial\Omega} = 0$ ), and its shape derivative reads:*

$$J'_{\varepsilon}(\Gamma_D)(\theta) = -\frac{1}{\varepsilon^2} \int_{\partial\Omega} h' \left( \frac{d_{\Gamma_D}^{\partial\Omega}}{\varepsilon} \right) \theta(\pi_{\Sigma_D}) \cdot n_{\Sigma_D}(\pi_{\Sigma_D}) u_{\Gamma_D, \varepsilon} \cdot p_{\Gamma_D, \varepsilon} ds,$$

where  $p_{\Gamma_D, \varepsilon} \in H^1(\Omega)$  is the weak solution to the following equation:

$$\begin{cases} -\nabla \cdot (Ae(p_{\Gamma_D, \varepsilon})) = -\frac{u_{\Gamma_D, \varepsilon}}{\text{Vol}(\Omega)} & \text{in } \Omega, \\ Ae(p_{\Gamma_D, \varepsilon})n_{\partial\Omega} + h_{\Gamma_D, \varepsilon}p_{\Gamma_D, \varepsilon} = 0 & \text{on } \partial\Omega. \end{cases}$$

### 4.13.3 The topological derivative

The setting presented in this section is exactly that from [Section 4.8](#), save from the region  $\Gamma_F$ . Thus, we can directly apply [Theorem 4.7](#) to give the expression for the topological expansion.

**Proposition 4.9.** *The perturbed criterion  $J((\Gamma)_{x_0,\varepsilon})$ , accounting for the replacement of  $\omega_{x_0,\varepsilon} \subset \Gamma$  by  $\Gamma_D$ , has the following asymptotic expansion:*

$$J(G_{x_0,\varepsilon}) = J(G) + \varepsilon M u_{\Gamma_D}(x_0) \cdot p_{\Gamma_D}(x_0) + o(\varepsilon).$$

Here, the polarization tensor  $M$  is defined by [\(4.67\)](#) and the adjoint state  $p_{\Gamma_D}$  is the unique solution  $H^1(\Omega)^d$  to the boundary value problem:

$$\begin{cases} -\nabla \cdot Ae(p_{\Gamma_D}) = -\frac{u_{\Gamma_D}}{2\text{Vol}(\Omega)} & \text{in } \Omega, \\ p_{\Gamma_D} = 0 & \text{on } \Gamma_D, \\ Ae(p_{\Gamma_D})n_{\partial\Omega} = 0 & \text{on } \Gamma_F \cup \Gamma_N \cup \Gamma. \end{cases}$$

### 4.13.4 Experiment setup

We consider the minimization of  $J(\Gamma_N, \Gamma_D)$  while also minimizing occupied surface area and contour length:

$$\min_{\Gamma_D \subset \partial\Omega} J(G) = \frac{1}{2\text{Vol}(\Omega)} \int_{\Omega} |u_{\Gamma_D}|^2 dx + \ell_1 \text{Area}(\Gamma_D) + \ell_2 \text{Cont}(\Gamma_D), \quad (4.94)$$

where  $\ell_1, \ell_2 > 0$  are penalization parameters. The domain  $\Omega$  is represented by a tetrahedral mesh  $\mathcal{T}$ , consisting of 17k vertices and 81k tetrahedrons, as shown in [Fig. 4.28](#).

On the top of the water tank, we have designated a fixed region, where no supports are permitted, meaning that it is not subject to optimization. The mesh is initialized with a circular region of with unit radius, located on the bottom of the water tank. The maximal edge length  $h_{\max}$  is fixed to 0.5 and the minimal edge length is fixed to  $h_{\min}$ .

We have conducted two experiments, with the primary difference being the contour penalization parameter  $\ell_2$ . In the first experiment, we set  $\ell_2 = 0$ , meaning no contour penalization was applied, while in the second experiment, we set  $\ell_2 > 0$ . The parameters for both experiments are detailed in [Table 4.3](#).

Parameter	Value
$\varepsilon$	0.000001
$\ell_1$	0.00001
$\ell_2$	0
$\lambda$	0.5769
$\mu$	0.3846
$\alpha$	4
$f$	(0, 0, -0.01)
$h_{\max}$	0.5
$h_{\min}$	0.05

(a) Experiment 1.

Parameter	Value
$\varepsilon$	0.000001
$\ell_1$	0.00001
$\ell_2$	0.00001
$\lambda$	0.5769
$\mu$	0.3846
$\alpha$	4
$f$	(0, 0, -0.01)
$h_{\max}$	0.5
$h_{\min}$	0.05

(b) Experiment 2.

Table 4.3: Values of the parameters used in the optimization process.

Our findings indicate that the iterative process produces reproducible results when we start with 5 iterations of topological optimization (focused solely on nucleating new small regions), followed by 45 iterations of geometrical optimization. Afterward, we perform a topological optimization step every 10 iterations until reaching 100 iterations, at which point only geometrical optimization is applied.

### 4.13.5 Analysis of results

The results of both experiments are illustrated in [Fig. 4.29](#) and [Fig. 4.31](#). In the first experiment, the algorithm clearly prioritizes the areas around the flat indentations, commonly referred to as “corrugations” or “ribs”, on the water tank. The initial topological steps strategically position the supports near these

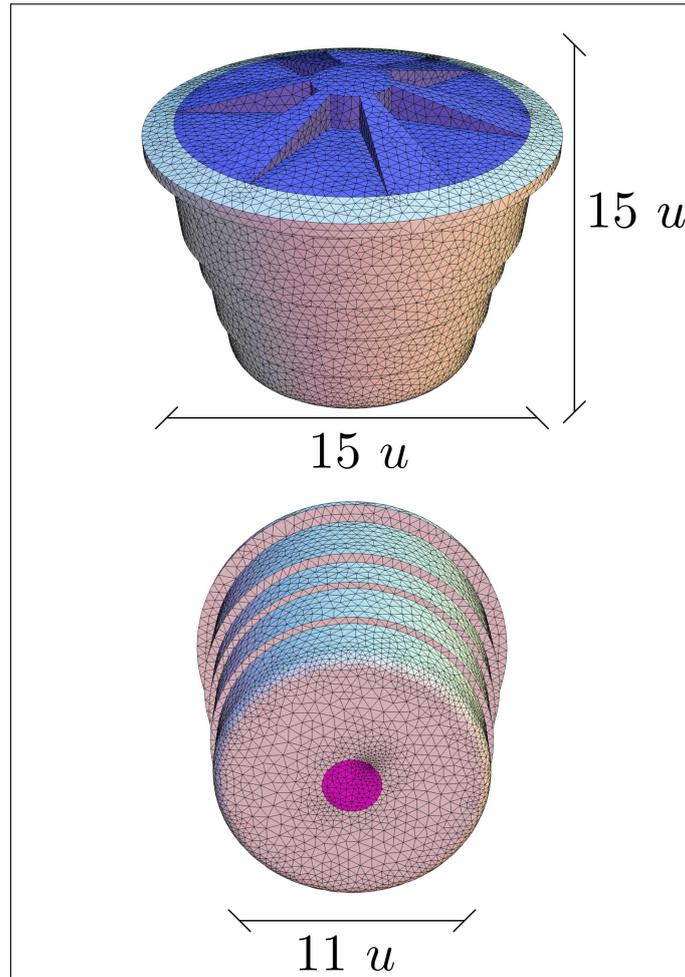
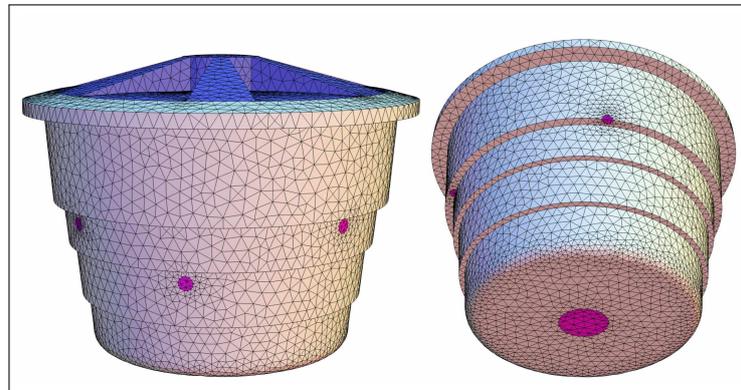


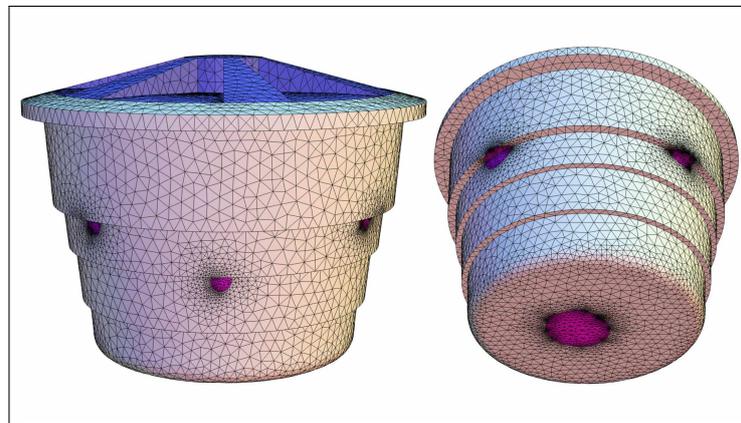
Figure 4.28: Different view angles of the tetrahedral mesh  $\mathcal{T}$  employed in the simulations. The major radius and height of the water tank are both 15 units in length, while the minor radius is 11 units wide. The blue region in the picture corresponds to  $\Gamma_F$  which is the region that will not be optimized and remains traction free. The small pink region on the bottom of the tank corresponds to the initial support region.

ribs, while the subsequent geometrical steps focus on covering most of these areas. The bottom region remains largely unchanged, highlighting its essential role in minimizing the displacement of the water tank. The final design, particularly when viewed from the bottom, underscores the importance of placing supports on the flat regions of the ribs, as the algorithm consistently targets these areas while minimizing material usage elsewhere on the mesh.

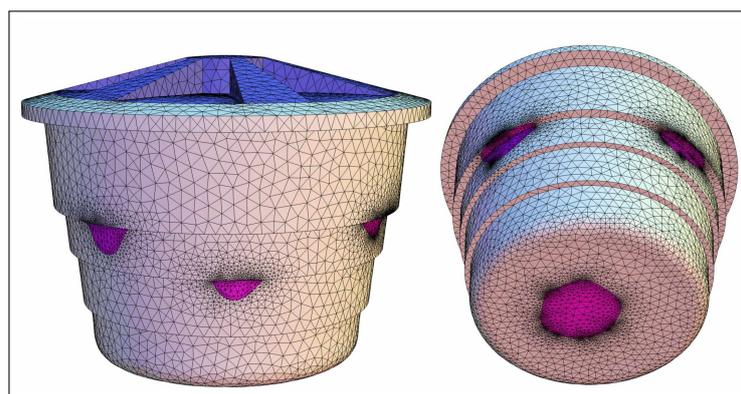
The second experiment, which incorporates contour penalization, further validates this observation. With the contour penalized, the surface shapes become more regular, evidenced by the supporting region on the bottom maintaining a more circular shape. In this final design, all supporting areas are confined to the ribs of the water tank or its bottom, with no supports extending beyond these critical regions. This behavior aligns with previous findings in the literature [113, 339], which demonstrate that corrugated structures exhibit high stiffness when forces are applied perpendicular (transverse) to the direction of the ribs, making them resistant to loads in this direction. Additionally, these structures are more flexible or compliant when forces are applied along the direction of the ribs, a property that can be advantageous in applications where movement is needed in one direction but not others. The smooth minimization of the objective by the algorithm further reinforces the validity of our results, confirming the effectiveness and reliability of the approach.



(a)  $i = 5$

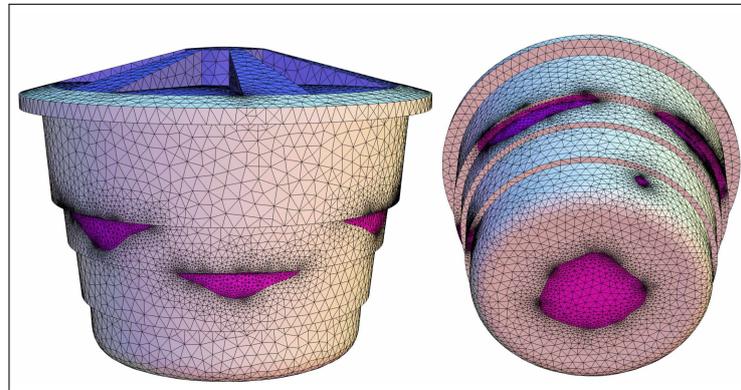


(b)  $i = 20$

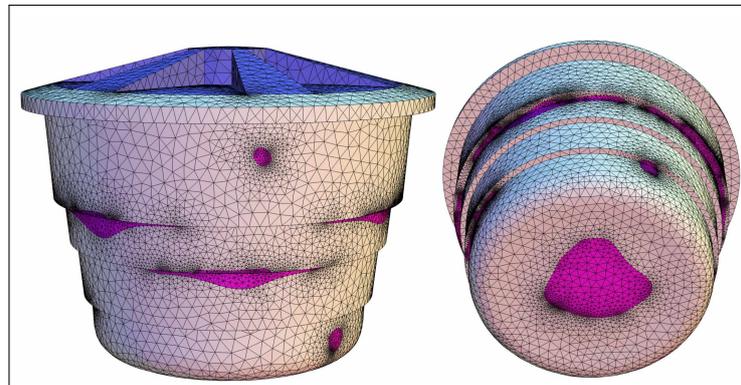


(c)  $i = 40$

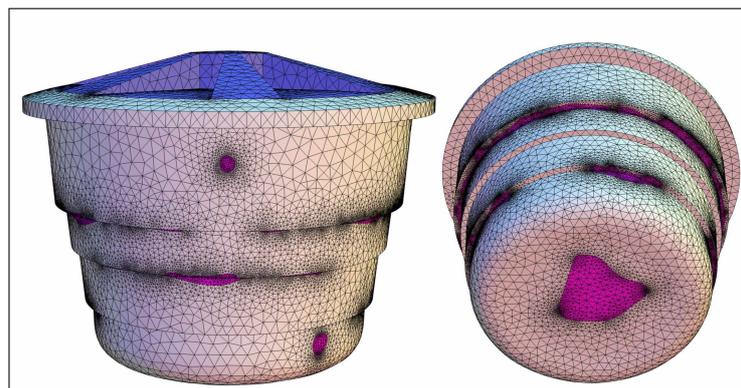
Figure 4.29: Snapshots of the optimization process of the boundary repartition for the first experiment. Here  $i$  indicates the number of iterations that have passed. The blue color corresponds to the region not being optimized, the pink region to the colour of the support region on the tank.



(d)  $i = 80$

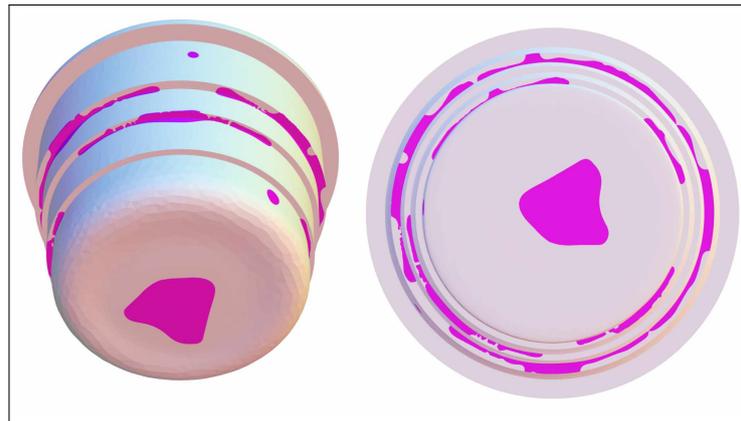


(e)  $i = 120$

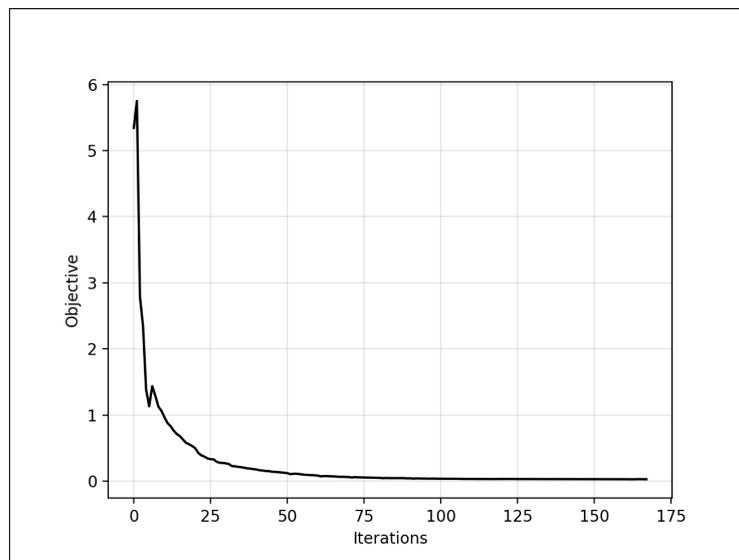


(f)  $i = 165$

Figure 4.29: (cont.) Snapshots of the optimization process of the boundary repartition for the first experiment. Here  $i$  indicates the number of iterations that have passed. The blue color corresponds to the region not being optimized, the pink region to the colour of the support region on the tank.

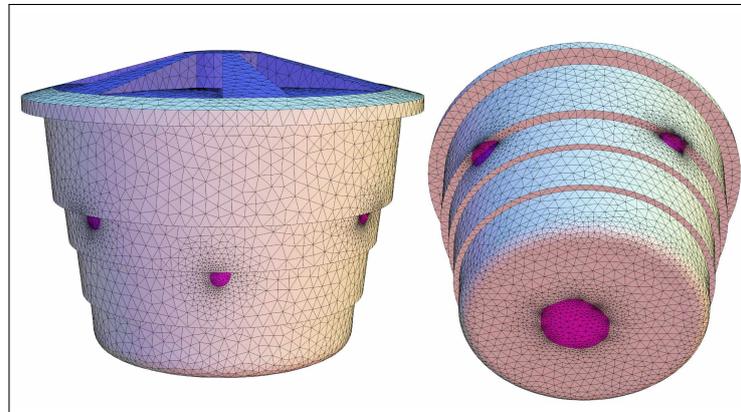


(a)  $i = 245$  (no mesh lines)

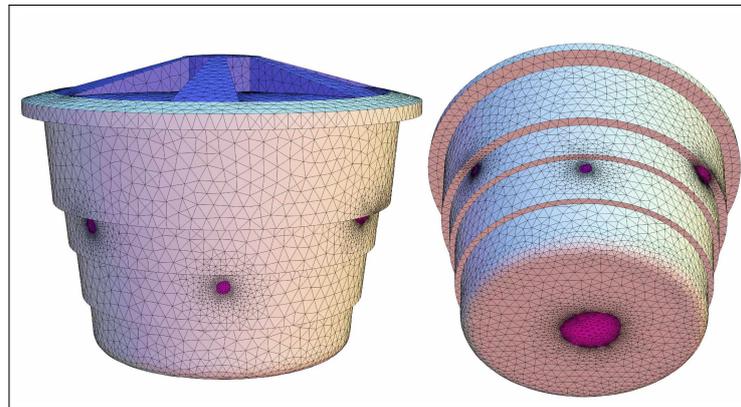


(b) Evolution of the objective criterion  $J_\varepsilon(\Gamma_D)$

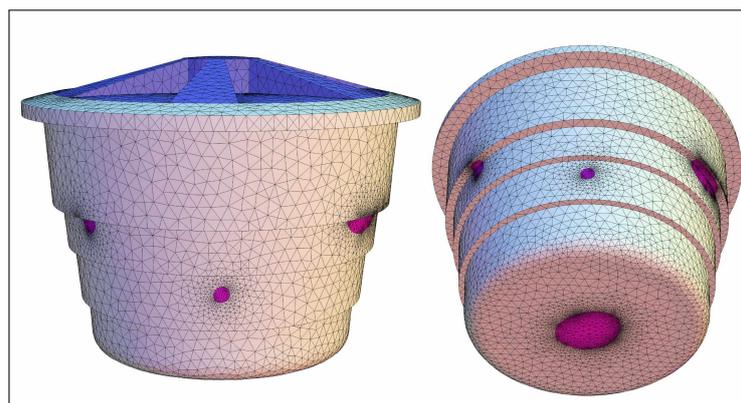
Figure 4.30: Final design of the supported region along with convergence history, for the first experiment. The bottom view of the tank is displayed to better illustrate the distribution of the supporting material in the areas prioritized by the algorithm.



(a)  $i = 20$

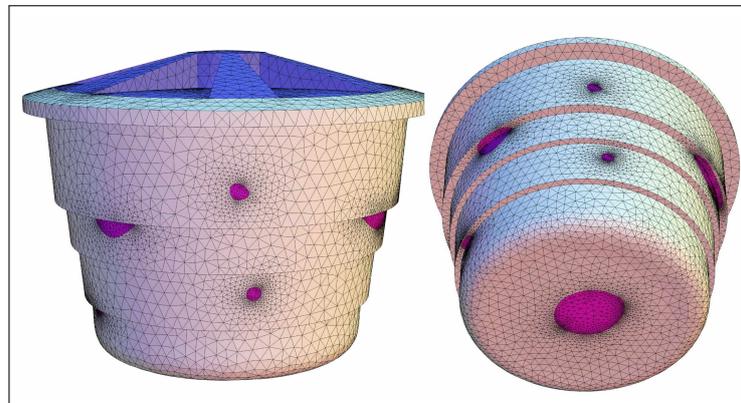


(b)  $i = 40$

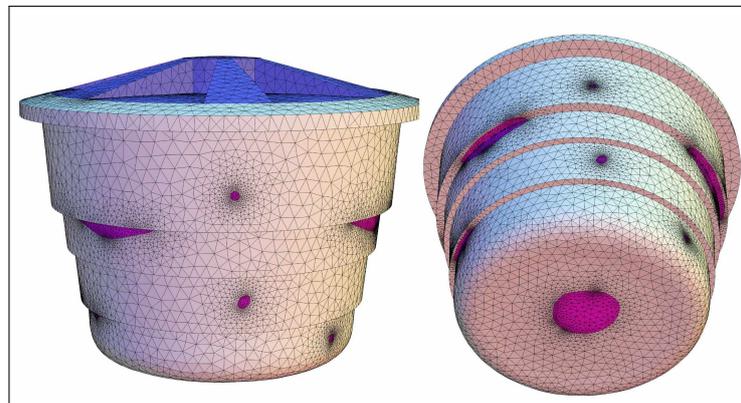


(c)  $i = 80$

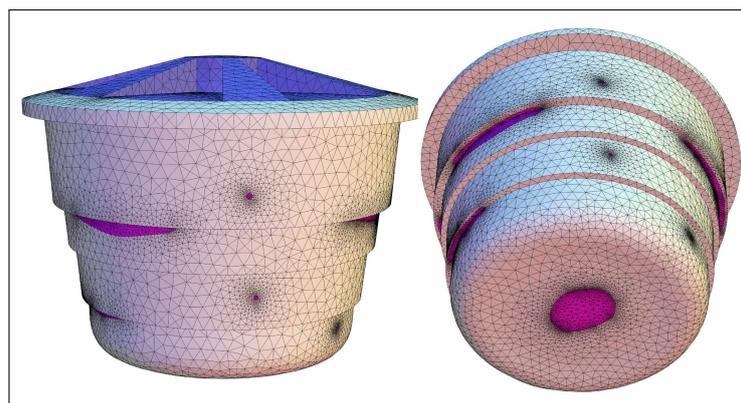
Figure 4.31: Snapshots of the optimization process of the boundary repartition for the second experiment. Here  $i$  indicates the number of iterations that have passed. The blue color corresponds to the cathode region and the orange color the anode region.



(d)  $i = 120$

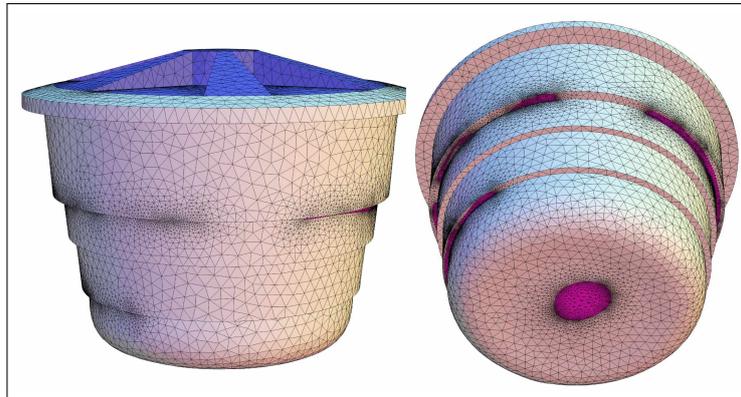


(e)  $i = 160$

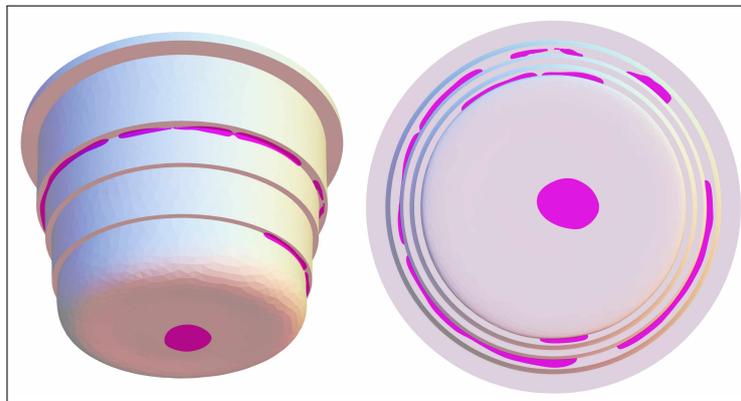


(f)  $i = 200$

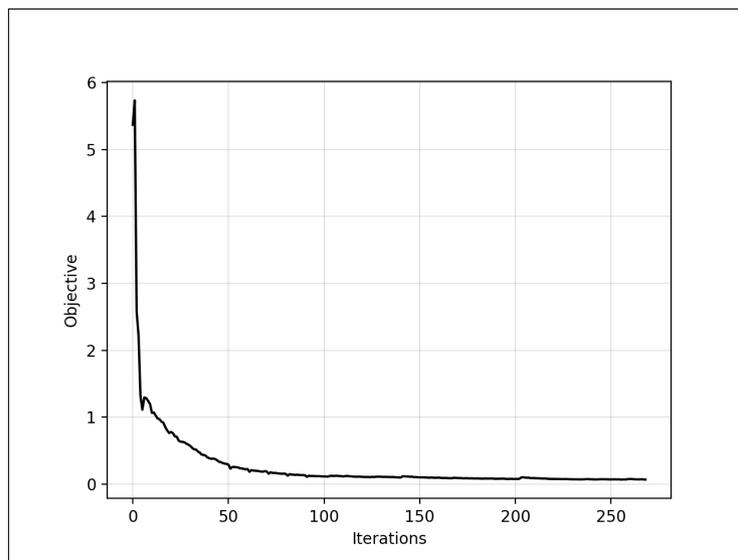
Figure 4.31: Snapshots of the optimization process of the boundary repartition for the second experiment. Here  $i$  indicates the number of iterations that have passed. The blue color corresponds to the region not being optimized, the pink region to the colour of the support region on the tank.



(g)  $i = 265$



(h)  $i = 265$  (no mesh lines)



(i) Evolution of the objective criterion  $J_\epsilon(\Gamma_D)$

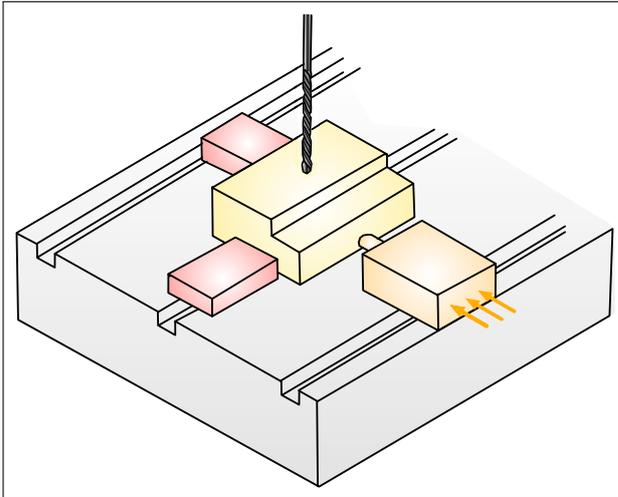
Figure 4.31: Final design of the supported region along with convergence history, for the second experiment (with contour penalization). The bottom view of the tank is displayed to better illustrate the distribution of the supporting material in the areas prioritized by the algorithm.

## 4.14 Optimization of a clamp-locator system

The “clamp-locator” system is crucial in various scientific and engineering applications for precisely positioning and holding objects or components during operations or processes. This system’s primary function is to ensure accuracy, stability, and repeatability, which are essential for achieving reliable results in experiments, manufacturing, and other technical procedures. A well-designed clamp-locator system must provide precise positioning to ensure that objects are accurately aligned according to required specifications. Any deviation from this precision can lead to errors in the results or final product, highlighting the importance of this system in maintaining accuracy.

Different fields present unique challenges for the clamp-locator system. For instance, in biological contexts, the system may need to handle delicate samples without causing damage [314], while industrial settings might require managing heavy or large components [222]. Example applications of clamp-locator systems include CNC machines [321], where they ensure parts are held in precise positions for machining, welding, or assembly; scientific research, where they position samples accurately in microscopes or other analytical instruments; medical procedures, where they hold surgical tools or components in place during operations; and robotics, where they ensure components are correctly aligned for robotic assembly or manipulation tasks.

While clamp-locator systems provide numerous advantages in manufacturing and machining operations, excessive clamping force can deform delicate workpieces, leading to higher scrap rates and material costs. Therefore, it is essential to minimize this deformation by optimizing the geometry and placement of the clamping regions. In this section, our goal is to address this issue using our boundary optimization technology by attempting to minimize the mean displacement of the piece.



(a) Schematic diagram of a clamp-locator system.



(b) Artistic depiction of a clamp-locator system in action on a CNC machine, manufacturing a component.

Figure 4.32: A clamp-locator system is essential in manufacturing and machining operations, particularly in CNC machining, to hold and position workpieces accurately and securely. It comprises clamps to apply pressure and locators to ensure precise positioning, enhancing accuracy, repeatability, and safety. This system is vital for operations requiring high precision, such as drilling, milling, assembly, and inspection, ensuring uniformity and efficiency in mass production by reducing setup time and maintaining consistent positioning.

### 4.14.1 The optimization problem

We consider that the mechanical piece is represented by a bounded Lipschitz domain  $\Omega \subset \mathbb{R}^3$ . The boundary  $\partial\Omega$  is decomposed into the following parts:

- The region  $\Gamma_D$  is that where locators are operating, i.e. the displacement of the piece is prevented.
- A strong force  $f : \Gamma_N \rightarrow \mathbb{R}^d$  is applied on the clamping region  $\Gamma_N$ .

- The tool applies a force  $g : \Gamma_T \rightarrow \mathbb{R}^d$  on the region  $\Gamma_T$ , which is fixed and not subject to optimization.
- No efforts are applied on the region  $\Gamma_F$ , which is not subject to optimization.
- The remaining region  $\Gamma$  is also traction-free, but it is subject to optimization.

Furthermore, we assume that the body forces experienced by the piece are not significant enough to be taken into account. In this situation, we model the displacement as the unique solution  $u \in H^1(\Omega)^d$  to the linearized elasticity system:

$$\begin{cases} -\nabla \cdot (Ae(u_{N,D})) = 0 & \text{in } \Omega, \\ Ae(u_{N,D})n_{\partial\Omega} = f & \text{on } \Gamma_N, \\ Ae(u_{N,D})n_{\partial\Omega} = g & \text{on } \Gamma_T, \\ Ae(u_{N,D})n_{\partial\Omega} = 0 & \text{on } \Gamma_F \cup \Gamma, \\ u_{N,D} = 0 & \text{on } \Gamma_D \end{cases} \quad (4.95)$$

for  $f, g \in C^\infty(\mathbb{R}^n)$ . We seek to minimize the total displacement of the mechanical piece by finding a configuration of  $\Gamma_N$  and  $\Gamma_D$ . This problem is formulated as:

$$\min_{\Gamma_N, \Gamma_D \subset \partial\Omega} \frac{1}{2\text{Vol}(\Omega)} \int_{\Omega} |u_{N,D}|^2 dx \quad (4.96)$$

where  $u_{N,D}$  is the solution to (4.95).

#### 4.14.2 The shape derivative

We trade the exact problem for the approximated problem:

$$\min_{\Gamma_N, \Gamma_D \subset \partial\Omega} J_\varepsilon(\Gamma_N, \Gamma_D) = \frac{1}{2\text{Vol}(\Omega)} \int_{\Omega} |u_{N,D,\varepsilon}|^2 dx,$$

where  $u_{N,D,\varepsilon} \in H^1(\Omega)$  is the unique solution to the following boundary value problem:

$$\begin{cases} -\nabla \cdot (Ae(u_{N,D,\varepsilon})) = 0 & \text{in } \Omega, \\ Ae(u_{N,D,\varepsilon})n_{\partial\Omega} = f & \text{on } \Gamma_N, \\ Ae(u_{N,D,\varepsilon})n_{\partial\Omega} = g & \text{on } \Gamma_T, \\ Ae(u_{N,D,\varepsilon})n_{\partial\Omega} = 0 & \text{on } \Gamma_F, \\ Ae(u_{N,D,\varepsilon})n_{\partial\Omega} + h_{\Gamma_D} u_{N,D,\varepsilon} = 0 & \text{on } \Gamma_D \cup \Gamma \end{cases}$$

where the function  $h_{N,\varepsilon} : \partial\Omega \rightarrow \mathbb{R}$  (resp.  $h_{D,\varepsilon}$ ) is defined by:

$$\forall x \in \partial\Omega, \quad h_{N,\varepsilon}(x) = h \left( \frac{d_{\Gamma_N}^{\partial\Omega}(x)}{\varepsilon} \right)$$

where  $d_{\Gamma_N}^{\partial\Omega}$  is the signed distance function to  $\Gamma_N$  on  $\partial\Omega$  and  $h \in C^\infty(\mathbb{R})$  satisfies (4.8). We then calculate the shape derivative of the approximated shape functional. We omit the proof since it is very similar to the proof of Proposition 4.4.

**Proposition 4.10.** *The criterion  $J_\varepsilon(\Gamma_D, \Gamma_T)$  is shape differentiable at  $\theta = 0$  (with  $\theta \cdot n_{\partial\Omega} = 0$ ), and its shape derivative reads:*

$$J'_\varepsilon(\Gamma_N, \Gamma_D)(\theta) = -\frac{1}{\varepsilon^2} \int_{\partial\Omega} h' \left( \frac{d_{\Gamma_D}^{\partial\Omega}}{\varepsilon} \right) \theta(\pi_{\partial\Gamma_D}) \cdot n_{\partial\Gamma_D}(\pi_\Sigma) u_{N,D,\varepsilon} \cdot p_{N,D,\varepsilon} ds + \int_{\partial\Gamma_N} g \cdot p_{N,D,\varepsilon} \theta \cdot n_{\partial\Gamma_N} d\sigma(x)$$

where  $p_{N,D,\varepsilon} \in H^1(\Omega)$  is the weak solution to the following equation:

$$\begin{cases} -\nabla \cdot (Ae(p_{N,D,\varepsilon})) = -\frac{u_{N,D,\varepsilon}}{\text{Vol}(\Omega)} & \text{in } \Omega, \\ Ae(p_{N,D,\varepsilon})n_{\partial\Omega} + h_{D,\varepsilon} p_{N,D,\varepsilon} = 0 & \text{on } \partial\Omega. \end{cases}$$

### 4.14.3 The topological derivative

The sensitivities of the shape functional  $J(\Gamma_D, \Gamma_N)$  with respect to the addition of a small surface disk  $\omega_{x_0, \varepsilon}$ , centered at  $x_0 \in \Gamma$ , are presented in the following result. The expansion of  $J((\Gamma_D)_{x_0, \varepsilon}, \Gamma_N)$  comes as a direct result of [Theorem 4.7](#) while the proof of the expansion for  $J(\Gamma_D, (\Gamma_N)_{x_0, \varepsilon})$  is analogous to the results of [Section 4.6.2](#), so it is omitted here for the sake of brevity

**Proposition 4.11.** *Let  $x_0 \in \Gamma$  be given. Then,*

(i) *The perturbed criterion  $J((\Gamma_N)_{x_0, \varepsilon}, \Gamma_D)$ , accounting for the addition of  $\omega_{x_0, \varepsilon} \subset \Gamma$  to  $\Gamma_N$ , has the following asymptotic expansion:*

$$J((\Gamma_N)_{x_0, \varepsilon}, \Gamma_D) = \begin{cases} J(\Gamma_N, \Gamma_D) - 2\varepsilon g(x_0) \cdot p_{N,D}(x_0) + o(\varepsilon) & \text{if } d = 2, \\ J(\Gamma_N, \Gamma_D) - \pi\varepsilon^2 g(x_0) \cdot p_{N,D}(x_0) + o(\varepsilon^2) & \text{if } d = 3. \end{cases}$$

(ii) *The perturbed criterion  $J(\Gamma_N, (\Gamma_D)_{x_0, \varepsilon})$ , accounting for the addition of the  $\omega_{x, \varepsilon} \subset \Gamma$  to  $\Gamma_D$ , has the following asymptotic expansion:*

$$J(\Gamma_N, (\Gamma_D)_{x_0, \varepsilon}) = \begin{cases} J(\Gamma_N, \Gamma_D) + \frac{1}{|\log \varepsilon|} \frac{\pi \mu}{1-\nu} u_0(x_0) \cdot p_{N,D}(x_0) + o\left(\frac{1}{|\log \varepsilon|}\right) & \text{if } d = 2, \\ J(\Gamma_N, \Gamma_D) + \varepsilon M u_0(x_0) \cdot p_{N,D}(x_0) + o(\varepsilon) & \text{if } d = 3. \end{cases}$$

In the above, the polarization tensor  $M$  is defined by [\(4.67\)](#) and the adjoint state  $p_{N,D} \in H^1(\Omega)^d$  is characterized by the following boundary value problem:

$$\begin{cases} -\nabla \cdot (Ae(p_{N,D})) = -\frac{u_{N,D}}{\text{Vol}(\Omega)} & \text{in } \Omega, \\ Ae(p_{N,D})n_{\partial\Omega} = 0 & \text{on } \Gamma \cup \Gamma_N \cup \Gamma_T \cup \Gamma_F, \\ p_{N,D} = 0 & \text{on } \Gamma_D. \end{cases}$$

### 4.14.4 Experiment setup

We consider the minimization of  $J(\Gamma_N, \Gamma_D)$  along with area penalization:

$$\min_{\Gamma_N, \Gamma_D \subset \partial\Omega} \frac{1}{2\text{Vol}(\Omega)} \int_{\Omega} |u_{\Gamma_N, \Gamma_D}|^2 dx + \ell_1 \text{Area}(\Gamma_N) + \ell_2 \text{Area}(\Gamma_N). \quad (4.97)$$

where  $\ell_2, \ell_1 > 0$  are weak penalization parameters on the surface area of the locator region and clamp region, respectively. The domain  $\Omega$  is discretized using a tetrahedral mesh  $\mathcal{T}$ , comprising 28,000 vertices and 146,000 tetrahedrons, as illustrated in [Figs. 4.33](#) and [4.34](#). The top and bottom regions (highlighted in pink) are excluded from the optimization process. The pink region represents  $\Gamma_F$ , while the orange region designates  $\Gamma_T$ , where the tool applies force. The parameters used in our experiments are outlined in [Table 4.4](#). To ensure the reproducibility of the optimization process, we found that performing a topological step every 10 iterations—including the first two iterations—was most effective for both the locator and clamp regions. This process continues until 100 iterations are completed, after which only geometrical optimization is carried out. The evolution scheme follows a strategy similar to that described in [Section 4.11](#), alternating between the optimization of the locator and clamp regions, one at a time.

Parameter	Value
$\varepsilon$	0.00001
$\ell_1$	0.0001
$\ell_2$	0.0001
$\alpha$	4
$f$	$-n_{\partial\Omega}/10$
$g$	(1, 0, 0)
$h_{max}$	0.2
$h_{min}$	0.02

Table 4.4: Values of the parameters used in the optimization process.

#### 4.14.5 Analysis of results

The snapshots of the experiment are shown in Fig. 4.35, with the final design and objective history illustrated in Fig. 4.36. The application of our method to the clamp-locator system is particularly significant for validating our approach, as it involves optimizing two distinct zones that impose different boundary conditions on a complex, curved geometry. This problem is challenging because many configurations could potentially stabilize the system and minimize displacement, making it difficult to identify an optimal configuration. Further complicating the issue, our experiments revealed that adding a new clamp region could destabilize the system, leading to an increase in the objective functional, as the new clamp might push the piece in a direction that isn't adequately located. Despite these challenges, the algorithm successfully minimizes the mean displacement, reducing the objective from 46.1951 to approximately 0.08, a 577-fold decrease. Notably, the final design in Fig. 4.36 predominantly relies on clamp regions to achieve this minimization. The optimal design features small locating regions at the front and back, with elliptical regions positioned opposite the direction of the load applied by the tool. This outcome is unexpected, as one might initially assume that the method would favor adding locator regions extensively and removing clamp regions, resulting in a trivial solution. However, the algorithm identifies a non-trivial local minimum, demonstrating its effectiveness and revealing various possible configurations.

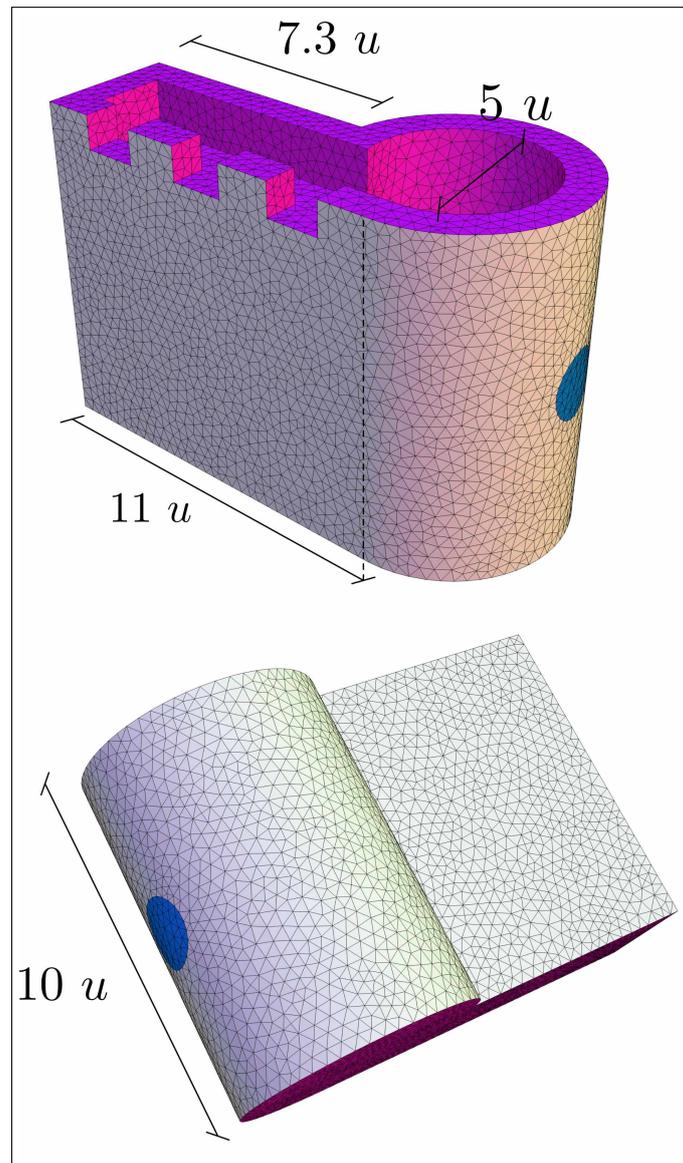


Figure 4.33: Different view angles of the tetrahedral mesh  $\mathcal{T}$  employed in the clamp-locator simulations. The pink region on the bottom and top of the mechanical piece represents the region  $\Gamma_F$ , which is traction free and not subject to optimization. The blue region corresponds to the initial locator region  $\Gamma_D$ . The rest is the traction free boundary  $\Gamma$ .

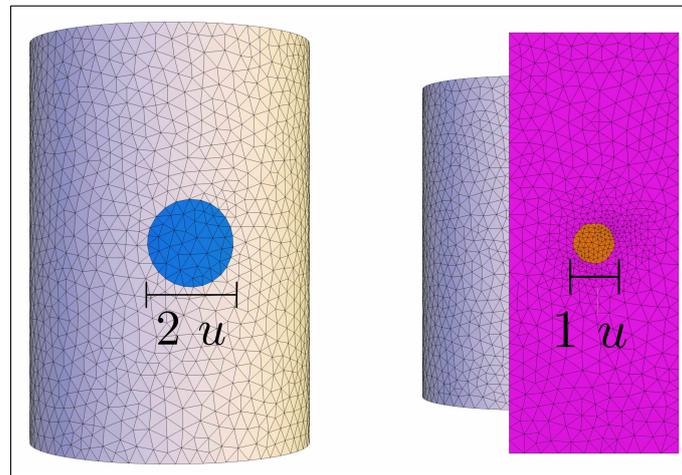
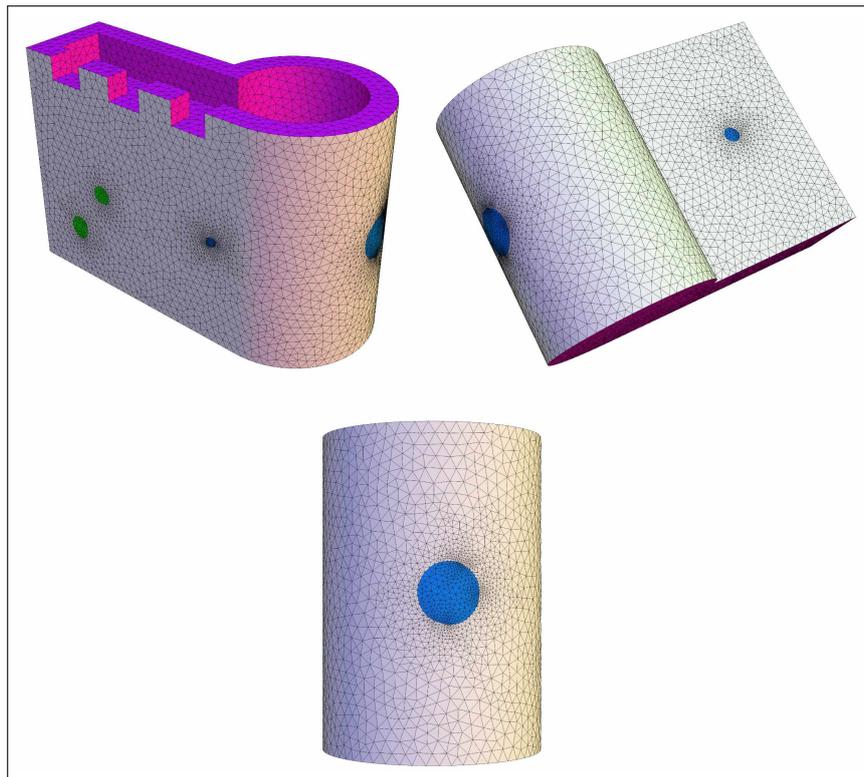
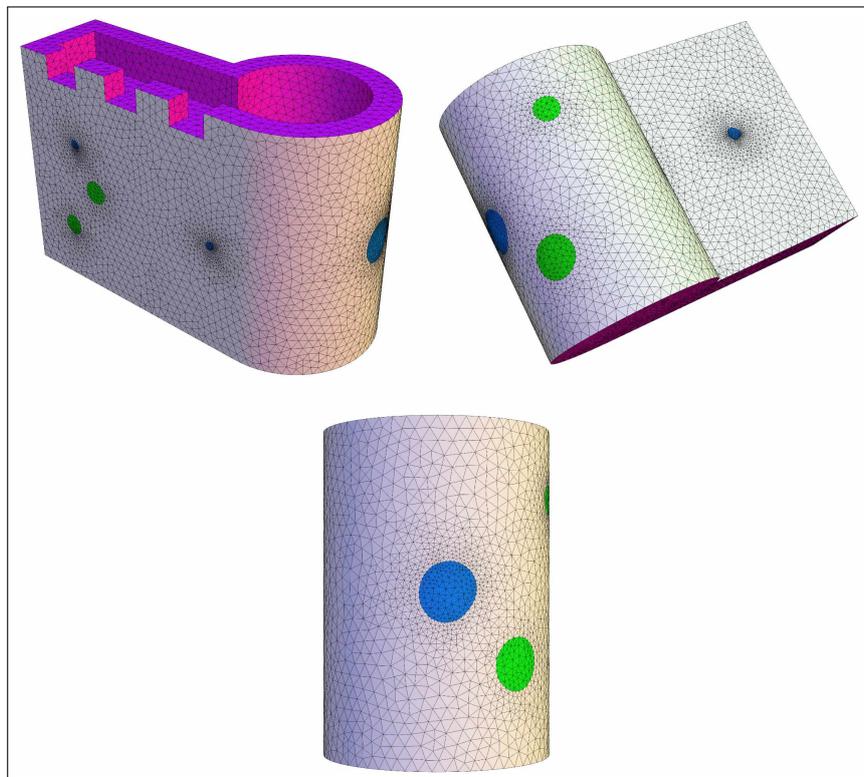


Figure 4.34: Angles of the front and back of the mechanical piece. The blue region corresponds to the initial guess of the locator region  $\Gamma_D$ . It is 2 units wide in diameter. The orange circle contained within the pink region  $\Gamma_F$  represents the region  $\Gamma_T$  where the tools is applying the force.



(a)  $i = 20$



(b)  $i = 40$

Figure 4.35: Snapshots of the optimization process of the boundary repartition. Here  $i$  indicates the number of iterations that have passed. The blue color corresponds to the locator region  $\Gamma_D$  and the green color represents the clamp region  $\Gamma_N$ .

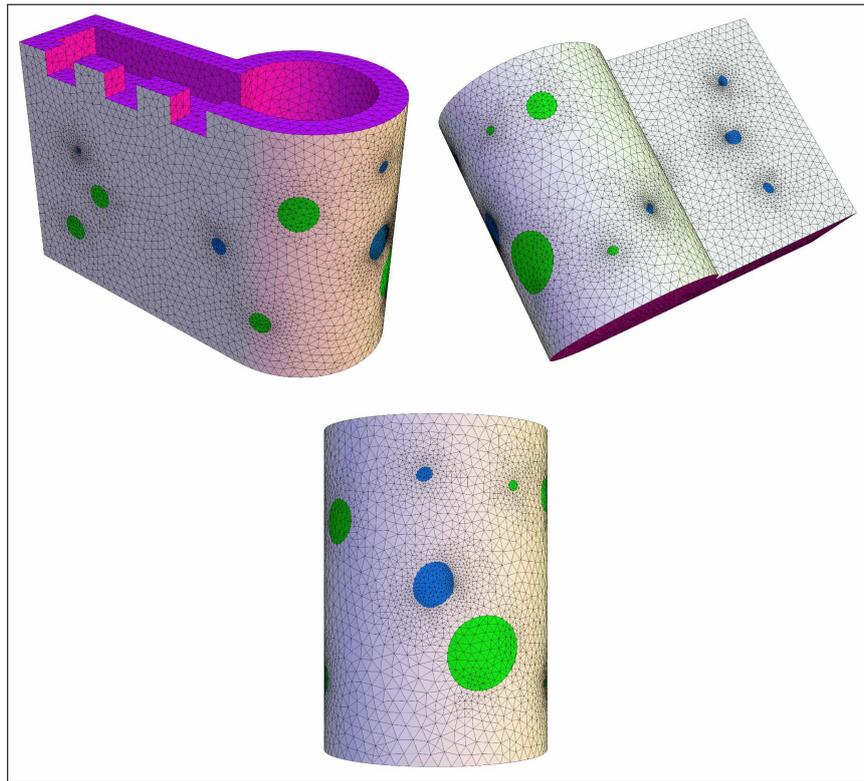
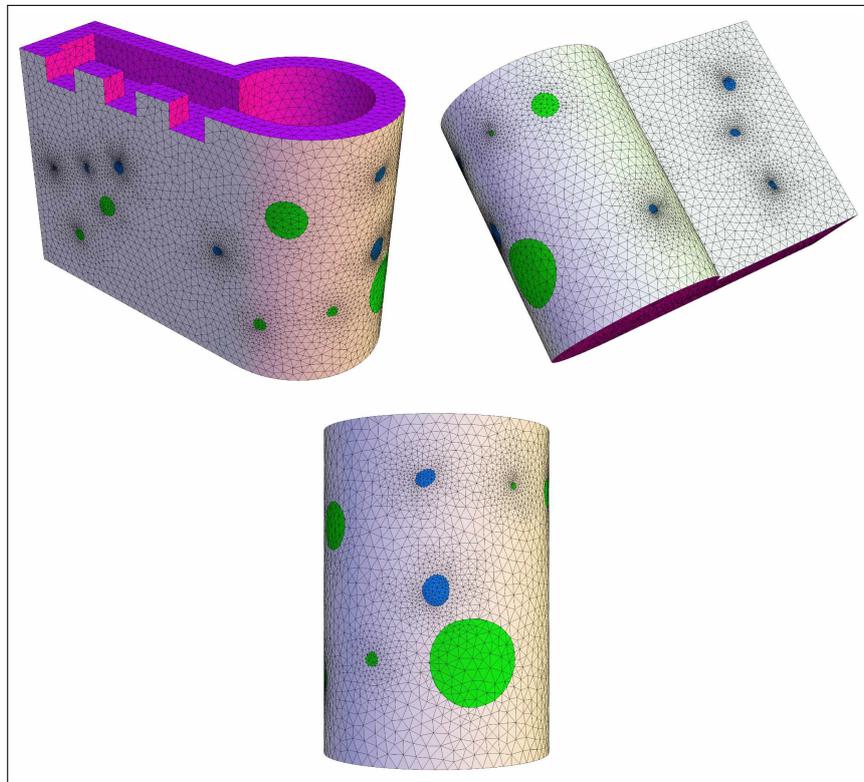
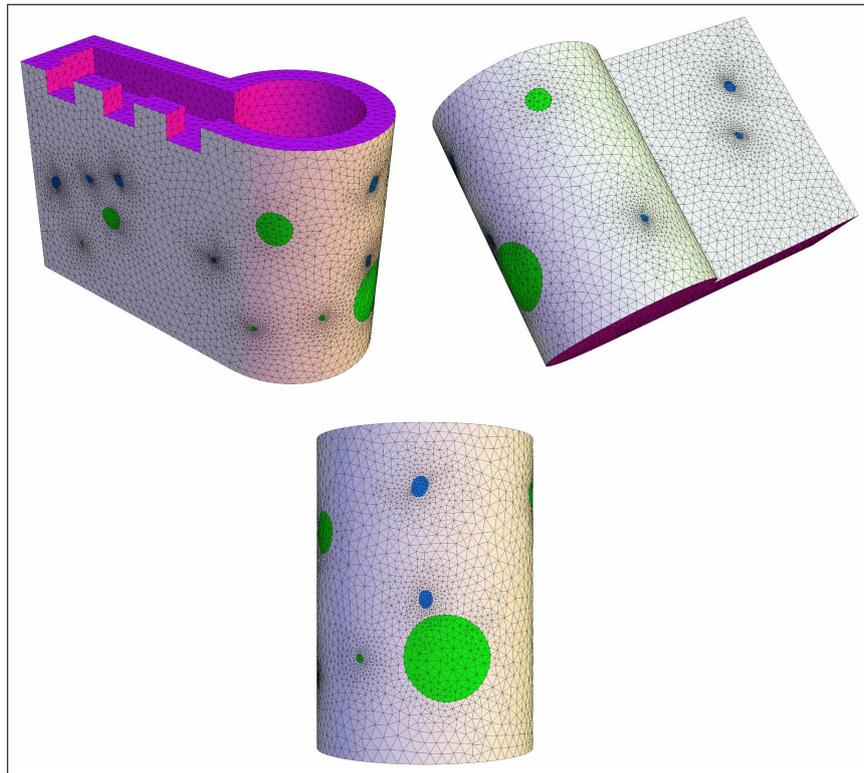
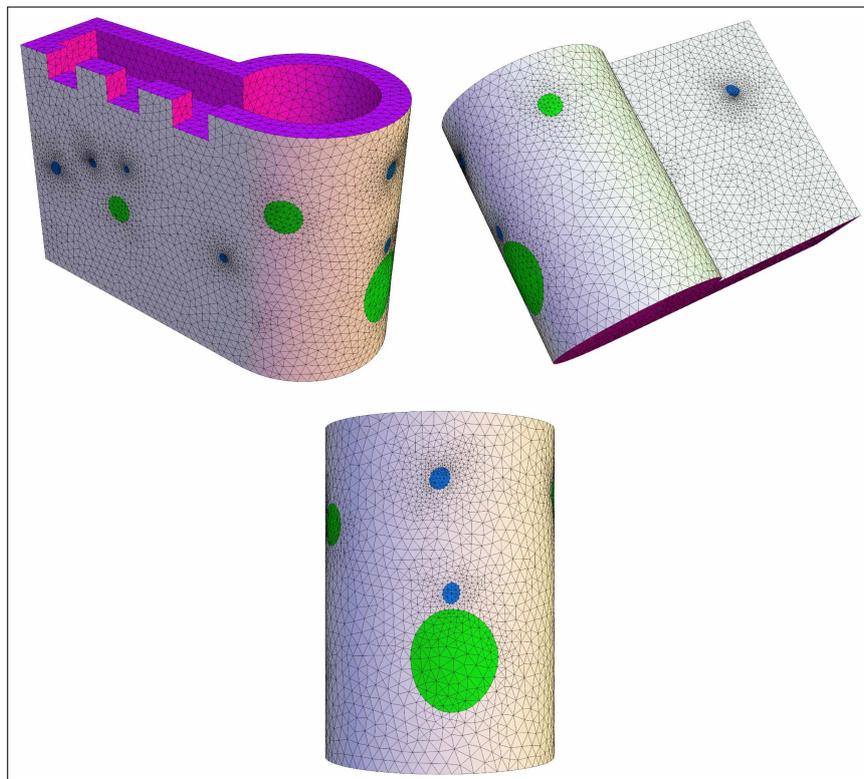
(c)  $i = 80$ (d)  $i = 120$ 

Figure 4.35: (cont.) Snapshots of the optimization process of the boundary repartition. Here  $i$  indicates the number of iterations that have passed. The blue color corresponds to the locator region  $\Gamma_D$  and the green color represents the clamp region  $\Gamma_N$ .



(e)  $i = 160$



(f)  $i = 200$

Figure 4.35: (cont.) Snapshots of the optimization process of the boundary repartition. Here  $i$  indicates the number of iterations that have passed. The blue color corresponds to the locator region  $\Gamma_D$  and the green color represents the clamp region  $\Gamma_N$ .

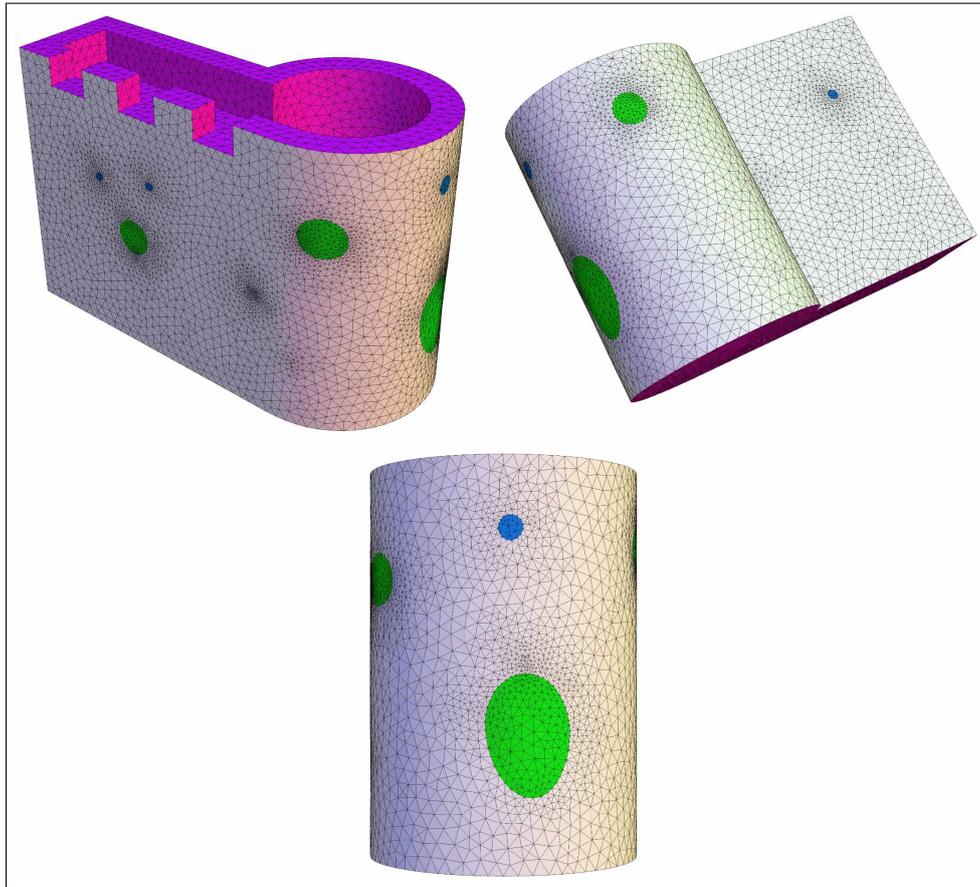
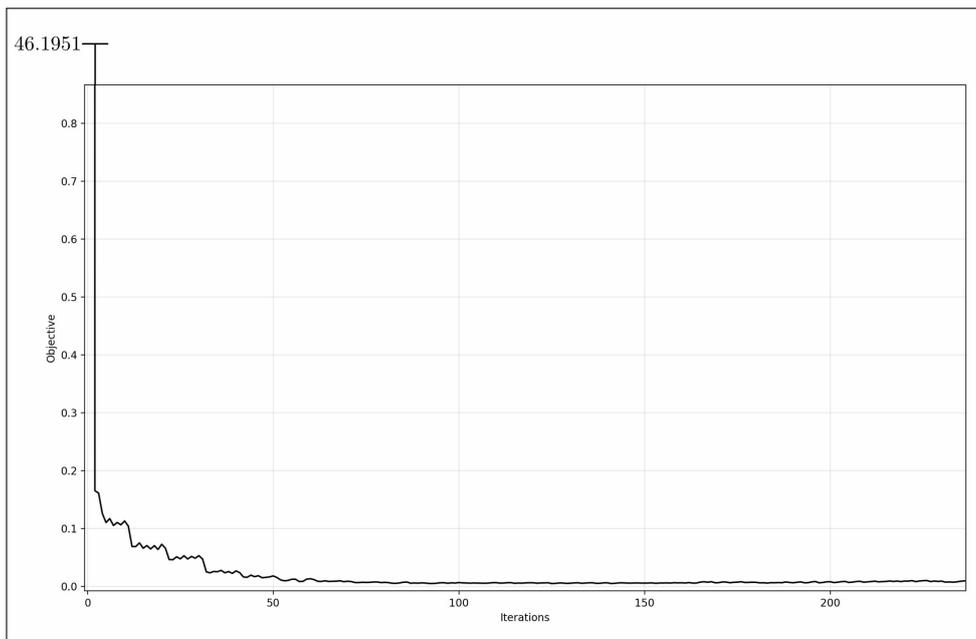
(a)  $i = 245$ (b) Evolution of the objective criterion  $J_\epsilon(\Gamma_D)$ 

Figure 4.36: Final design of the clamp-locator design for the mechanical piece along with the objective convergence history.

# Chapter 5

## Rodin: A numerical C++20 library for shape and topology optimization

---

Rodin is a lightweight and modular finite element framework designed to support the implementation of shape and topology optimization algorithms. It offers a comprehensive suite of functionalities essential for these tasks, including tools for refining and remeshing the underlying shape and elegant mechanisms for specifying and solving variational problems. This chapter will explore the design principles, key features, and practical applications of Rodin, demonstrating how it facilitates efficient and flexible optimization workflows in numerical simulations. At the time of writing, the library is available at:

<https://github.com/cbritopacheco/rodin>,

under the `Boost Software License 1.0`. Furthermore, the latest documentation can be found at:

<https://cbritopacheco.github.io/rodin>.

All the numerical examples generated in this thesis are a product of Rodin.

### 5.1 Introduction

In the realm of computational science and engineering, the finite element method [136, 190, 92] is a powerful numerical technique used to solve complex partial differential equations that describe a wide range of physical phenomena, from fluid dynamics to structural mechanics and electromagnetics. These equations are fundamental in modeling the behavior of systems under various conditions, allowing scientists and engineers to predict outcomes, optimize designs, and solve real-world problems.

However, translating the mathematical formulations of these PDEs into computational code can be challenging. The process often involves reformulating the equations in ways that are not immediately intuitive, which can lead to errors, inefficiencies, and a steep learning curve for those new to the field.

#### 5.1.1 Philosophy

To address these challenges, there is a growing interest in finite element libraries that closely resemble the mathematical notation used in the original PDE formulations [24]. Such libraries aim to bridge the gap between theoretical models and their computational implementation, providing tools that not only perform the necessary numerical computations but do so in a way that remains faithful to the original mathematical expressions. This approach offers numerous advantages, such as:

1. **Intuitive understanding and ease of use.** When a finite element library mirrors the mathematical notation of PDEs, it aligns more naturally with the thought process of scientists, engineers, and

mathematicians. This close resemblance helps users to more easily translate their theoretical models into computational code, reducing the cognitive load associated with interpreting or reformatting the problem into a less familiar or more abstract code structure.

2. **Reduced error rate.** The more closely a computational implementation matches the original mathematical formulation, the less room there is for errors during transcription from theory to code [207]. When the code closely reflects the mathematical equations, it becomes easier to verify and validate the correctness of the implementation, leading to more reliable and accurate simulations.
3. **Improved collaboration.** Many projects in computational science and engineering are collaborative, often involving individuals with different expertise (e.g., mathematicians, engineers, computer scientists). A library that uses familiar mathematical notation allows different experts to understand, review, and contribute to the code without needing extensive knowledge of the programming details, fostering interdisciplinary collaboration.
4. **Educational value.** For students and those new to the field, a finite element library that uses familiar mathematical notation can serve as an educational tool. It helps them bridge the gap between learning the theory behind PDEs and applying it computationally, thereby enhancing their understanding of both the mathematical concepts and the numerical methods used to solve them.
5. **Direct mapping from theory to implementation.** A library that closely follows the mathematical notation allows for a direct mapping from theoretical models to computational implementation. This is particularly important in fields where complex PDEs govern physical phenomena, such as fluid dynamics, electromagnetics, or structural analysis. The ability to directly implement these models without needing to significantly alter their form increases the efficiency of model development and testing.
6. **Enhanced flexibility for complex models.** When dealing with complex or non-standard PDEs, a library that uses a notation similar to mathematical formulations allows researchers to more easily modify or extend existing models. The mathematical resemblance makes it easier to experiment with different formulations, boundary conditions, and material properties, providing greater flexibility in exploring new ideas or refining existing models.
7. **Alignment with theoretical developments.** Theoretical advancements in numerical methods and PDE analysis often introduce new techniques and formulations. A finite element library that aligns with mathematical notation is better positioned to integrate these developments quickly. This close alignment allows users to stay up-to-date with the latest methodologies, ensuring that their computational tools remain relevant and effective.

### 5.1.2 Similar libraries to Rodin

Several finite element libraries have been developed with the goal of closely resembling the mathematical notation used in the modeling of partial differential equations and physical phenomena. These libraries strive to maintain a high level of abstraction, making it easier for users to translate mathematical formulations directly into code. Some of the notable libraries that follow this trend include:

- **FEniCS** [23] is one of the most prominent examples of a finite element library that emphasizes a mathematical approach to problem formulation. It allows users to define PDEs using variational forms that closely resemble their mathematical expressions. The library provides tools for automatic differentiation, mesh generation, and solving a wide range of PDEs, making it a popular choice for researchers and educators.
- **FiPy** [173] is a Python-based finite volume library for solving PDEs, particularly those involving diffusion, convection, and reaction. While it is not a finite element library in the strictest sense, FiPy's design philosophy aligns with the trend of resembling mathematical notation. It allows users to define PDEs and boundary conditions in a way that mirrors their mathematical formulation, making it intuitive and easy to use.
- **MFEM** [32] is a modular finite element library that supports high-performance computing. It also offers a high-level abstraction that aligns with the mathematical notation used in PDE formulations. MFEM provides tools for defining variational problems and supports advanced features like adaptive mesh refinement and high-order elements, while maintaining an intuitive interface for users.

- **FreeFem++** [182] is an open-source, high-level integrated development environment (IDE) for solving partial differential equations using the finite element method. It is widely recognized for its ease of use and its ability to allow users to write code that is very close to the mathematical formulation of the problems they are solving. In FreeFem++, the user can define the domain, the PDE, and the associated boundary conditions using a syntax that closely matches the mathematical notation. This makes it particularly appealing to researchers and engineers who are already familiar with the mathematical theory behind PDEs, as it reduces the cognitive gap between theory and implementation. FreeFem++ supports a wide variety of applications, including fluid dynamics, elasticity, heat transfer, and more. Its high-level scripting language enables rapid prototyping and experimentation with different models, while also supporting advanced features like automatic mesh generation, adaptive mesh refinement, and parallel computing.

These libraries are part of a growing movement in computational science to create tools that not only solve complex problems but do so in a way that is accessible and closely aligned with the theoretical formulations. This approach helps to lower the barriers to entry, making advanced computational techniques more widely usable and understandable across different disciplines.

## 5.2 Modules

In this section, we provide an overview of the core modules that make up Rodin. Each module is discussed in detail, outlining its role within the framework and how it interacts with other components. This allows for a comprehensive understanding of how Rodin operates and how its components can be leveraged for efficient and accurate finite element computations.

- **Alert.** This module manages all the messages, warnings, and exceptions generated by Rodin. It ensures that users are kept informed about important events, potential issues, and errors during the execution of their computations.
- **Assembly.** This module is responsible for converting template expressions into linear algebra objects, such as matrices and vectors. It essentially "assembles" the mathematical expressions into forms that can be processed and solved by numerical methods.
- **Context.** This module helps manage the computing environment in which Rodin operates. It carries information about the machine's architecture, such as whether it's running on CUDA, OpenMPI, or a distributed network. Currently, only a Sequential context (single-threaded execution) is implemented, but this module is designed to be expandable for more complex architectures in the future.
- **FormLanguage.** This module supports the creation and manipulation of variational form expressions. It provides type traits and utility functions that help in defining and working with the mathematical forms used in finite element methods.
- **Variational.** This module contains the core classes that make up Rodin's variational form language. These classes are used to construct template expressions that closely resemble mathematical notation, making it easier to model and solve PDEs in a way that aligns with theoretical formulations.
- **Geometry.** This module manages all the functions and data structures related to geometry, particularly the mesh on which the PDEs are solved. It handles the representation of geometric entities and ensures that the mesh is correctly structured for the numerical solution process.
- **IO.** The IO module handles all input and output operations in Rodin. It allows users to save and load solutions, meshes, and other data in various formats, ensuring compatibility with different tools and workflows.
- **QF.** This module includes all the quadrature formulae supported by Rodin. Quadrature formulas are essential for numerical integration, which is a key part of the finite element method.
- **Solver.** The Solver module contains the numerical solvers used to solve the linear systems that arise from the assembly process. It includes implementations of methods like Conjugate Gradient, among others, which are essential for solving large, sparse systems efficiently.

- **Threads.** This module offers tools for managing multithreading in Rodin. It provides the necessary support to run computations in parallel, leveraging multi-core processors to speed up the solution process.
- **Utility.** The Utility module provides various metaprogramming utilities that are used throughout Rodin. These utilities help streamline code development and ensure that the framework remains flexible and efficient.

## 5.3 Theoretical basis

This section establishes the theoretical foundation upon which the library is built. It introduces essential concepts such as polygons, polyhedra, and meshes, which form the geometric and mathematical basis for the library's functionality.

### 5.3.1 A primer on meshes

Meshes are crucial in the finite element method as they discretize a continuous domain into smaller elements, enabling local approximation of the solution to partial differential equations. This discretization transforms the continuous problem into a system of algebraic equations, allowing FEM to handle complex geometries and boundary conditions efficiently. The fundamental building block of a mesh is a polytope.

**Definition 5.1.** *In dimension 2, a **polygon** is a domain whose boundary is a finite union of segments. In dimension 3, a **polyhedron** is a domain whose boundary is a finite union of polygons. When the distinction is not relevant, the term **polytope** is employed.*

With this definition, we can give the mathematical definition of a mesh.

**Definition 5.2.** *Let  $\Omega$  be a domain in  $\mathbb{R}^d$ . A **mesh** is defined as a collection of open polytopes, denoted as  $\mathcal{T} = \{\tau_k\}_{k=1,\dots,N}$ , which collectively cover the domain  $\Omega$  such that:*

$$\bar{\Omega} = \bigcup_{k=1}^N \bar{\tau}_k.$$

*This means that the union of the closures of all the polytopes forms the closure of the domain  $\Omega$ . Additionally, the triangulation must satisfy two important conditions:*

1. *Non-overlapping polytopes. The polytopes  $\tau_k$  must not overlap, i.e.,  $\tau_k \cap \tau_l = \emptyset$  whenever  $k \neq l$ .*
2. *Conformity of the mesh. The mesh  $\mathcal{T}$  must be conforming, meaning that for any two polytopes  $\tau_k$  and  $\tau_l$ , their intersection  $\bar{\tau}_k \cap \bar{\tau}_l$  is either a shared vertex, a shared edge, or, in the case of three-dimensional shapes, a shared face of the triangulation  $\mathcal{T}$ .*

*The dimension of a mesh  $D = \dim(\mathcal{T})$  is defined as the dimension of the highest dimensional polytope.*

**Remark 5.1 (Cells, faces and vertices).** *In a mesh, **cells** are the highest-dimensional elements (e.g., triangles or quadrilaterals in 2D, tetrahedra or hexahedra in 3D) that partition the domain for numerical approximation. The **faces** are  $(D-1)$ -dimensional boundaries between cells (e.g., edges in 2D, or polygonal surfaces in 3D) and may also form part of the domain boundary. The **vertices** are 0-dimensional points where faces meet, defining the corners of cells. See [Table 5.1](#) and [Fig. 5.1](#).*

In practice, a mesh is generated from a reference cell, say  $\hat{\tau}$ , and a set of geometric transformations mapping  $\hat{\tau}$  to the physical mesh cells  $\tau$ .

**Definition 5.3 (Polytope transformation).** *Let  $\tau \in \mathcal{T}$ . If there exists a  $C^1$ -diffeomorphism  $T_\tau : \hat{\tau} \rightarrow \tau$ , then we say  $T_\tau$  is a **polytope transformation** from the **reference polytope**  $\hat{\tau}$  to the **physical polytope**  $\tau$ . See [Fig. 5.2](#)*

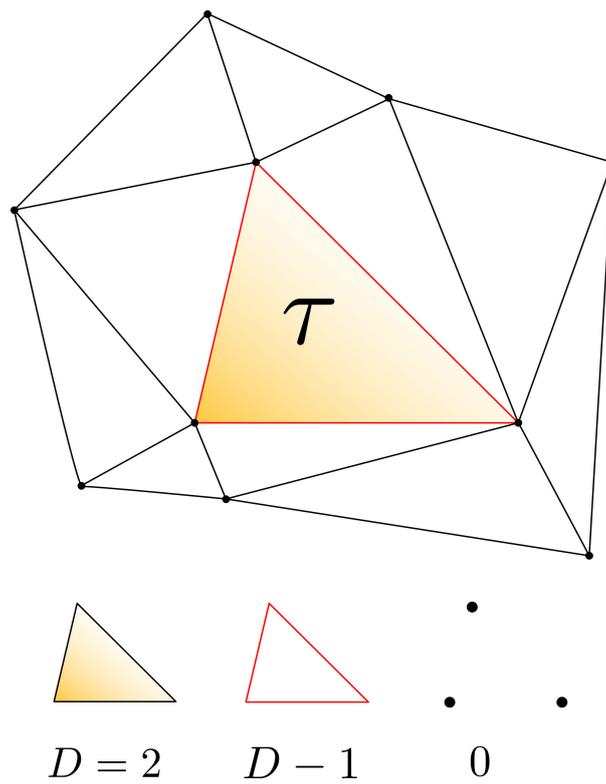


Figure 5.1: Mesh entities in a mesh of dimension  $D = 2$ . The yellow triangle represents the polytope of highest dimension, the red lines are referred to as the edges (or faces) of the mesh, while the 0 dimensional points are the vertices.

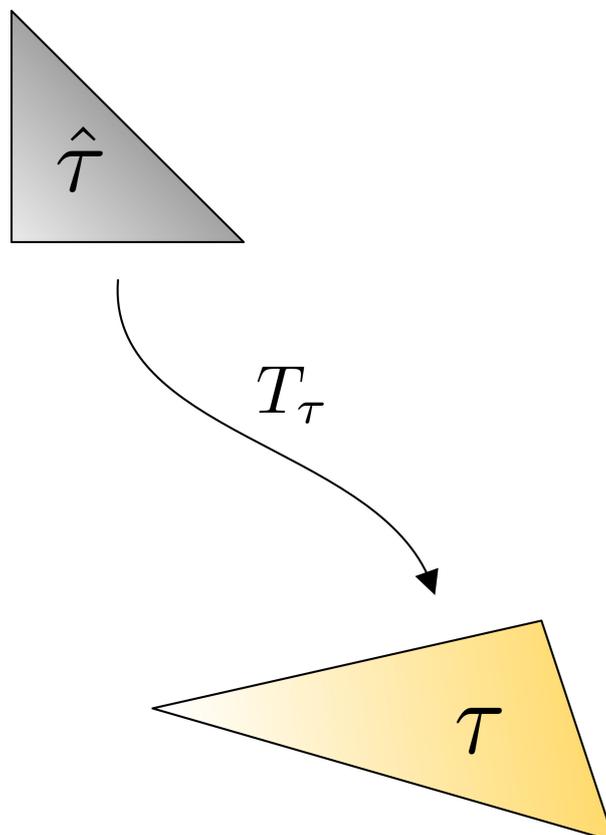


Figure 5.2: Transformation from a reference element  $\hat{\tau}$  to the physical element  $\tau$ .

Entity	Dimension	Codimension
Vertex	0	$D$
Edge	1	$D - 1$
Face	2	$D - 2$
Cell	$D$	0

Table 5.1: Named entities of low topological dimension or codimension.

The fact that the mesh is generated from reference elements enables a systematic framework for the finite element method. By defining each element of the mesh as a transformation of a reference element (e.g., a standard triangle or tetrahedron), the same set of shape functions and numerical integration schemes can be used uniformly across the mesh. In the sequel we shall revisit the basic concepts of this.

### 5.3.2 A primer on finite elements

The finite element method is a numerical approach that solves problems by approximating them within a finite-dimensional function space. These function spaces are usually constructed using basis functions that are defined over a reference element. In line with the Ciarlet definition [92], a finite element is characterized by a triplet  $\{\tau, P, \Sigma\}$ , where  $\tau$  represents the physical element,  $P$  denotes the polynomial space, and  $\Sigma$  is a collection of functionals.

**Definition 5.4 (Finite element).** A *finite element* consists of a triplet  $\{\tau, P, \Sigma\}$  where:

- $\tau$  is a compact, connected, Lipschitz subset of  $\mathbb{R}^d$  with non-empty interior.
- $P$  is a vector space of functions  $p : \tau \rightarrow \mathbb{R}^n$  for some positive integer  $n$  (typically  $n = 1$  or  $n = d$ ).
- $\Sigma$  is a set of  $k \in \mathbb{N}$  linear forms  $\{\sigma_1, \dots, \sigma_k\}$  acting on the elements of  $P$ , and such that the linear mapping

$$P \ni p \mapsto (\sigma_1(p), \dots, \sigma_k(p)) \in \mathbb{R}^k,$$

is bijective. The linear forms  $\{\sigma_1, \dots, \sigma_k\}$  are called the **local degrees of freedom**.

**Proposition 5.1.** There exists a basis  $\{\varphi_1, \dots, \varphi_k\}$  in  $P$  such that

$$\sigma_i(\varphi_j) = \delta_{ij}, \quad 1 \leq i, j \leq k.$$

In this case, the functions  $\{\varphi_1, \dots, \varphi_k\}$  are called the **local shape functions**.

The most classical example of a finite element is arguably the Lagrange finite element. These are also called nodal finite elements, since they're values are specified on the nodes of the polytope  $\tau$ .

**Example 5.1 (Lagrange finite element).** Let  $\{\tau, P, \Sigma\}$  be a finite element. If there is a set of points  $\{a_1, \dots, a_k\}$  in  $\tau$  such that, for all  $\varphi \in P$ ,  $\sigma_i(\varphi) = \varphi(a_i)$ ,  $1 \leq i \leq k$ ,  $\{\tau, P, \Sigma\}$  is called a **Lagrange finite element**. The points  $\{a_1, \dots, a_k\}$  are called the **nodes of the finite element**, and the local shape functions  $\{\varphi_1, \dots, \varphi_k\}$  (which are such that  $\varphi_i(a_j) = \delta_{ij}$  for  $1 \leq i, j \leq k$ ) are called the **nodal basis of  $P$** .

Finite elements are of interest to us mostly because they allow for the local interpolation of values within the polytope  $\tau$ . The following definition is of great use when working with the representations of functions that are defined on some Banach space  $V(\tau)$  of functions defined over the polytope  $\tau$ .

**Definition 5.5 (Local interpolation operator).** Let  $\{\tau, P, \Sigma\}$  be a fixed finite element. Denote by  $\{\sigma_1, \dots, \sigma_k\}$  the local degrees of freedom and  $\{\varphi_1, \dots, \varphi_k\}$  the local ( $\mathbb{R}^m$ -valued) shape functions. Then we define  $I_\tau$  as the **local interpolation operator** given by:

$$I_\tau(v) := \sum_{i=1}^k \sigma_i(v) \varphi_i \in P.$$

This previous definition gives a formula for representing any function of  $V(\tau)$  in terms of the local basis function and the degrees of freedom. For the most part these definitions are local in nature, meaning that they are all defined within the context of one element. Next, we shall see how they can be used to generate global spaces over the whole mesh  $\mathcal{T}$ .

To this end, given a mesh  $\mathcal{T}$ , for each  $\tau \in \mathcal{T}$ , the first step is to define the corresponding Banach space  $V(\tau)$  of  $\mathbb{R}^n$ -valued functions, along with a linear bijective mapping

$$\psi_\tau : V(\tau) \longrightarrow V(\hat{\tau}),$$

where  $V(\hat{\tau})$  is a Banach space for associated reference elements  $\hat{\tau}$ .

**Proposition 5.2 (Finite element generation).** *For every  $\tau \in \mathcal{T}$ , assume that it is equipped with a polytope transformation  $T_\tau$ . Let  $\{\hat{\tau}, \hat{P}, \hat{\Sigma}\}$  be a finite element. Then, the triplet  $\{\tau, P_\tau, \Sigma_\tau\}$  defined by:*

$$\begin{cases} \tau = T_\tau(\hat{\tau}), \\ P_\tau = \{\psi_\tau^{-1}(\hat{p}); \hat{p} \in \hat{P}\}, \\ \Sigma_\tau = \{\{\sigma_{\tau,i}\}_{1 \leq i \leq k}; \sigma_{\tau,i}(p) = \hat{\sigma}_i(\psi_\tau(p)), \forall p \in P_\tau\}, \end{cases}$$

*is also a finite element. The local shape functions are  $\varphi_{\tau,i} = \psi_\tau^{-1}(\hat{\varphi}_i)$ ,  $1 \leq i \leq k$ , and the associated local interpolation operator is*

$$I_\tau : V(\tau) \ni v \longmapsto I_\tau v = \sum_{i=1}^k \sigma_{\tau,i}(v) \varphi_{\tau,i} \in P_\tau.$$

*We refer to  $\hat{\tau}$  as the **reference finite element** and  $\tau$  as the **physical finite element**.*

A particular case is the Lagrange finite element generated by the pullback of the transformation  $T_\tau$ .

**Example 5.2.** *Let  $\{\hat{\tau}, \hat{P}, \hat{\Sigma}\}$  be a Lagrange finite element. One can define  $V(\hat{\tau}) = C^0(\hat{\tau}; \mathbb{R}^n)$ , and similarly for  $V(\tau)$ . By introducing the mapping*

$$\psi_\tau : V(\tau) \ni v \longmapsto \psi_\tau(v) = v \circ T_\tau \in V(\hat{\tau}),$$

*a linear bijection is obtained. Consequently, for all  $\tau \in \mathcal{T}$ , the finite element  $\{\tau, P_\tau, \Sigma_\tau\}$  is a Lagrange finite element. Specifically, we have:*

$$\sigma_i(v) = \hat{\sigma}_i(\psi_\tau(v)) = \psi_\tau(v)(\hat{a}_i) = v \circ T_\tau(\hat{a}_i),$$

*and by defining  $a_{\tau,i} = T_\tau(\hat{a}_i)$  for  $1 \leq i \leq k$ , we identify  $\{a_{\tau,i}\}_{1 \leq i \leq k}$  as the nodes of  $\{\tau, P_\tau, \Sigma_\tau\}$ .*

Furthermore, one can define a global interpolation operator to be able to represent functions on the whole mesh  $\mathcal{T}$ .

**Definition 5.6 (Global interpolation operator).** *Let  $\{\tau, P_\tau, \Sigma_\tau\}_{\tau \in \mathcal{T}}$  be a  $\mathcal{T}$ -based family of finite elements. The **global interpolation operator**  $I_h$  is defined by:*

$$I_h : \text{Dom}(I_h) \ni v \longmapsto \sum_{\tau \in \mathcal{T}_h} \sum_{i=1}^k \sigma_{\tau,i}(v|_\tau) \varphi_{\tau,i} \in W_h,$$

*where its domain  $\text{Dom}(I_h)$  is given by:*

$$\text{Dom}(I_h) = \{v \in [L^1(\Omega; \mathbb{R}^n)], \forall \tau \in \mathcal{T}, v|_\tau \in V(\tau)\},$$

*and  $W_h$  is the codomain of  $I_h$ , defined by:*

$$W_h = \{v_h \in [L^1(\Omega_h)]^m; \forall \tau \in \mathcal{T}_h, v|_\tau \in P_\tau\}.$$

*The space  $W_h$  is called an **approximation space**. We have abused the notation by implicitly extending  $\varphi_{\tau,i}$  by zero outside  $\tau$ .*

**Definition 5.7 (Conforming approximation).** Let  $W_h$  be defined as previously and let  $V$  be a Banach space.  $W_h$  is said to be  $V$ -conforming if  $W_h \subset V$ .

### 5.3.3 From weak formulations to linear algebra

To set ideas on how to use the previous concepts, we shall explain the case of assembling a linear system for the Poisson equation:

$$\begin{cases} -\Delta u = f & \text{in } \Omega, \\ u = 0 & \text{on } \partial\Omega, \end{cases} \quad (5.1)$$

where  $f \in L^2(\Omega)$ . We show how to incorporate the local interpolation operator into the discretization process and explain how to build element matrices, which are then assembled into the global stiffness matrix.

We start with the weak formulation of the Poisson equation by multiplying by a test function  $v \in H_0^1(\Omega)$ , integrating over the domain  $\Omega$ , and applying integration by parts. This gives:

$$\forall v \in H_0^1(\Omega), \quad \int_{\Omega} \nabla u \cdot \nabla v \, dx = \int_{\Omega} f v \, dx, \quad (5.2)$$

which is known as the weak formulation of (5.1).

To approximate the solution, we discretize the domain by using a finite element mesh  $\mathcal{T}$ . Each element  $\tau \in \mathcal{T}$  is mapped from a reference element  $\hat{\tau}$  via the transformation  $T_{\tau} : \hat{\tau} \rightarrow \tau$ . We assume that we are endowed with a conforming finite element space  $V_h \subset V$ , which is spanned by the local basis functions  $\{\varphi_{\tau,i}\}_{i=1}^k$  on each element  $\tau$ . Firstly, we decompose the integrals in (5.2), to write:

$$\forall v \in H_0^1(\Omega), \quad \sum_{\tau \in \mathcal{T}} \int_{\tau} \nabla u \cdot \nabla v \, dx = \sum_{\tau \in \mathcal{T}} \int_{\tau} f v \, dx. \quad (5.3)$$

Utilizing the local interpolation operator, the global approximate solution  $u_h$  on each  $\tau$  is represented as:

$$u_h|_{\tau} = \sum_{j=1}^k u_{\tau,j} \varphi_{\tau,j}.$$

In this manner, for each element  $\tau$ , the weak form is restricted to  $\tau$  and test functions  $v = \varphi_{\tau,i}$ , leading us to write the following:

$$\sum_{\tau \in \mathcal{T}} \sum_{j=1}^k u_{\tau,j} A_{ij}^{\tau} = \sum_{\tau \in \mathcal{T}} \sum_{j=1}^k u_{\tau,j} b_j^{\tau}, \quad 1 \leq i \leq k,$$

where the so called element matrix  $\mathbf{A}^{\tau}$  is defined by:

$$A_{ij}^{\tau} = \int_{\tau} \nabla \varphi_{\tau,j} \cdot \nabla \varphi_{\tau,i} \, dx.$$

Similarly, the element vector  $\mathbf{b}^{\tau}$  is:

$$b_i^{\tau} = \int_{\tau} f \varphi_{\tau,i} \, dx.$$

After computing the element matrices  $\mathbf{A}^{\tau}$  and vectors  $\mathbf{b}^{\tau}$  for each element, we assemble them into the global stiffness matrix  $\mathbf{A}$  and global load vector  $\mathbf{b}$ . This is done by summing the contributions of each element matrix and load vector into their corresponding locations in the global system:

$$\mathbf{A} = \sum_{\tau \in \mathcal{T}} \mathbf{A}^{\tau}, \quad \mathbf{b} = \sum_{\tau \in \mathcal{T}} \mathbf{b}^{\tau}.$$

The global system of equations is then:

$$\mathbf{A}\mathbf{u} = \mathbf{b},$$

where  $\mathbf{A}$  is the global stiffness matrix,  $\mathbf{u}$  is the vector of unknown coefficients,  $\mathbf{b}$  is the global mass vector. This global system can be solved for the coefficients  $\mathbf{u}$ , which represent the solution in terms of the finite element basis functions.

## 5.4 DSL embedding via template metaprogramming

With the introduction of advanced metaprogramming tools in recent versions of C++, it has become increasingly feasible to embed domain-specific languages (DSLs) tailored for finite element formulations directly within the language. This is primarily enabled by the improved metaprogramming features and cleaner syntax introduced in C++20, allowing developers to create expressive, efficient, and flexible abstractions that closely mirror the mathematical notation used in finite element methods.

The mathematical concepts discussed earlier can be effectively represented in a C++ library through metaprogramming. In C++, metaprogramming facilitates the development of libraries where the code naturally reflects the mathematical expressions of partial differential equations (PDEs). By leveraging templates, ‘constexpr’, and operator overloading, developers can define custom types representing vectors, matrices, and differential operators in a way that closely aligns with their mathematical equivalents. This not only improves code readability and maintainability but also enables powerful compile-time optimizations. These optimizations can enhance performance by eliminating unnecessary computations and selecting the most efficient algorithms for specific problems, making them especially beneficial when solving complex PDEs.

One of the primary advantages of implementing such a library in C++ is the significant performance boost over libraries written in interpreted languages like Python. C++ is a compiled language, meaning its code is directly translated into machine code that the CPU can execute, leading to much faster execution times compared to Python’s line-by-line interpretation. Although high-performance Python libraries exist—often relying on underlying C or C++ code—a direct C++ implementation avoids the overhead associated with the Python interpreter. This is particularly important for large-scale simulations and computationally intensive tasks, where performance is critical.

Rather than detailing the various classes of the library, which are numerous and already documented at the link provided at the start of this chapter, the following sections will present examples to illustrate the library’s ease of use and key functionalities.

## 5.5 Code examples and illustrations

In this section, we will provide detailed code examples and illustrations demonstrating the practical implementation of the concepts previously discussed. These examples will include the Poisson equation, the Helmholtz equation, the case linear elasticity, a 2D shape optimization example. The examples will guide you through the numerical implementation of each problem, from setting up the mesh and finite element spaces to solving the system and applying boundary conditions.

### 5.5.1 The Poisson equation

In this tutorial, we walk through the implementation of the Poisson equation. We will focus on the weak formulation of the Poisson equation in (5.1) and how this translates into a computational problem using the Rodin library. Each code component is explained alongside the corresponding mathematical concept. In the finite element method, we first derive the weak formulation of the problem. This has already been done in (5.2). This weak form is the foundation for our computational approach. We will now translate this into code using the Rodin framework.

The following C++ code implements the weak formulation of the Poisson equation using FEM. The Rodin library provides functionalities to handle meshes, define variational forms, and solve linear systems.

---

Listing 1: `examples/PDEs/Poisson.cpp`: C++ code for solving the Poisson equation.

---

```
#include <Rodin/Solver.h>
#include <Rodin/Geometry.h>
#include <Rodin/Variational.h>

using namespace Rodin;
using namespace Rodin::Geometry;
using namespace Rodin::Variational;

int main(int, char**)
```

---

```

{
  // Step 1: Create a mesh
  Mesh mesh;
  mesh = mesh.UniformGrid(Polytope::Type::Triangle, 16, 16);
  mesh.getConnectivity().compute(1, 2);

  // Step 2: Define finite element space
  P1 vh(mesh);

  // Step 3: Define trial and test functions
  TrialFunction u(vh);
  TestFunction v(vh);

  // Step 4: Set up the Poisson problem
  ScalarFunction f = 1;
  Problem poisson(u, v);
  poisson = Integral(Grad(u), Grad(v))
    - Integral(f, v)
    + DirichletBC(u, Zero());

  // Step 5: Solve the problem
  Solver::SparseLU(poisson).solve();

  // Step 6: Save the files for visualization
  mesh.save("Poisson.mesh");
  u.getSolution().save("Poisson.gf");

  return 0;
}

```

---

### Explanation of the code

Let us explain each piece of code in [Listing 1](#).

1. We discretize the domain into triangular elements using `UniformGrid`.

```

Mesh mesh;
mesh = mesh.UniformGrid(Polytope::Type::Triangle, 16, 16);
mesh.getConnectivity().compute(1, 2);

```

The domain is divided into a 16x16 grid of triangles, and the mesh connectivity is computed to handle relationships between elements.

2. We define the  $\mathbb{P}_1$  finite element space over the mesh.

```

P1 vh(mesh);

```

This space contains linear functions over each triangular element.

3. We define the trial and test functions of the bilinear and linear form of the weak formulation [\(5.2\)](#). The trial function  $u$  represents the unknown solution, while the test function  $v$  is used in the variational formulation:

```

TrialFunction u(vh);
TestFunction v(vh);

```

4. We define the weak form of the Poisson equation using integration by parts and apply homogeneous Dirichlet boundary conditions:

```

Problem poisson(u, v);
poisson = Integral(Grad(u), Grad(v))
         - Integral(f, v)
         + DirichletBC(u, Zero());

```

In Rodin, we use a FreeFem++ like approach, in which we assume that a variational problem is defined by some bilinear form  $a(u, v)$  and a linear form  $b(v)$ . Then, we consider that the weak formulation of the problem is:

$$\text{Find } u \in V_h \quad \text{s.t.} \quad \forall v \in V_h, \quad a(u, v) - b(v) = 0.$$

In this case the problem definition is given to us by the left-hand side of the expression above. Here in the example, we have the equivalence between code and mathematics:

$$\text{Integral(Grad(u), Grad(v))} \longleftrightarrow a(u, v) = \int_{\Omega} \nabla u \cdot \nabla v \, dx,$$

$$\text{Integral(f, v)} \longleftrightarrow b(v) = \int_{\Omega} f v \, dx$$

To impose the Dirichlet boundary condition, the library provides the `DirichletBC` class, which is used in the `Problem` definition. Then, the expression `DirichletBC(u, Zero())` enforces  $u = 0$  on the boundary.

5. We solve the system using the sparse LU decomposition.

```
Solver::SparseLU(poisson).solve();
```

6. We save the files for visualization:

```
mesh.save("Poisson.mesh");
u.getSolution().save("Poisson.gf");
```

Rodin uses the MFEM format [32] to write the meshes and functions. One can then visualize these using the GLVis program [211]. The result of the previous code example is visible in Fig. 5.3.

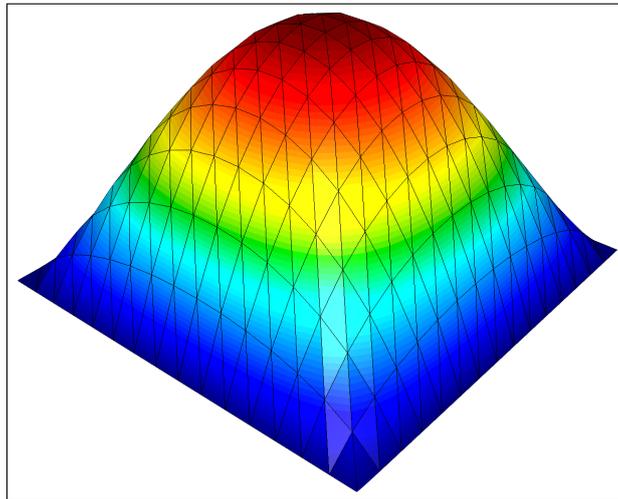


Figure 5.3: Visualization of the solution found by Listing 1.

### 5.5.2 The Helmholtz equation

In this tutorial, we will implement the Helmholtz equation using the Finite Element Method in C++. The Helmholtz equation is commonly used to describe wave propagation phenomena in acoustics, electromagnetics, and quantum mechanics. The equation is given by:

$$-\Delta u - k^2 u = f \quad \text{in } \Omega$$

with boundary conditions, where  $\Delta$  is the Laplace operator,  $u$  is the complex-valued wave function,  $k$  is the wave number, and  $f$  is a given source function. The solution to the Helmholtz equation involves complex numbers because it models oscillatory wave behavior. Rodin natively supports the use of complex numbers, which makes the implementation of the Helmholtz equation straightforward. The Helmholtz equation in variational (weak) form is derived by multiplying the equation by a test function  $v$  and integrating by parts. The weak formulation is:

$$\int_{\Omega} \nabla u \cdot \nabla v \, dx - k^2 \int_{\Omega} uv \, dx = \int_{\Omega} fv \, dx \quad \forall v \in H_0^1(\Omega)$$

where  $\nabla u \cdot \nabla v$  represents the gradient of the trial and test functions, and the second term includes the wave number  $k$ . This form is suitable for discretization using FEM.

Listing 2: `examples/PDEs/Helmholtz.cpp`: C++ code for solving the Poisson equation.

```
#include <Rodin/Solver.h>
#include <Rodin/Geometry.h>
#include <Rodin/Variational.h>

using namespace Rodin;
using namespace Rodin::Solver;
using namespace Rodin::Geometry;
using namespace Rodin::Variational;

static constexpr Real waveLength = 0.5;
static constexpr Real pi = Math::Constants::pi();
const Real waveNumber = 2 * Math::Constants::pi() / waveLength;

int main(int, char**)
{
    // Build a mesh
    Mesh mesh;
    mesh = mesh.UniformGrid(Polytope::Type::Triangle, { 64, 64 });
    mesh.getConnectivity().compute(1, 2);
    mesh.scale(1.0 / 63.0);

    // Functions
    P1<Complex> vh(mesh);

    ComplexFunction f =
        [&](const Point& p)
        {
            return Complex(1, 1) * waveNumber * waveNumber * cos(waveNumber * p.x()
                ) * cos(waveNumber * p.y());
        };

    TrialFunction u(vh);
    TestFunction v(vh);

    Problem helmholtz(u, v);
    helmholtz = Integral(Grad(u), Grad(v))
        - waveNumber * waveNumber * Integral(u, v)
        - Integral(f, v)
        ;

    CG(helmholtz).solve();

    GridFunction uRe(rh);
    GridFunction uIm(rh);
```

```

uRe = Re(u.getSolution());
uIm = Im(u.getSolution());

// Save solution
uRe.save("uRe.gf");
uIm.save("uIm.gf");
mesh.save("Grid.mesh");

return 0;
}

```

### Explanation of the code

Let us explain each piece of code in [Listing 2](#).

1. We start by building a triangular mesh that discretizes the domain.

```

Mesh mesh;
mesh = mesh.UniformGrid(Polytope::Type::Triangle, { 64, 64 });
mesh.getConnectivity().compute(1, 2);
mesh.scale(1.0 / 63.0);

```

This creates a 64x64 grid of triangles. The mesh is scaled to fit within the desired domain.

2. The wave number  $k$  is calculated based on the wavelength  $\lambda$ .

```

static constexpr Real waveLength = 0.5;
const Real waveNumber = 2 * Math::Constants::pi() / waveLength;

```

The wave number determines the frequency of oscillation.

3. We define the finite element space  $\mathbb{P}_1$  for complex-valued functions.

```

P1<Complex> vh(mesh);

```

4. The source function  $f$  generates an oscillating wave over the domain.

```

ComplexFunction f =
    [&](const Point& p)
    {
        return Complex(p.x(), 1) * waveNumber * waveNumber
            * cos(waveNumber * p.x()) * cos(waveNumber * p.y());
    };

```

5. We define the unknown wave function  $u$  and the test function  $v$ .

```

TrialFunction u(vh);
TestFunction v(vh);

```

6. The variational form of the Helmholtz equation is constructed.

```

Problem helmholtz(u, v);
helmholtz = Integral(Grad(u), Grad(v))
    - waveNumber * waveNumber * Integral(u, v)
    - Integral(f, v);

```

This corresponds to the weak form of the Helmholtz equation.

7. We solve the system using the Conjugate Gradient (CG) method.

```

CG(helmholtz).solve();

```

8. Finally, we extract the real and imaginary parts of the solution and save them.

```

GridFunction uRe(rh);
GridFunction uIm(rh);

uRe = Re(u.getSolution());
uIm = Im(u.getSolution());

uRe.save("uRe.gf");
uIm.save("uIm.gf");
mesh.save("Grid.mesh");

```

The results of this code can be visualized in [Fig. 5.4](#).

### 5.5.3 The linear elasticity equation

The linear elasticity equation describes how materials deform under applied forces. It is governed by Hooke's Law and is used to compute the displacement field of an elastic body. The equation is expressed as:

$$-\nabla \cdot \sigma = f \quad \text{in } \Omega$$

where  $\sigma$  is the stress tensor, related to the strain tensor  $\varepsilon(u)$  via the material's constitutive law (Hooke's Law),  $f$  is the body force (e.g., gravitational or external forces) and  $u$  is the displacement vector field, which we aim to compute.

In linear elasticity, the stress tensor  $\sigma$  is related to the strain tensor  $\varepsilon(u)$  by the Lamé coefficients  $\lambda$  and  $\mu$ :

$$\sigma(u) = \lambda \operatorname{tr}(\varepsilon(u))I + 2\mu\varepsilon(u)$$

where:  $\lambda$  and  $\mu$  are the Lamé coefficients, material-dependent constants. Additionally,  $\varepsilon(u) = \frac{1}{2}(\nabla u + \nabla u^T)$  is the strain tensor. The weak form of the linear elasticity equation can be written as:

$$\int_{\Omega} \sigma(u) : \varepsilon(v) \, dx = \int_{\Omega} f \cdot v \, dx + \int_{\Gamma_N} g \cdot v \, dx$$

Here,  $v$  is the test function,  $g$  is the Neumann boundary condition (traction force), and  $\Gamma_N$  is the part of the boundary where Neumann conditions are applied. Dirichlet boundary conditions are enforced on  $\Gamma_D$ .

The following code demonstrates the implementation of the linear elasticity equation using the Rodin library. We solve for the displacement field  $u$  of a material subjected to a downward pulling force. For simplicity we assume that  $f = 0$ .

---

Listing 3: `examples/PDEs/Elasticity.cpp`: C++ code for solving the linear elasticity equation.

---

```

#include <Rodin/Solver.h>
#include <Rodin/Geometry.h>
#include <Rodin/Variational.h>
#include <Rodin/Variational/LinearElasticity.h>

using namespace Rodin;
using namespace Rodin::Geometry;
using namespace Rodin::Variational;

int main(int argc, char** argv)
{
    const char* meshFile = "../resources/examples/PDEs/Elasticity.mfem.mesh";

    // Define boundary attributes
    Attribute Gamma = 1, GammaD = 2, GammaN = 3, Gamma0 = 4;

    // Load mesh
    Mesh mesh;
    mesh.load(meshFile);
    mesh.getConnectivity().compute(1, 2);

```

```

// Functions
size_t d = mesh.getSpaceDimension();
P1 fes(mesh, d);

// Lamé coefficients
const Real lambda = 0.5769, mu = 0.3846;

// Pull force
VectorFunction g{0, -1};

// Define problem
TrialFunction u(fes);
TestFunction v(fes);

Problem elasticity(u, v);
elasticity = LinearElasticityIntegral(u, v)(lambda, mu)
    - BoundaryIntegral(g, v).over(GammaN)
    + DirichletBC(u, VectorFunction{0, 0}).on(GammaD);

Solver::CG(elasticity).solve();

// Save solution
u.getSolution().save("Elasticity.gf");
mesh.save("Elasticity.mesh");

return 0;
}

```

### Explanation of the code

1. The following headers bring in the necessary functionalities from the Rodin library for solving the elasticity problem:

```

#include <Rodin/Solver.h>
#include <Rodin/Geometry.h>
#include <Rodin/Variational.h>
#include <Rodin/Variational/LinearElasticity.h>

```

2. We define boundary attributes to apply Dirichlet and Neumann conditions and load the mesh from a file.

```

Attribute Gamma = 1, GammaD = 2, GammaN = 3, Gamma0 = 4;
Mesh mesh;
mesh.load(meshFile);
mesh.getConnectivity().compute(1, 2);

```

Here attribute 2 corresponds to  $\Gamma_D$  and attribute 3 to  $\Gamma_N$ .

3. We define a finite element space  $\mathbb{P}_1$  for the displacement vector field:

```

size_t d = mesh.getSpaceDimension();
P1 fes(mesh, d);

```

4. The material properties, such as the Lamé coefficients  $\lambda$  and  $\mu$ , define the relationship between stress and strain in the material.

```

const Real lambda = 0.5769, mu = 0.3846;

```

5. We define a downward force (e.g., gravity) acting on the material.

```

VectorFunction f{0, -1};

```

6. We define the trial and test functions, set up the variational form of the elasticity equation, and apply the boundary conditions:

```
TrialFunction u(fes);
TestFunction v(fes);

Problem elasticity(u, v);
elasticity = LinearElasticityIntegral(u, v)(lambda, mu)
            - BoundaryIntegral(f, v).over(GammaN)
            + DirichletBC(u, VectorFunction{0, 0}).on(GammaD);
```

7. The Conjugate Gradient solver is used to solve the linear system:

```
Solver::CG(elasticity).solve();
```

8. Finally, we save the computed displacement field and mesh for post-processing:

```
u.getSolution().save("Elasticity.gf");
mesh.save("Elasticity.mesh");
```

The solution output by the code is presented in [Fig. 5.5](#).

#### 5.5.4 The level-set cantilever optimization in 2D

This tutorial covers the implementation of the classical 2D level-set cantilever example, which is described in [Section 1.6](#) of [Chapter 1](#), using the finite element method and the Rodin library. The goal is to minimize the compliance (maximize stiffness) of a cantilever structure subjected to external loads, while optimizing its shape using a level set method.

The level set method represents the domain implicitly using a function (signed distance function), which is iteratively updated to track the evolving boundary of the structure during the optimization process.

Below is the C++ code implementing the level set cantilever optimization.

Listing 4: `examples/ShapeOptimization/LevelSetCantilever2D.cpp`: C++ code for performing the body-fitted shape optimization of a cantilever.

```
#include <Rodin/Solver.h>
#include <Rodin/Geometry.h>
#include <Rodin/Variational.h>
#include <Rodin/Variational/LinearElasticity.h>
#include <RodinExternal/MMG.h>

using namespace Rodin;
using namespace Rodin::Geometry;
using namespace Rodin::External;
using namespace Rodin::Variational;

using FES = VectorP1<Mesh<Context::Sequential>>;

// Define interior and exterior for level set discretization
static constexpr Attribute Interior = 1, Exterior = 2;

// Define boundary attributes
static constexpr Attribute Gamma0 = 1, GammaD = 2, GammaN = 3, Gamma = 4;

// Lamé coefficients
static constexpr double mu = 0.3846;
static constexpr double lambda = 0.5769;
```

```

// Optimization parameters
static constexpr size_t maxIt = 300;
static constexpr double hmax = 0.05;
static constexpr double hmin = 0.1 * hmax;
static constexpr double hausd = 0.5 * hmin;
static constexpr double ell = 0.4;
const constexpr Real dt = 4 * (hmax - hmin);
static constexpr double alpha = dt;

// Compliance
inline Real compliance(const GridFunction<FES>& w)
{
    auto& vh = w.getFiniteElementSpace();
    TrialFunction u(vh);
    TestFunction v(vh);
    BilinearForm bf(u, v);
    bf = LinearElasticityIntegral(u, v)(lambda, mu);
    return bf(w, w);
};

int main(int, char**)
{
    const char* meshFile = "../resources/examples/ShapeOptimization/LevelSetCantilever2D.mfem.mesh";

    // Load mesh
    MMG::Mesh th;
    th.load(meshFile);
    MMG::Optimizer().setHMax(hmax).setHMin(hmin).optimize(th);

    // Optimization loop
    std::vector<double> obj;
    std::ofstream fObj("obj.txt");
    for (size_t i = 0; i < maxIt; i++)
    {
        th.getConnectivity().compute(1, 2);

        Alert::Info() << "----- Iteration: " << i << Alert::Raise;

        Alert::Info() << " | Trimming mesh." << Alert::Raise;
        SubMesh trimmed = th.trim(Exterior);
        trimmed.save("Omega.mesh");

        Alert::Info() << " | Building finite element spaces." << Alert::Raise;
        const size_t d = th.getSpaceDimension();
        P1 sh(th);
        P1 vh(th, d);

        P1 shInt(trimmed);
        P1 vhInt(trimmed, d);

        Alert::Info() << " | Solving state equation." << Alert::Raise;
        auto f = VectorFunction{0, -1};
        TrialFunction u(vhInt);
        TestFunction v(vhInt);

        // Elasticity equation
        Problem elasticity(u, v);
        elasticity = LinearElasticityIntegral(u, v)(lambda, mu)

```

---

```

        - BoundaryIntegral(f, v).over(GammaN)
        + DirichletBC(u, VectorFunction{0, 0}).on(GammaD);
Solver::CG(elasticity).solve();

Alert::Info() << "    | Computing shape gradient." << Alert::Raise;
auto jac = Jacobian(u.getSolution());
jac.traceOf(Interior);
auto e = 0.5 * (jac + jac.T());
auto Ae = 2.0 * mu * e + lambda * Trace(e) * IdentityMatrix(d);
auto n = FaceNormal(th);
n.traceOf(Interior);

// Hilbert extension-regularization procedure
TrialFunction g(vh);
TestFunction w(vh);
Problem hilbert(g, w);
hilbert = Integral(alpha * alpha * Jacobian(g), Jacobian(w))
        + Integral(g, w)
        - FaceIntegral(Dot(Ae, e) - ell, Dot(n, w)).over(Gamma)
        + DirichletBC(g, VectorFunction{0, 0, 0}).on(GammaN);
Solver::CG(hilbert).solve();
auto& dJ = g.getSolution();
dJ.save("dJ.gf");
vh.getMesh().save("dJ.mesh");

// Update objective
double objective = compliance(u.getSolution()) + ell * th.getVolume(Interior);
obj.push_back(objective);
fObj << objective << "\n";
fObj.flush();
Alert::Info() << "    | Objective: " << obj.back() << Alert::Raise;
Alert::Info() << "    | Distancing domain." << Alert::Raise;

P1 dh(th);
auto dist = MMG::Distancer(dh).setInteriorDomain(Interior)
        .distance(th);

// Advect the level set function
Alert::Info() << "    | Advecting the distance function." << Alert::Raise;
GridFunction norm(sh);
norm = Frobenius(dJ);
dJ /= norm.max();

MMG::Advect(dist, dJ).step(dt);

// Recover the implicit domain
Alert::Info() << "    | Meshing the domain." << Alert::Raise;

th = MMG::ImplicitDomainMesher().split(Interior, {Interior, Exterior})
        .split(Exterior, {Interior, Exterior})
        .setRMC(1e-6)
        .setHMax(hmax)
        .setHMin(hmin)
        .setHausdorff(hausd)
        .setAngleDetection(false)
        .setBoundaryReference(Gamma)
        .setBaseReferences(GammaD)
        .discretize(dist);

```

---

```

    MMG::Optimizer().setHMax(hmax)
                    .setHMin(hmin)
                    .setHausdorff(hausd)
                    .setAngleDetection(false)
                    .optimize(th);
}

Alert::Info() << "Saved final mesh to Omega.mesh" << Alert::Raise;

return 0;
}

```

### Explanation of the code

Let us explain the main parts of the code in [Listing 4](#).

1. First, we include all the necessary functionality, along with setting up the namespaces.

```

#include <Rodin/Solver.h>
#include <Rodin/Geometry.h>
#include <Rodin/Variational.h>
#include <Rodin/Variational/LinearElasticity.h>
#include <RodinExternal/MMG.h>

using namespace Rodin;
using namespace Rodin::Geometry;
using namespace Rodin::External;
using namespace Rodin::Variational;

using FES = VectorP1<Mesh<Context::Sequential>>;

```

2. We define the system parameters and attributes. Here, we assume that the mesh has two materials which represent the interior and exterior of the structure.

```

// Define interior and exterior for level set discretization
static constexpr Attribute Interior = 1, Exterior = 2;

// Define boundary attributes
static constexpr Attribute Gamma0 = 1, GammaD = 2, GammaN = 3, Gamma = 4;

// Lamé coefficients
static constexpr double mu = 0.3846;
static constexpr double lambda = 0.5769;

// Optimization parameters
static constexpr size_t maxIt = 300;
static constexpr double hmax = 0.05;
static constexpr double hmin = 0.1 * hmax;
static constexpr double hausd = 0.5 * hmin;
static constexpr double ell = 0.4;
const constexpr Real dt = 4 * (hmax - hmin);
static constexpr double alpha = dt;

```

3. We define the function to compute the compliance.

```

// Compliance
inline Real compliance(const GridFunction<FES>& w)
{
    auto& vh = w.getFiniteElementSpace();
    TrialFunction u(vh);
}

```

```

    TestFunction v(vh);
    BilinearForm bf(u, v);
    bf = LinearElasticityIntegral(u, v)(lambda, mu);
    return bf(w, w);
};

```

4. We load the initial mesh from a file and optimize its quality using the MMG optimizer.

```

MMG::Mesh th;
th.load(meshFile);
MMG::Optimizer()./* Set parameters */.optimize(th);
th.save("Omega0.mesh", IO::FileFormat::MEDIT);

```

5. The main loop iterates over the maximum number of iterations (`maxIt`) to trim the exterior of the domain, solve the elasticity equation, and compute the shape gradient.

```

for (size_t i = 0; i < maxIt; i++)
{
// Main loop
}

```

6. The mesh is trimmed to build finite element spaces on the whole computational domain, and on the actual shape itself.

```

SubMesh trimmed = th.trim(Exterior);
trimmed.save("Omega.mesh");

const size_t d = th.getSpaceDimension();
P1 sh(th);
P1 vh(th, d);

P1 shInt(trimmed);
P1 vhInt(trimmed, d);

```

7. The state equation (linear elasticity) is solved at each iteration, over the subspace, to compute the displacement field under the applied forces.

```

Problem elasticity(u, v);
elasticity = LinearElasticityIntegral(u, v)(lambda, mu)
    - BoundaryIntegral(f, v).over(GammaN)
    + DirichletBC(u, VectorFunction{0, 0}).on(GammaD);
Solver::CG(elasticity).solve();

```

8. We compute the shape gradient, which drives the shape optimization process, and apply a Hilbert regularization for stability.

```

auto jac = Jacobian(u.getSolution());
jac.traceOf(Interior);
auto e = 0.5 * (jac + jac.T());
auto Ae = 2.0 * mu * e + lambda * Trace(e) * IdentityMatrix(d);
auto n = FaceNormal(th);
n.traceOf(Interior);

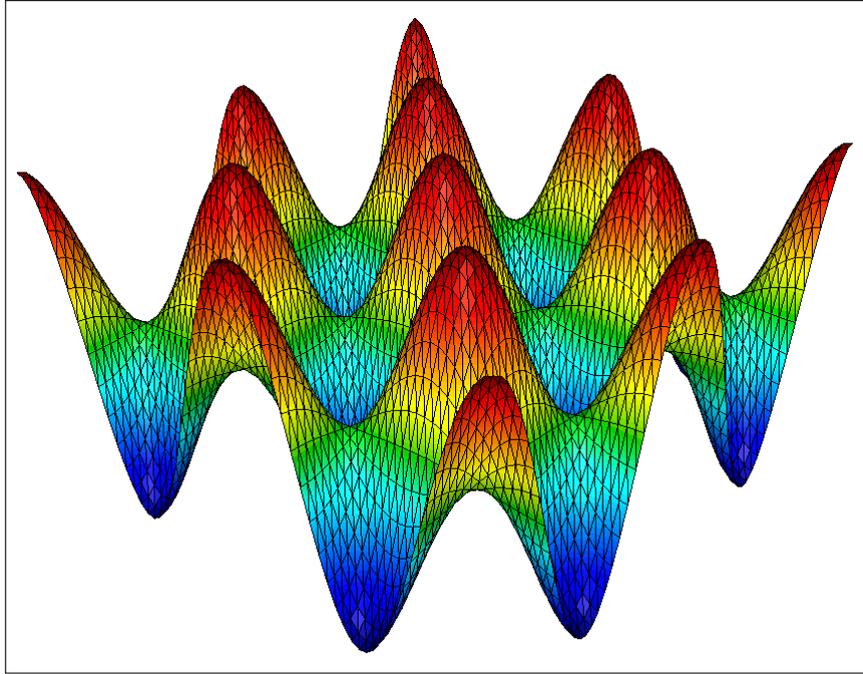
// Hilbert extension-regularization procedure
TrialFunction g(vh);
TestFunction w(vh);
Problem hilbert(g, w);
hilbert = Integral(alpha * alpha * Jacobian(g), Jacobian(w))
    + Integral(g, w)
    - FaceIntegral(Dot(Ae, e) - ell, Dot(n, w)).over(Gamma)
    + DirichletBC(g, VectorFunction{0, 0, 0}).on(GammaN);
Solver::CG(hilbert).solve();
auto& dJ = g.getSolution();

```

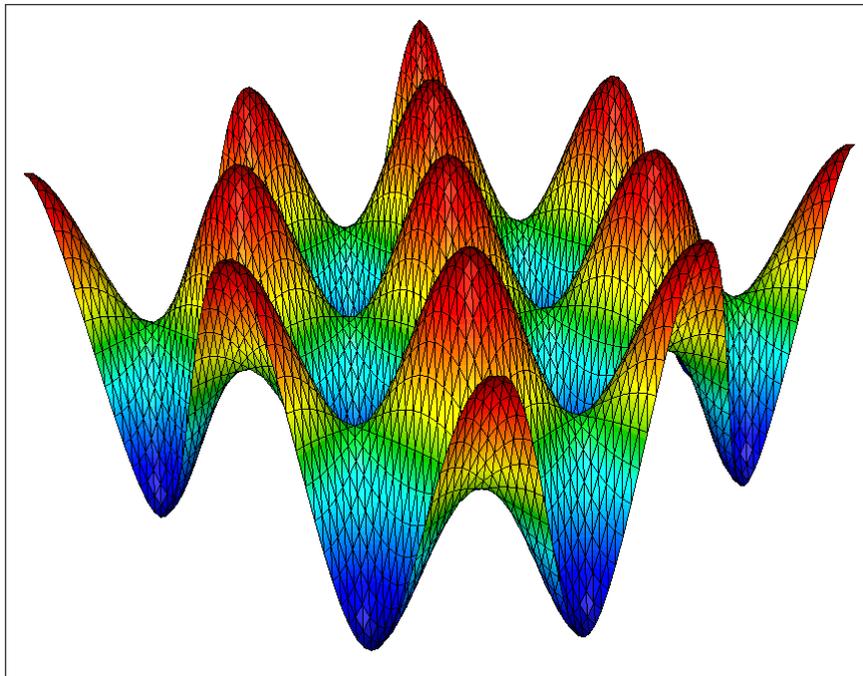
9. At the end of each iteration, the level set function is advected, and the implicit domain is remeshed for the next iteration.

```
MMG::Advect(dist, dJ).step(dt);  
th = MMG::ImplicitDomainMesher().split(Interior, {Interior, Exterior})  
    .setHMax(hmax)  
    .discretize(dist);
```

The are those presented in [Section 1.6](#).



(a) Real part of the solution yielded by Listing 2.



(b) Imaginary part of the solution yielded by Listing 2.

Figure 5.4: Visualization of the solution outputted in the Helmholtz example

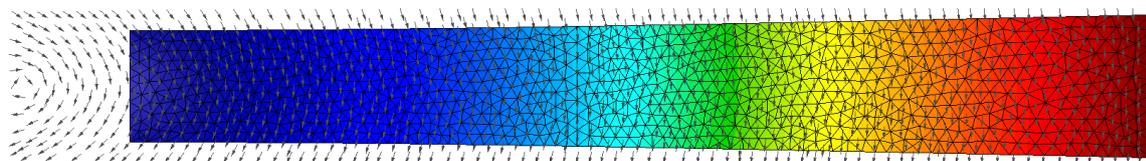


Figure 5.5: Linear elasticity solution outputted by [Listing 3](#)

# Conclusions and perspectives

---

In this final section, we reflect on the contributions made through the development of advanced shape and topology optimization techniques, particularly on surfaces, and outline key directions for future research. In [Chapter 1](#), we reviewed classical methods in shape and topology optimization, and in [Section 1.8](#), we compared previous work with our own contributions, highlighting a substantial theoretical and practical gap in the general field of shape optimization on surfaces.

To address this gap, in [Chapter 2](#), we developed a general framework for performing geometrical shape optimization of domains on Riemannian manifolds. In this chapter, we presented various results commonly used when computing shape derivatives, but due to time constraints, we were unable to prove every relevant result found in classical literature. This presents an area for future work. For example, exploring the relationship between shape derivatives and tubular neighborhoods offers promising avenues for deeper theoretical insights. Theoretical results such as [Theorem 2.4](#) and [Proposition 2.2](#) describe the structure of shape derivatives and show that they primarily capture local information near the shape. These results also reveal the connection with the signed distance function, suggesting the potential for a unified theory of geometrical shape optimization that integrates these recurring concepts. Future work could focus on establishing structure theorems for the shape derivative  $J'(\Sigma)(\theta)$  of submanifolds  $\Sigma$  of codimension greater than 1, where the vector field  $\theta$  is tangential to the ambient manifold  $M$ . Another promising direction involves fully applying this theory to shells, which can be regarded as curved Riemannian manifolds, providing the necessary tools for the intrinsic optimization of regions  $G \subset M$  on shells.

In [Chapter 3](#), we proposed an efficient numerical framework for tracking arbitrarily large evolutions of a region  $G(t)$  embedded within a fixed two- or three-dimensional surface  $S$ . Our method, [Algorithm 2](#), hinges on the combination of an explicit representation of  $G(t)$  using a high-quality mesh for precise geometric and mechanical computations, with the level set method to capture potentially dramatic motion. The core of this approach is a set of efficient numerical algorithms that allow seamless transitions between these representations. Beyond numerical validation, we demonstrated its effectiveness in two physical applications: the evolution of a fire front under a velocity field depending on its geometric features and those of the landscape surface  $S$  in [Section 3.5](#), and its use in [Chapter 4](#), particularly in [Section 4.2](#), where it was applied to optimize the region of the boundary  $S = \partial\Omega$  of a 3D domain  $\Omega$  with homogeneous Dirichlet boundary conditions in thermal mechanics. These developments open the door to numerous applications. For instance, building upon previous work [\[108\]](#), we extended the model example in [Section 4.2.4](#) to various mechanical contexts, including linearly elastic structures, where optimizing regions supporting boundary conditions aids in designing fixture systems. Future work in this area could apply these methods to problems such as precision agriculture [\[304, 294\]](#), where one tracks a patch of land obeying specific agricultural metrics, using terrain mesh modeling and crop evolution tracking. Another relevant application is vegetation tracking [\[193\]](#), where the evolution of vegetated areas is tracked alongside observable quantities such as moisture and density. Modern approaches to vegetation tracking could be used in restoration solutions [\[266\]](#). From a more mechanical perspective, and in line with the recent monograph [\[101\]](#), on the longer term, we wish to investigate the reinforcement and design of openings in shell structures, whose delicate mechanical equations could benefit from the datum of a high-quality, body-fitted computational meshes.

In [Chapter 4](#), building on the results from [Chapter 3](#) and the recent contribution [\[57\]](#), we introduced and applied the concept of topological derivatives specifically for the optimization of regions that support boundary conditions. Throughout [Sections 4.7, 4.11, 4.13](#) and [4.14](#), we provided various numerical

examples across different physical contexts, each offering further opportunities for enhancement. For example, the cathode-anode partitioning problem in [Section 4.11](#) could be improved by incorporating the full Navier-Stokes and magnetic field coupling model, yielding more accurate designs that better reflect the true physical systems. In the acoustic cloaking problem discussed in [Section 4.7](#), we encountered limitations related to mesh resolution and execution time. A finer mesh would enable better visualization of the scattered and total fields, but would introduce new challenges, such as increased computational time and memory constraints. The structural optimization problems in [Sections 4.13](#) and [4.14](#) could also be refined by introducing more realistic physical constraints and scenarios that pose real-world challenges. For example, the clamp-locator problem could be reformulated to focus on finding stable configurations for securing a workpiece in place, ultimately leading to the design of an “intelligent fixing system.” In this context, we refer to [\[221\]](#), which provides modern insights into such systems. More broadly, the formal techniques described in this chapter can be extended to derive topological sensitivities for other equations and boundary conditions. A particularly promising area of interest is fluid mechanics, where optimizing slip and no-slip boundary conditions (see [\[280\]](#) for definitions) could be applied to hydrophobic surfaces to improve the tribological performance of sliding bearings. This could enhance factors such as load-carrying capacity and friction reduction [\[346\]](#). Additionally, these techniques could be applied to the optimization of journal bearings [\[98\]](#), a type of bearing that supports a rotating shaft, allowing smooth rotation within a stationary housing.

Regarding our numerical library Rodin, we aim to refine the developed methods for practical, large-scale applications and extend them to address more complex, real-world challenges across various industries. Despite its numerical success and validation, Rodin is still in its early stages. Many API methods have already been defined, with base functionality documented and established, but the library still needs to scale effectively for broader use. While Rodin currently supports native multithreading, this is insufficient at larger scales. To fully utilize high-performance computing, we must implement data partitioning for key structures, such as meshes and finite element spaces, to enable their use in distributed environments like clusters. Integration with established mesh partitioning tools, such as METIS [\[200\]](#) and Scotch [\[263\]](#), is necessary for this scaling effort. We also plan to integrate Rodin with other high-performance computing libraries like PETSc [\[40\]](#) and MFEM [\[32\]](#) for the efficient assembly and solution of large distributed linear systems. Furthermore, incorporating modern open-source finite element libraries, such as Basix [\[288\]](#), will enable the use of a wide variety of finite elements, including those of arbitrary order, and streamline integration with Rodin’s infrastructure. The ultimate objective is to implement a full de Rham complex (see [\[35\]](#)), covering approximation spaces that include  $H^1$ , suitable for problems requiring square-integrable derivatives,  $H(\text{div})$ , which handles fields where the divergence operator plays a crucial role, and  $H(\text{curl})$ , important for vector fields with square-integrable curl, such as those in electromagnetism.  $L^2$ , representing square-integrable functions, also forms the foundational space within this structure. Due to Rodin’s software architecture, where complexity is abstracted behind a native high-level domain-specific language in C++, developers will have access to a user-friendly interface without needing to manage low-level details. This approach will enhance integration with other C/C++ tools, improving development speed and usability.

In conclusion, the main contribution of this work aligns with the broader objectives of research in shape optimization:

*To develop robust algorithms capable of handling extreme geometries and to create a unified framework that integrates shape, topology, and material optimization into one cohesive process.*

Through the theory, practice, and implementation presented in this manuscript, we believe that this work has contributed meaningfully toward this goal.

# Bibliography

---

- [1] R. ABGRALL, *Numerical discretization of the first-order hamilton-jacobi equation on triangular meshes*, Communications on pure and applied mathematics, 49 (1996), pp. 1339–1373.
- [2] M. ABRAMOWITZ AND I. A. STEGUN, *Handbook of mathematical functions: with formulas, graphs, and mathematical tables*, vol. 55, Courier Corporation, 1965.
- [3] S. ADRIAENSSENS, P. BLOCK, D. VEENENDAAL, AND C. WILLIAMS, *Shell structures for architecture: form finding and optimization*, Routledge, 2014.
- [4] A. P. AGRAWAL, S. ALI, AND S. RATHORE, *Finite element stress analysis for shape optimization of spur gear using ansys*, Materials Today: Proceedings, 64 (2022), pp. 1147–1152.
- [5] T. AKENINE-MOLLER, E. HAINES, AND N. HOFFMAN, *Real-time rendering*, AK Peters/crc Press, 2019.
- [6] J. ALEXANDERSEN, *A detailed introduction to density-based topology optimisation of fluid flow problems with implementation in matlab*, Structural and Multidisciplinary Optimization, 66 (2023), p. 12.
- [7] G. ALLAIRE, *Numerical analysis and optimization: an introduction to mathematical modelling and numerical simulation*, OUP Oxford, 2007.
- [8] G. ALLAIRE, *A brief introduction to homogenization and miscellaneous applications*, in ESAIM: Proceedings, vol. 37, EDP Sciences, 2012, pp. 1–49.
- [9] G. ALLAIRE, E. BONNETIER, G. FRANCFORT, AND F. JOUVE, *Shape optimization by the homogenization method*, Numerische Mathematik, 76 (1997), pp. 27–68.
- [10] G. ALLAIRE AND C. DAPOGNY, *A linearized approach to worst-case design in parametric and geometric shape optimization*, Mathematical Models and Methods in Applied Sciences, 24 (2014), pp. 2199–2257.
- [11] ———, *A deterministic approximation method in shape optimization under random uncertainties*, SMAI Journal of computational mathematics, 1 (2015), pp. 83–143.
- [12] G. ALLAIRE, C. DAPOGNY, AND P. FREY, *Topology and geometry optimization of elastic structures by exact deformation of simplicial mesh*, Comptes Rendus Mathematique, 349 (2011), pp. 999–1003.
- [13] ———, *A mesh evolution algorithm based on the level set method for geometry and topology optimization*, Structural and Multidisciplinary Optimization, 48 (2013), pp. 711–715.
- [14] ———, *Shape optimization with a level set based mesh evolution method*, Computer Methods in Applied Mechanics and Engineering, 282 (2014), pp. 22–53.
- [15] G. ALLAIRE, C. DAPOGNY, AND F. JOUVE, *Shape and topology optimization*, in Geometric partial differential equations, part II, A. Bonito and R. Nchetto eds., Handbook of Numerical Analysis, 22 (2021), pp. 1–132.

- [16] G. ALLAIRE, C. DAPOGNY, AND F. JOUVE, *Shape and topology optimization*, in Geometric partial differential equations, part II, A. Bonito and R. Nocketto eds., Handbook of Numerical Analysis, vol. 22, (2021), pp. 1–132.
- [17] G. ALLAIRE, F. DE GOURNAY, F. JOUVE, AND A.-M. TOADER, *Structural optimization using topological and shape sensitivity via a level set method*, Control and cybernetics, 34 (2005), p. 59.
- [18] G. ALLAIRE AND M. H. GFRERER, *Autofreefem: Automatic code generation with freefem++ and latex output for shape and topology optimization of non-linear multi-physics problems*, arXiv preprint arXiv:2407.11713, (2024).
- [19] G. ALLAIRE AND A. HENROT, *On some recent advances in shape optimization*, Comptes Rendus de l'Académie des Sciences-Series IIB-Mechanics, 329 (2001), pp. 383–396.
- [20] G. ALLAIRE AND F. JOUVE, *Coupling the level set method and the topological gradient in structural optimization*, in IUTAM Symposium on Topological Design Optimization of Structures, Machines and Materials: Status and Perspectives, Springer, 2006, pp. 3–12.
- [21] G. ALLAIRE, F. JOUVE, AND A.-M. TOADER, *A level-set method for shape optimization*, Comptes Rendus Mathématique, 334 (2002), pp. 1125–1130.
- [22] G. ALLAIRE AND M. SCHOENAUER, *Conception optimale de structures*, vol. 58, Springer, 2007.
- [23] M. ALNÆS, J. BLECHTA, J. HAKE, A. JOHANSSON, B. KEHLET, A. LOGG, C. RICHARDSON, J. RING, M. E. ROGNES, AND G. N. WELLS, *The fenics project version 1.5*, Archive of numerical software, 3 (2015).
- [24] M. S. ALNAES, A. LOGG, K. B. OLGAARD, M. E. ROGNES, AND G. N. WELLS, *Unified form language: A domain-specific language for weak formulations of partial differential equations*, ACM Transactions on Mathematical Software (TOMS), 40 (2014), pp. 1–37.
- [25] M. ALVES, *Frequency-domain models*, in Numerical Modelling of Wave Energy Converters, Elsevier, 2016, pp. 11–30.
- [26] L. AMBROSIO AND G. BUTTAZZO, *An optimal design problem with perimeter penalization*, Calculus of Variations and Partial Differential Equations, 1 (1993), pp. 55–69.
- [27] L. AMBROSIO AND N. DANCER, *Geometric evolution problems, distance function and viscosity solutions*, Springer, 2000.
- [28] M. AMBROZ, M. BALAŽOVJECH, M. MEDL'A, AND K. MIKULA, *Numerical modeling of wildland surface fire propagation by evolving surface curves*, Advances in Computational Mathematics, 45 (2019), pp. 1067–1103.
- [29] H. B. AMEUR, M. BURGER, AND B. HACKL, *Level set methods for geometric inverse problems in linear elasticity*, Inverse Problems, 20 (2004), p. 673.
- [30] H. AMMARI AND H. KANG, *Polarization and moment tensors: with applications to inverse problems and effective medium theory*, vol. 162, Springer Science & Business Media, 2007.
- [31] S. AMSTUTZ AND H. ANDRÄ, *A new algorithm for topology optimization using a level-set method*, Journal of Computational Physics, 216 (2006), pp. 573–588.
- [32] R. ANDERSON, J. ANDREJ, A. BARKER, J. BRAMWELL, J.-S. CAMIER, J. CERVENY, V. DOBREV, Y. DUDOUIT, A. FISHER, T. KOLEV, ET AL., *Mfem: A modular finite element methods library*, Computers & Mathematics with Applications, 81 (2021), pp. 42–74.
- [33] J. L. ANIBAL, C. A. MADER, AND J. R. MARTINS, *Aerodynamic shape optimization of an electric aircraft motor surface heat exchanger with conjugate heat transfer constraint*, International Journal of Heat and Mass Transfer, 189 (2022), p. 122689.
- [34] N. E. ANTOINE AND I. M. KROO, *Aircraft optimization for minimal environmental impact*, Journal of aircraft, 41 (2004), pp. 790–797.

- 
- [35] D. N. ARNOLD, R. S. FALK, AND R. WINTHER, *Differential complexes and stability of finite element methods i. the de rham complex*, in *Compatible spatial discretizations*, Springer, 2006, pp. 23–46.
- [36] B. AYUSO AND L. D. MARINI, *Discontinuous galerkin methods for advection-diffusion-reaction problems*, *SIAM Journal on Numerical Analysis*, 47 (2009), pp. 1391–1420.
- [37] T. J. BAKER, *Mesh movement and metamorphosis*, *Engineering with Computers*, 18 (2002), pp. 188–198.
- [38] G. BALARAC, F. BASILE, P. BÉNARD, F. BORDEU, J.-B. CHAPELIER, L. CIRROTTOLA, G. CAUMON, C. DAPOGNY, P. FREY, A. FROEHLI, ET AL., *Tetrahedral remeshing in the context of large-scale numerical simulation and high performance computing*, to appear in *Maths in Action*, (2021).
- [39] J. BALAS, J. SLADEK, AND V. SLADEK, *Stress analysis by boundary element methods*, Elsevier, 2013.
- [40] S. BALAY, K. BUSCHELMAN, W. D. GROPP, D. KAUSHIK, M. G. KNEPLEY, L. C. MCINNES, B. F. SMITH, AND H. ZHANG, *Petsc*, See <http://www.mcs.anl.gov/petsc>, (2001).
- [41] J. H. BALBI, F. MORANDINI, X. SILVANI, J. B. FILIPPI, AND F. RINIERI, *A physical model for wildland fires*, *Combustion and Flame*, 156 (2009), pp. 2217–2230.
- [42] J.-H. BALBI, J.-L. ROSSI, T. MARCELLI, AND P.-A. SANTONI, *A 3d physical real-time model of surface fires across fuel beds*, *Combustion Science and Technology*, 179 (2007), pp. 2511–2537.
- [43] E. BÄNGTSSON, D. NORELAND, AND M. BERGGREN, *Shape optimization of an acoustic horn*, *Computer methods in applied mechanics and engineering*, 192 (2003), pp. 1533–1571.
- [44] J. BAPTISTE FILIPPI, F. BOSSEUR, C. MARI, C. LAC, P. LE MOIGNE, B. CUENOT, D. VEYNANTE, D. CARIOLLE, AND J.-H. BALBI, *Coupled atmosphere-wildland fire modelling*, *Journal of Advances in Modeling Earth Systems*, 1 (2009).
- [45] J. R. BARBER, *Contact mechanics*, vol. 20, Springer, 2018.
- [46] N. BARRAL AND F. ALAUZET, *Three-dimensional cfd simulations with large displacement of the geometries using a connectivity-change moving mesh approach*, *Engineering with Computers*, 35 (2019), pp. 397–422.
- [47] T. J. BARTH AND J. A. SETHIAN, *Numerical schemes for the hamilton–jacobi and level set equations on triangulated domains*, *Journal of Computational Physics*, 145 (1998), pp. 1–40.
- [48] L. L. BEGHINI, A. BEGHINI, N. KATZ, W. F. BAKER, AND G. H. PAULINO, *Connecting architecture and engineering through structural topology optimization*, *Engineering Structures*, 59 (2014), pp. 716–726.
- [49] L. BEIRÃO DA VEIGA, F. BREZZI, A. CANGIANI, G. MANZINI, L. D. MARINI, AND A. RUSSO, *Basic principles of virtual element methods*, *Mathematical Models and Methods in Applied Sciences*, 23 (2013), pp. 199–214.
- [50] M. P. BENDSOE, *Optimization of structural topology, shape, and material*, vol. 414, Springer, 1995.
- [51] M. P. BENDSOE AND N. KIKUCHI, *Generating optimal topologies in structural design using a homogenization method*, *Computer methods in applied mechanics and engineering*, 71 (1988), pp. 197–224.
- [52] M. P. BENDSOE AND O. SIGMUND, *Topology optimization: theory, methods, and applications*, Springer Science & Business Media, 2013.
- [53] M. BERTALMIO, L.-T. CHENG, S. OSHER, AND G. SAPIRO, *Variational problems and partial differential equations on implicit surfaces*, *Journal of Computational Physics*, 174 (2001), pp. 759–780.
-

- [54] M. BERTALMIO, G. SAPIRO, AND G. RANDALL, *Region tracking on level-sets methods*, IEEE transactions on Medical Imaging, 18 (1999), pp. 448–451.
- [55] A. BONITO, A. DEMLOW, AND R. H. NOCHETTO, *Finite element methods for the laplace-beltrami operator*, in Handbook of Numerical Analysis, vol. 21, Elsevier, 2020, pp. 1–103.
- [56] E. BONNETIER, C. DAPOGNY, AND M. S. VOGELIUS, *Small perturbations in the type of boundary conditions for an elliptic operator*, To appear in Journal de Mathématiques Pures et Appliquées, (2021).
- [57] E. BONNETIER, C. DAPOGNY, AND M. S. VOGELIUS, *Small perturbations in the type of boundary conditions for an elliptic operator*, Journal de Mathématiques Pures et Appliquées, (2022).
- [58] T. BORRVALL AND J. PETERSSON, *Topology optimization of fluids in stokes flow*, International journal for numerical methods in fluids, 41 (2003), pp. 77–107.
- [59] N. BOURBAKI, *Topological vector spaces: Chapters 1–5*, Springer Science & Business Media, 2013.
- [60] B. BOURDIN, G. A. FRANCFORT, AND J.-J. MARIGO, *The variational approach to fracture*, Journal of elasticity, 91 (2008), pp. 5–148.
- [61] K. BRAKKE, *The motion of a surface by its mean curvature*, Princeton University.
- [62] H. BREZIS, *Functional analysis, Sobolev spaces and partial differential equations*, Springer Science & Business Media, 2010.
- [63] C. BRITO-PACHECO AND C. DAPOGNY, *Body-fitted tracking within a surface via a level set based mesh evolution method*, (2023).
- [64] L. BRONSARD AND R. V. KOHN, *Motion by mean curvature as the singular limit of ginzburg-landau dynamics*, Journal of differential equations, 90 (1991), pp. 211–237.
- [65] K. BRYAN AND T. LEISE, *Impedance imaging, inverse problems, and harry potter’s cloak*, SIAM review, 52 (2010), pp. 359–377.
- [66] C. BUI, C. DAPOGNY, AND P. FREY, *An accurate anisotropic adaptation method for solving the level set advection equation*, International Journal for Numerical Methods in Fluids, 70 (2012), pp. 899–922.
- [67] M. BURGER, *A framework for the construction of level set methods for shape optimization and reconstruction*, Interfaces and Free boundaries, 5 (2003), pp. 301–329.
- [68] M. BURGER, B. HACKL, AND W. RING, *Incorporating topological derivatives into level set methods*, Journal of Computational Physics, 194 (2004), pp. 344–362.
- [69] M. BURGER AND S. J. OSHER, *A survey on level set methods for inverse problems and optimal design*, European journal of applied mathematics, 16 (2005), pp. 263–301.
- [70] G. BUTTAZZO AND G. DAL MASO, *An existence result for a class of shape optimization problems*, Archive for rational mechanics and analysis, 122 (1993), pp. 183–195.
- [71] G. BUTTAZZO AND P. GUASONI, *Shape optimization problems over classes of convex domains*, Journal of Convex Analysis, 4 (1997), pp. 343–352.
- [72] J. CAGNOL, I. LASIECKA, C. LEBIEDZIK, AND J.-P. ZOLÉSIO, *Uniform stability in structural acoustic models with flexible curved walls*, Journal of Differential Equations, 186 (2002), pp. 88–121.
- [73] J. CAGNOL AND C. LEBIEDZIK, *On the free boundary conditions for a dynamic shell model based on intrinsic differential geometry*, Applicable Analysis, 83 (2004), pp. 607–633.
- [74] M. CALABRESE, T. PRIMO, AND A. DEL PRETE, *Optimization of machining fixture for aeronautical thin-walled components*, Procedia CIRP, 60 (2017), pp. 32–37.
- [75] M. CAMPEN AND L. KOBBELT, *Exact and robust (self-) intersections for polygonal meshes*, in Computer Graphics Forum, vol. 29, Wiley Online Library, 2010, pp. 397–406.

- 
- [76] M. CARIGLIA, T. HOURI, P. KRTOUŠ, AND D. KUBIZŇÁK, *On integrability of the geodesic deviation equation*, The European Physical Journal C, 78 (2018), pp. 1–17.
- [77] H. CARTAN, *Differential calculus*, Hermann, 1983.
- [78] J. CÉA, *Conception optimale ou identification de formes, calcul rapide de la dérivée directionnelle de la fonction coût*, ESAIM: Mathematical Modelling and Numerical Analysis, 20 (1986), pp. 371–402.
- [79] T. CHAN AND L. VESE, *Active contours without edges*, Image Processing, IEEE Transactions on, 10 (2001), pp. 266–277.
- [80] I. CHAVEL, *Riemannian geometry: a modern introduction*, vol. 98, Cambridge university press, 2006.
- [81] H. CHEN, Y. ZHANG, I. MEZIC, C. MEINHART, AND L. PETZOLD, *Numerical simulation of an electroosmotic micromixer*, in ASME International Mechanical Engineering Congress and Exposition, vol. 37165, 2003, pp. 653–658.
- [82] X. CHEN, *Generation and propagation of interfaces for reaction-diffusion equations*, Journal of Differential equations, 96 (1992), pp. 116–141.
- [83] N. CHENEY, J. GOULD, AND W. CATCHPOLE, *The influence of fuel, weather and fire shape variables on fire-spread in grasslands*, International Journal of Wildland Fire, 3 (1993), pp. 31–44.
- [84] L.-T. CHENG, P. BURCHARD, B. MERRIMAN, AND S. OSHER, *Motion of curves constrained on surfaces using a level-set approach*, Journal of Computational Physics, 175 (2002), pp. 604–644.
- [85] S.-W. CHENG, T. K. DEY, J. SHEWCHUK, AND S. SAHNI, *Delaunay mesh generation*, CRC Press Boca Raton, FL, 2013.
- [86] H. CHI, Y. ZHANG, T. L. E. TANG, L. MIRABELLA, L. DALLORO, L. SONG, AND G. H. PAULINO, *Universal machine learning for topology optimization*, Computer Methods in Applied Mechanics and Engineering, 375 (2021), p. 112739.
- [87] D. L. CHOPP AND J. A. SETHIAN, *Flow under curvature: singularity formation, minimal surfaces, and geodesics*, Experimental Mathematics, 2 (1993), pp. 235–255.
- [88] A. N. CHRISTIANSEN, J. A. BÆRENTZEN, M. NOBEL-JORGENSEN, N. AAGE, AND O. SIGMUND, *Combined shape and topology optimization of 3d structures*, Computers & Graphics, 46 (2015), pp. 25–35.
- [89] A. N. CHRISTIANSEN, M. NOBEL-JORGENSEN, N. AAGE, O. SIGMUND, AND J. A. BÆRENTZEN, *Topology optimization using an explicit interface representation*, Structural and Multidisciplinary Optimization, 49 (2014), pp. 387–399.
- [90] T. CHUNG, *Computational Fluid Dynamics*, Cambridge University Press, 2002.
- [91] P. G. CIARLET, *Theory of shells*, Elsevier, 2000.
- [92] ———, *The finite element method for elliptic problems*, vol. 40, Siam, 2002.
- [93] P. G. CIARLET, *Mathematical Elasticity: Three-Dimensional Elasticity*, Society for Industrial and Applied Mathematics, Philadelphia, PA, 2021.
- [94] S. COMINELLI, D. E. QUADRELLI, C. SINIGAGLIA, AND F. BRAGHIN, *Design of arbitrarily shaped acoustic cloaks through partial differential equation-constrained optimization satisfying sonic-metamaterial design requirements*, Proceedings of the Royal Society A, 478 (2022), p. 20210750.
- [95] E. COPSON, *On the problem of the electrified disc*, Proceedings of the Edinburgh Mathematical Society, 8 (1947), pp. 14–19.
- [96] G.-H. COTTET AND E. MAITRE, *A level set method for fluid-structure interactions with immersed surfaces*, Mathematical models and methods in applied sciences, 16 (2006), pp. 415–438.
- [97] M. G. CRANDALL, H. ISHII, AND P.-L. LIONS, *User’s guide to viscosity solutions of second order partial differential equations*, Bulletin of the American mathematical society, 27 (1992), pp. 1–67.
-

- [98] S. CUI, C. ZHANG, M. FILLON, AND L. GU, *Optimization performance of plain journal bearings with partial wall slip*, Tribology International, 145 (2020), p. 106137.
- [99] B. DACOROGNA, *Introduction to the Calculus of Variations*, World Scientific Publishing Company, 2014.
- [100] C. DAPOGNY, *The topological ligament in shape optimization: an approach based on thin tubular inhomogeneities asymptotics*, SMAI Journal of Computational Mathematics, (2021), pp. 185–266.
- [101] ———, *A few contributions in shape and topology optimization*, PhD thesis, Université Grenoble Alpes, 2024.
- [102] C. DAPOGNY, C. DOBRZYNSKI, AND P. FREY, *Three-dimensional adaptive domain remeshing, implicit domain meshing, and applications to free and moving boundary problems*, Journal of computational physics, 262 (2014), pp. 358–378.
- [103] C. DAPOGNY, C. DOBRZYNSKI, P. FREY, AND A. FROELHY, *mmg*, available at: <https://www.mmgtools.org>, 2019.
- [104] C. DAPOGNY AND F. FEPPON, *Shape optimization using a level set based mesh evolution method: an overview and tutorial*, Comptes Rendus. Mathématique, 361 (2023), pp. 1267–1332.
- [105] C. DAPOGNY AND P. FREY, *Computation of the signed distance function to a discrete contour on adapted triangulation*, Calcolo, 49 (2012), pp. 193–219.
- [106] C. DAPOGNY, P. FREY, AND A. FROELHY, *ISCD Toolbox*, <https://github.com/ISCDtoolbox>, 2019.
- [107] C. DAPOGNY, P. FREY, F. OMNES, AND Y. PRIVAT, *Geometrical shape optimization in fluid mechanics using freefem++*, Structural and Multidisciplinary Optimization, 58 (2018), pp. 2761–2788.
- [108] C. DAPOGNY, N. LEBBE, AND E. OUDET, *Optimization of the shape of regions supporting boundary conditions*, Numerische Mathematik, 146 (2020), pp. 51–104.
- [109] C. DAPOGNY, B. LEVY, AND E. OUDET, *A lagrangian shape and topology optimization framework based on semi-discrete optimal transport*, arXiv preprint arXiv:2409.07873, (2024).
- [110] M. DAUGE, *Elliptic boundary value problems on corner domains: smoothness and asymptotics of solutions*, vol. 1341, Springer, 2006.
- [111] P. J. DAVIS AND P. RABINOWITZ, *Methods of numerical integration*, Courier Corporation, 2007.
- [112] T. A. DAVIS, *Algorithm 832: Umfpack v4. 3—an unsymmetric-pattern multifrontal method*, ACM Transactions on Mathematical Software (TOMS), 30 (2004), pp. 196–199.
- [113] I. DAYYANI, A. SHAW, E. S. FLORES, AND M. FRISWELL, *The mechanics of composite corrugated structures: A review with applications in morphing aircraft*, Composite Structures, 133 (2015), pp. 358–380.
- [114] F. DE GOURNAY, *Velocity extension for the level-set method and multiple eigenvalues in shape optimization*, SIAM journal on control and optimization, 45 (2006), pp. 343–367.
- [115] P. DE MOTTONI AND M. SCHATZMAN, *Geometrical evolution of developed interfaces*, Transactions of the American Mathematical Society, 347 (1995), pp. 1533–1589.
- [116] M. C. DELFOUR AND J.-P. ZOLÉSIO, *Shapes and geometries: metrics, analysis, differential calculus, and optimization*, SIAM, 2011.
- [117] Y. DENG, W. ZHANG, Z. LIU, J. ZHU, AND J. G. KORVINK, *Topology optimization for surface flows*, Journal of Computational Physics, 467 (2022), p. 111415.
- [118] Y. DENG, T. ZHOU, Z. LIU, Y. WU, S. QIAN, AND J. G. KORVINK, *Topology optimization of electrode patterns for electroosmotic micromixer*, International Journal of Heat and Mass Transfer, 126 (2018), pp. 1299–1315.

- 
- 
- [119] J. DESAI, A. FAURE, G. MICHALIDIS, G. PARRY, AND R. ESTEVEZ, *Topology optimization in acoustics and elasto-acoustics via a level-set method*, Journal of Sound and Vibration, 420 (2018), pp. 73–103.
- [120] D. A. DI PIETRO AND A. ERN, *Mathematical aspects of discontinuous Galerkin methods*, vol. 69, Springer Science & Business Media, 2011.
- [121] M. P. DO CARMO, *Differential geometry of curves and surfaces: revised and updated second edition*, Courier Dover Publications, 2016.
- [122] M. P. DO CARMO AND J. FLAHERTY FRANCIS, *Riemannian geometry*, vol. 115, Birkhäuser Boston, 1992.
- [123] C. DOBRZYNSKI AND P. FREY, *Anisotropic delaunay mesh adaptation for unsteady simulations*, in Proceedings of the 17th international Meshing Roundtable, Springer, 2008, pp. 177–194.
- [124] G. DOĞAN, *Shape calculus for shape energies in image processing*, arXiv preprint arXiv:1307.5797, (2013).
- [125] G. DOĞAN, P. MORIN, AND R. H. NOCHETTO, *A variational shape optimization approach for image segmentation with a mumford–shah functional*, SIAM Journal on Scientific Computing, 30 (2008), pp. 3028–3049.
- [126] G. DOGAN, P. MORIN, R. H. NOCHETTO, AND M. VERANI, *Discrete gradient flows for shape optimization and applications*, Computer methods in applied mechanics and engineering, 196 (2007), pp. 3898–3914.
- [127] A. DOI AND A. KOIDE, *An efficient method of triangulating equi-valued surfaces by using tetrahedral cells*, IEICE TRANSACTIONS on Information and Systems, 74 (1991), pp. 214–224.
- [128] Q. DU AND X. FENG, *The phase field method for geometric moving interfaces and their numerical approximations*, Handbook of numerical analysis, 21 (2020), pp. 425–508.
- [129] M. B. DÜHRING, J. S. JENSEN, AND O. SIGMUND, *Acoustic design by topology optimization*, Journal of sound and vibration, 317 (2008), pp. 557–575.
- [130] P. D. DUNNING AND H. A. KIM, *Introducing the sequential linear programming level-set method for topology optimization*, Structural and Multidisciplinary Optimization, 51 (2015), pp. 631–643.
- [131] L. J. DURLOFSKY, B. ENGQUIST, AND S. OSHER, *Triangle based adaptive stencils for the solution of hyperbolic conservation laws*, Journal of Computational Physics, 98 (1992), pp. 64–73.
- [132] G. DZIUK AND C. M. ELLIOTT, *Finite elements on evolving surfaces*, IMA journal of numerical analysis, 27 (2007), pp. 262–292.
- [133] ———, *Finite element methods for surface pdes*, Acta Numerica, 22 (2013), pp. 289–396.
- [134] W. E AND B. YU, *The deep ritz method: a deep learning-based numerical algorithm for solving variational problems*, Communications in Mathematics and Statistics, 6 (2018), pp. 1–12.
- [135] K. ECKER, *Regularity theory for mean curvature flow*, vol. 57, Springer Science & Business Media, 2012.
- [136] A. ERN AND J.-L. GUERMOND, *Finite elements I: Approximation and interpolation*, vol. 72, Springer Nature, 2021.
- [137] ———, *Finite Elements II: Galerkin Approximation, Elliptic and Mixed PDEs*, Springer, Apr. 2021.
- [138] L. EULER, *Methodus inveniendi lineas curvas maximi minimive proprietate gaudentes sive solutio problematis isoperimetrci latissimo sensu accepti*, vol. 1, Springer Science & Business Media, 1952.
- [139] L. C. EVANS, *Partial differential equations*, Graduate studies in mathematics, 19 (1998).
- [140] L. C. EVANS AND R. F. GARIEPY, *Measure theory and fine properties of functions*, vol. 5, CRC press Boca Raton, 1992.
- 
-

- [141] L. C. EVANS AND R. F. GARIEPY, *Measure theory and fine properties of functions*, CRC press, 2015.
- [142] F. FEPPON, G. ALLAIRE, F. BORDEU, J. CORTIAL, AND C. DAPOGNY, *Shape optimization of a coupled thermal fluid–structure problem in a level set mesh evolution framework*, *SeMA Journal*, 76 (2019), pp. 413–458.
- [143] F. FEPPON, G. ALLAIRE, AND C. DAPOGNY, *Null space gradient flows for constrained optimization with applications to shape optimization*, *ESAIM: Control, Optimisation and Calculus of Variations*, 26 (2020), p. 90.
- [144] F. FEPPON, G. ALLAIRE, C. DAPOGNY, AND P. JOLIVET, *Topology optimization of thermal fluid–structure systems using body-fitted meshes and parallel computing*, *Journal of Computational Physics*, 417 (2020), p. 109574.
- [145] J.-B. FILIPPI, F. BOSSEUR, AND D. GRANDI, *Forefire: open-source code for wildland fire spread models*, *Advances in forest fire research*.(Ed. DX Viegas)(Imprensa da Universidade de Coimbra: Coimbra, Portugal), (2014).
- [146] J.-B. FILIPPI, X. PIALAT, AND C. B. CLEMENTS, *Assessment of forefire/meso-nh for wildland fire/atmosphere coupled simulation of the fireflux experiment*, *Proceedings of the Combustion Institute*, 34 (2013), pp. 2633–2640.
- [147] M. A. FINNEY, *Calculation of fire spread rates across random landscapes*, *International Journal of Wildland Fire*, 12 (2003), pp. 167–174.
- [148] M. FIRL, *Optimal shape design of shell structures*, 2010.
- [149] G. B. FOLLAND, *Introduction to partial differential equations*, Princeton university press, 1995.
- [150] ———, *Introduction to partial differential equations*, Princeton university press, 2020.
- [151] G. FREMIOT AND J. SOKOLOWSKI, *Shape sensitivity analysis of problems with singularities*, *Shape optimization and optimal design* (Cambridge, 1999), *Lecture Notes in Pure and Appl. Math.*, 216 (1999), pp. 255–276.
- [152] P. J. FREY AND P.-L. GEORGE, *Mesh generation: application to finite elements*, ISTE, 2007.
- [153] T.-P. FRIES, S. OMERVIĆ, D. SCHÖLLHAMMER, AND J. STEIDL, *Higher-order meshing of implicit geometries—part i: Integration and interpolation in cut elements*, *Computer Methods in Applied Mechanics and Engineering*, 313 (2017), pp. 759–784.
- [154] T.-P. FRIES AND D. SCHÖLLHAMMER, *Higher-order meshing of implicit geometries, part ii: Approximations on manifolds*, *Computer Methods in Applied Mechanics and Engineering*, 326 (2017), pp. 270–297.
- [155] S. F. FRISKEN, R. N. PERRY, A. P. ROCKWOOD, AND T. R. JONES, *Adaptively sampled distance fields: A general representation of shape for computer graphics*, in *Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, 2000, pp. 249–254.
- [156] A. FUHRMANN, G. SOBOTKA, AND C. GROSS, *Distance fields for rapid collision detection in physically based modeling*, in *Proceedings of GraphiCon*, vol. 2003, Citeseer, 2003, pp. 58–65.
- [157] G. FUJII, M. TAKAHASHI, AND Y. AKIMOTO, *Acoustic cloak designed by topology optimization for acoustic–elastic coupled systems*, *Applied Physics Letters*, 118 (2021).
- [158] P. GANGL, *Sensitivity-based topology and shape optimization with application to electrical machines*, PhD thesis, Ph. D. thesis, Johannes Kepler University Linz, 2016.
- [159] P. GANGL AND N. KRENN, *Topology optimization of a rotating electric machine by the topological derivative*, *PAMM*, 23 (2023), p. e202200052.
- [160] T. GAO, Y. WANG, Y. PANG, AND J. CAO, *Hull shape optimization for autonomous underwater vehicles using cfd*, *Engineering applications of computational fluid mechanics*, 10 (2016), pp. 599–607.

- 
- 
- [161] I. M. GELFAND, R. A. SILVERMAN, ET AL., *Calculus of variations*, Courier Corporation, 2000.
- [162] C. GEUZAIN AND J.-F. REMACLE, *Gmsh: A 3-d finite element mesh generator with built-in pre-and post-processing facilities*, International journal for numerical methods in engineering, 79 (2009), pp. 1309–1331.
- [163] Y. GIGA, *Surface evolution equations*, Springer, 2006.
- [164] D. GILBARG AND N. S. TRUDINGER, *Elliptic partial differential equations of second order*, springer, 2015.
- [165] D. GILBARG, N. S. TRUDINGER, D. GILBARG, AND N. TRUDINGER, *Elliptic partial differential equations of second order*, vol. 224, Springer, 1977.
- [166] G. H. GOLUB AND J. H. WELSCH, *Calculation of gauss quadrature rules*, Mathematics of computation, 23 (1969), pp. 221–230.
- [167] A. GREENLEAF, Y. KURYLEV, M. LASSAS, AND G. UHLMANN, *Cloaking devices, electromagnetic wormholes, and transformation optics*, SIAM review, 51 (2009), pp. 3–33.
- [168] ———, *Invisibility and inverse problems*, Bulletin of the American Mathematical Society, 46 (2009), pp. 55–97.
- [169] J. B. GREER, *An improvement of a recent eulerian method for solving pdes on general geometries*, Journal of Scientific Computing, 29 (2006), pp. 321–352.
- [170] R. GRIESMAIER AND M. S. VOGELIUS, *Enhanced approximate cloaking by optimal change of variables*, Inverse Problems, 30 (2014), p. 035014.
- [171] R. GRIGORIEFF AND I. SLOAN, *Galerkin approximation with quadrature for the screen problem in  $r \geq 3$* , The Journal of Integral Equations and Applications, (1997), pp. 293–319.
- [172] P. GRISVARD, *Elliptic problems in nonsmooth domains*, SIAM, 2011.
- [173] J. E. GUYER, D. WHEELER, AND J. A. WARREN, *Fipy: Partial differential equations with python*, Computing in Science & Engineering, 11 (2009), pp. 6–15.
- [174] J. GWINNER AND E. P. STEPHAN, *Advanced boundary element methods*, Treatment of Boundary Value, Transmission and Contact Problems. Cham: Springer, (2018).
- [175] W. HACKBUSCH AND W. HACKBUSCH, *The integral equation method*, Springer, 1995.
- [176] W. HACKBUSCH AND S. A. SAUTER, *On the efficient use of the galerkin-method to solve fredholm integral equations*, Applications of mathematics, 38 (1993), pp. 301–322.
- [177] J. HADAMARD, *Mémoire sur le problème d'analyse relatif à l'équilibre des plaques élastiques encastrées*, vol. 33, Imprimerie nationale, 1908.
- [178] R. S. HAMILTON, *The inverse function theorem of nash and moser*, Bulletin (New Series) of the American Mathematical Society, 7 (1982), pp. 65–222.
- [179] S. HANG, *Tetgen, a delaunay-based quality tetrahedral mesh generator*, ACM Trans. Math. Softw, 41 (2015), p. 11.
- [180] T. L. HEATH, *A history of Greek mathematics*, vol. 1, Courier Corporation, 1981.
- [181] P. HÉBRARD AND A. HENROTT, *Optimal shape and position of the actuators for the stabilization of a string*, Systems & control letters, 48 (2003), pp. 199–209.
- [182] F. HECHT, *New development in freefem++*, Journal of numerical mathematics, 20 (2012), pp. 251–266.
- [183] A. HENROT AND M. PIERRE, *Variation et optimisation de formes: une analyse géométrique*, vol. 48, Springer Science & Business Media, 2005.
- [184] ———, *Variation et optimisation de formes: une analyse géométrique*, vol. 48, Springer Science & Business Media, 2006.
- 
-

- [185] ———, *Shape Variation and Optimization*, EMS Tracts in Mathematics Vol. 28, 2018.
- [186] A. HENROT AND J. SOKOŁOWSKI, *Mathematical challenges in shape optimization*, Control and Cybernetics, 34 (2005), pp. 37–57.
- [187] P. HILD, I. R. IONESCU, T. LACHAND-ROBERT, AND I. ROȘCA, *The blocking of an inhomogeneous bingham fluid. applications to landslides*, ESAIM: Mathematical Modelling and Numerical Analysis, 36 (2002), pp. 1013–1026.
- [188] R. HIPTMAIR AND J. LI, *Shape derivatives in differential forms i: an intrinsic perspective*, Annali di matematica pura ed applicata, 192 (2013), pp. 1077–1098.
- [189] F.-H. HO AND P. MAILLARD, *Efficient approximation of branching random walk gibbs measures*, Electronic Journal of Probability, 27 (2022), pp. 1–18.
- [190] K. H. HUEBNER, D. L. DEWHIRST, D. E. SMITH, AND T. G. BYROM, *The finite element method for engineers*, John Wiley & Sons, 2001.
- [191] R. HUISKES AND R. BOEKLAGE, *Mathematical shape optimization of hip prosthesis design*, Journal of Biomechanics, 22 (1989), pp. 793–804.
- [192] T. ILMANEN, *Generalized flow of sets by mean curvature on a manifold*, Indiana University mathematics journal, (1992), pp. 671–705.
- [193] E. ISTANBULLUOĞLU AND R. L. BRAS, *Vegetation-modulated landscape evolution: Effects of vegetation on landscape processes, drainage density, and topography*, Journal of Geophysical Research: Earth Surface, 110 (2005).
- [194] J. D. JACKSON, *Classical electrodynamics*, John Wiley & Sons, 2021.
- [195] H. JALILI, M. RAAD, AND D. A. FALLAH, *Numerical study on the mixing quality of an electroosmotic micromixer under periodic potential*, Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science, 234 (2020), pp. 2113–2125.
- [196] J. S. JENSEN AND O. SIGMUND, *Topology optimization for nano-photonics*, Laser & Photonics Reviews, 5 (2011), pp. 308–321.
- [197] A. J. JERRI, *Introduction to integral equations with applications*, John Wiley & Sons, 1999.
- [198] G.-S. JIANG AND D. PENG, *Weighted eno schemes for hamilton–jacobi equations*, SIAM Journal on Scientific computing, 21 (2000), pp. 2126–2143.
- [199] K. L. JOHNSON, *Contact mechanics*, Cambridge university press, 1987.
- [200] G. KARYPIS AND V. KUMAR, *Metis: A software package for partitioning unstructured graphs, partitioning meshes, and computing fill-reducing orderings of sparse matrices*, (1997).
- [201] N. KAYA, *Machining fixture locating and clamping position optimization using genetic algorithms*, Computers in industry, 57 (2006), pp. 112–120.
- [202] C. KIM, M. JUNG, T. YAMADA, S. NISHIWAKI, AND J. YOO, *Freefem++ code for reaction-diffusion equation-based topology optimization: for high-resolution boundary representation using adaptive mesh refinement*, Structural and Multidisciplinary Optimization, 62 (2020), pp. 439–455.
- [203] K.-Y. KIM AND S.-S. KIM, *Shape optimization of rib-roughened surface to enhance turbulent heat transfer*, International Journal of Heat and Mass Transfer, 45 (2002), pp. 2719–2727.
- [204] R. KIMMEL, *Intrinsic scale space for images on surfaces: The geodesic curvature flow*, Graphical models and image processing, 59 (1997), pp. 365–372.
- [205] R. KIMMEL AND J. A. SETHIAN, *Computing geodesic paths on manifolds*, Proceedings of the national academy of Sciences, 95 (1998), pp. 8431–8435.
- [206] M. KLINE, *Mathematical Thought from Ancient to Modern Times: Volume 2*, vol. 2, Oxford university press, 1990.

- 
- 
- [207] P. KNUPP AND K. SALARI, *Verification of computer codes in computational science and engineering*, Chapman and Hall/CRC, 2002.
- [208] H. C. KO, G. SHIN, S. WANG, M. P. STOYKOVICH, J. W. LEE, D.-H. KIM, J. S. HA, Y. HUANG, K.-C. HWANG, AND J. A. ROGERS, *Curvilinear electronics formed using silicon membrane circuits and elastomeric transfer elements*, *small*, 5 (2009), pp. 2703–2709.
- [209] L. KOBBELT AND M. BOTSCH, *A survey of point-based techniques in computer graphics*, *Computers & Graphics*, 28 (2004), pp. 801–814.
- [210] R. V. KOHN, H. SHEN, M. S. VOGELIUS, AND M. I. WEINSTEIN, *Cloaking via change of variables in electric impedance tomography*, *Inverse Problems*, 24 (2008), p. 015016.
- [211] T. KOLEV AND V. DOBREV, *Glvis: OpenGl finite element visualization tool*, tech. rep., Lawrence Livermore National Laboratory (LLNL), Livermore, CA (United States), 2010.
- [212] G. KÖTHER, *Topological vector spaces*, in *Topological Vector Spaces I*, Springer, 1983, pp. 123–201.
- [213] V. A. KOZLOV, V. MAZIA, AND J. ROSSMANN, *Elliptic boundary value problems in domains with point singularities*, vol. 52, American Mathematical Soc., 1997.
- [214] S. KRENK, *A circular crack under asymmetric loads and some related integral equations*, *Journal of Applied Mechanics*, 46 (1979).
- [215] L. KROG, A. TUCKER, M. KEMP, AND R. BOYD, *Topology optimisation of aircraft wing box ribs*, in 10th AIAA/ISSMO multidisciplinary analysis and optimization conference, 2004, p. 4481.
- [216] J. L. LAGRANGE, *Essai d’une nouvelle methode pour de’terminer les maxima, et les minima des formules integrales indefinies*, De l’Imprimerie royale, 1761.
- [217] B.-J. LAI, L.-T. ZHU, Z. CHEN, B. OUYANG, AND Z.-H. LUO, *Review on blood flow dynamics in lab-on-a-chip systems: An engineering perspective*, *Chem & Bio Engineering*, (2024).
- [218] N. LEBBE, C. DAPOGNY, E. OUDET, K. HASSAN, AND A. GLIERE, *Robust shape and topology optimization of nanophotonic devices using the level set method*, *Journal of Computational Physics*, 395 (2019), pp. 710–746.
- [219] J. M. LEE, *Riemannian manifolds: an introduction to curvature*, vol. 176, Springer Science & Business Media, 2006.
- [220] ———, *Introduction to Riemannian manifolds*, vol. 2, Springer, 2018.
- [221] J. LEOPOLD, *Clamping modeling-state-of-the-art and future trends*, in *Intelligent Robotics and Applications: First International Conference, ICIRA 2008 Wuhan, China, October 15-17, 2008 Proceedings, Part II 1*, Springer, 2008, pp. 289–300.
- [222] J. LEOPOLD AND L. HONG, *Clamping modelling: state-of-the-art and future trends*, *Industrial Robot: An International Journal*, 36 (2009), pp. 249–254.
- [223] R. J. LEVEQUE, *Finite difference methods for ordinary and partial differential equations: steady-state and time-dependent problems*, vol. 98, Siam, 2007.
- [224] B. LI, Z. PENG, H. WEN, J. FAN, AND H. SONG, *Research on the optimization design of acoustic stealth shape of the underwater vehicle head*, *Acoustics Australia*, 48 (2020), pp. 39–47.
- [225] J. LI, X. DU, AND J. R. MARTINS, *Machine learning in aerodynamic shape optimization*, *Progress in Aerospace Sciences*, 134 (2022), p. 100849.
- [226] W. E. LORENSEN AND H. E. CLINE, *Marching cubes: A high resolution 3d surface construction algorithm*, *ACM siggraph computer graphics*, 21 (1987), pp. 163–169.
- [227] J. M. M. LUZ FILHO, R. MATTOSO, AND L. FERNANDEZ, *A freefem code for topological derivative-based structural optimization*, *Structural and Multidisciplinary Optimization*, 66 (2023), p. 74.
- 
-

- [228] J. MA, M. Y. WANG, AND X. ZHU, *Compliant fixture layout design using topology optimization method*, in 2011 IEEE International Conference on Robotics and Automation, IEEE, 2011, pp. 3757–3763.
- [229] Z. MA, O. STALNOV, AND X. HUANG, *Design method for an acoustic cloak in flows by topology optimization*, *Acta Mechanica Sinica*, 35 (2019), pp. 964–971.
- [230] C. B. MACDONALD AND S. J. RUUTH, *Level set equations on surfaces via the closest point method*, *Journal of Scientific Computing*, 35 (2008), pp. 219–240.
- [231] ———, *The implicit closest point method for the numerical solution of partial differential equations on surfaces*, *SIAM Journal on Scientific Computing*, 31 (2010), pp. 4330–4350.
- [232] C. MADER AND J. R. MARTINS, *Optimal flying wings: A numerical optimization study*, in 53rd AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Materials Conference 20th AIAA/ASME/AHS Adaptive Structures Conference 14th AIAA, 2012, p. 1758.
- [233] J. MARGERIT AND O. SÉRO-GUILLAUME, *Modelling forest fires. part ii: reduction to two-dimensional models and simulation of propagation*, *International Journal of Heat and Mass Transfer*, 45 (2002), pp. 1723–1737.
- [234] E. MARTINET, *Numerical optimization of neumann eigenvalues of domains in the sphere*, *Journal of Computational Physics*, 508 (2024), p. 113002.
- [235] J. MARTÍNEZ-FRUTOS, G. ALLAIRE, C. DAPOGNY, AND F. PERIAGO, *Structural optimization under internal porosity constraints using topological derivatives*, *Computer Methods in Applied Mechanics and Engineering*, 345 (2019), pp. 1–25.
- [236] S. MAZUMDER, *Numerical methods for partial differential equations: finite difference and finite volume methods*, Academic Press, 2015.
- [237] W. C. H. MCLEAN, *Strongly elliptic systems and boundary integral equations*, Cambridge university press, 2000.
- [238] R. D. MINDLIN, *Force at a point in the interior of a semi-infinite solid*, *physics*, 7 (1936), pp. 195–202.
- [239] M. K. MISZTAL AND J. A. BÆRENTZEN, *Topology-adaptive interface tracking using the deformable simplicial complex*, *ACM Transactions on Graphics (TOG)*, 31 (2012), pp. 1–12.
- [240] N. MOES, J.-F. REMACLE, J. LAMBRECHTS, B. LE, AND N. CHEVAUGEON, *The extreme mesh deformation approach (x-mesh) for the stefan phase change model*, *Journal of Computational Physics*, (2023), p. 111878.
- [241] B. MOHAMMADI AND O. PIRONNEAU, *Shape optimization in fluid mechanics*, *Annu. Rev. Fluid Mech.*, 36 (2004), pp. 255–279.
- [242] ———, *Applied shape optimization for fluids*, Oxford university press, 2010.
- [243] D. MORVAN, G. ACCARY, S. MERADJI, AND N. FRANGIEH, *Fifty years of progress in wildland fire modelling: from empirical to fully physical cfd models*, *Comptes Rendus. Mécanique*, 350 (2022), pp. 1–9.
- [244] K. MUKHERJEA, *Differential Calculas in Normed Linear Spaces*, vol. 26, Springer, 2007.
- [245] H. MUNTHE-KAAS, *High order runge-kutta methods on manifolds*, *Applied Numerical Mathematics*, 29 (1999), pp. 115–127.
- [246] T. MURA, *Micromechanics of defects in solids*, Springer Science & Business Media, 2013.
- [247] F. MURAT AND J. SIMON, *Sur le contrôle par un domaine géométrique*, *Pré-publication du Laboratoire d’Analyse Numérique*,(76015), (1976).
- [248] T. G. MYERS, J. P. CHARPIN, AND S. J. CHAPMAN, *The flow and solidification of a thin fluid film on an arbitrary three-dimensional surface*, *Physics of Fluids*, 14 (2002), pp. 2788–2803.
- 
-

- 
- 
- [249] N.-T. NGUYEN, S. T. WERELEY, AND S. A. M. SHAEGH, *Fundamentals and applications of microfluidics*, Artech house, 2019.
- [250] L. I. NICOLAESCU, *The coarea formula*, in seminar notes. Citeseer, 2011.
- [251] F. NISHANTH AND B. WANG, *Topology optimization of electric machines: A review*, in 2022 IEEE Energy Conversion Congress and Exposition (ECCE), IEEE, 2022, pp. 1–8.
- [252] A. N. NORRIS, *Acoustic cloaking theory*, Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences, 464 (2008), pp. 2411–2434.
- [253] A. NOVOTNY, C. LOPES, AND R. SANTOS, *Topological derivative-based topology optimization of structures subject to self-weight loading*, Structural and Multidisciplinary Optimization, 63 (2021), pp. 1853–1861.
- [254] A. A. NOVOTNY AND J. SOKOŁOWSKI, *Topological derivatives in shape optimization*, Springer Science & Business Media, 2012.
- [255] C. F. OLLIVIER-GOOCH, *Quasi-eno schemes for unstructured meshes based on unlimited data-dependent least-squares reconstruction*, Journal of Computational Physics, 133 (1997), pp. 6–17.
- [256] S. OSHER AND R. FEDKIW, *Signed distance functions*, in Level set methods and dynamic implicit surfaces, Springer, 2003, pp. 17–22.
- [257] ———, *Level set methods and dynamic implicit surfaces*, vol. 153, Springer Science & Business Media, 2006.
- [258] S. OSHER AND J. A. SETHIAN, *Fronts propagating with curvature-dependent speed: algorithms based on hamilton-jacobi formulations*, Journal of computational physics, 79 (1988), pp. 12–49.
- [259] R. OSSERMAN, *The isoperimetric inequality*, Bulletin of the American Mathematical Society, 84 (1978), pp. 1182–1238.
- [260] Q. PAN, X. ZHAI, AND F. CHEN, *Density-based isogeometric topology optimization of shell structures*, Computer-Aided Design, 176 (2024), p. 103773.
- [261] J. J. PARK, P. FLORENCE, J. STRAUB, R. NEWCOMBE, AND S. LOVEGROVE, *Deepsdf: Learning continuous signed distance functions for shape representation*, in Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2019, pp. 165–174.
- [262] L. E. PAYNE, *Isoperimetric inequalities and their applications*, SIAM review, 9 (1967), pp. 453–488.
- [263] F. PELLEGRINI, *Scotch and libscotch 5.1 user’s guide*, (2008).
- [264] O. PIRONNEAU, *Finite element methods for fluids*, Wiley Chichester, 1989.
- [265] B.-I. POPA AND S. A. CUMMER, *Cloaking with optimized homogeneous anisotropic layers*, Physical Review A, 79 (2009), p. 023806.
- [266] M. POULIN, R. ANDERSEN, AND L. ROCHEFORT, *A new approach for tracking vegetation change after restoration: a case study with peatlands*, Restoration Ecology, 21 (2013), pp. 363–371.
- [267] B. PROTAS, T. R. BEWLEY, AND G. HAGEN, *A computational framework for the regularization of adjoint analysis in multiscale pde systems*, Journal of Computational Physics, 195 (2004), pp. 49–89.
- [268] J. QIAN, Y.-T. ZHANG, AND H.-K. ZHAO, *Fast sweeping methods for eikonal equations on triangular meshes*, SIAM Journal on Numerical Analysis, 45 (2007), pp. 83–107.
- [269] S. QIAN AND H. H. BAU, *A chaotic electroosmotic stirrer*, Analytical Chemistry, 74 (2002), pp. 3616–3625.
- [270] A. QUARTERONI AND G. ROZZA, *Optimal control and shape optimization of aorto-coronary bypass anastomoses*, Mathematical Models and Methods in Applied Sciences, 13 (2003), pp. 1801–1823.
- [271] S. RALLABHANDI, *Sonic boom minimization through vehicle shape optimization and probabilistic acoustic propagation*, Georgia Institute of Technology, 2005.
- 
-

- [272] P. W. ROBINSON, S. SILTANEN, T. LOKKI, AND L. SAVIOJA, *Concert hall geometry optimization with parametric modeling tools and wave-based acoustic simulations*, *Building Acoustics*, 21 (2014), pp. 55–63.
- [273] R. C. ROTHERMEL, *A mathematical model for predicting fire spread in wildland fuels*, vol. 115, Intermountain Forest & Range Experiment Station, Forest Service, US . . . , 1972.
- [274] G. I. ROZVANY, *A critical review of established methods of structural topology optimization*, *Structural and multidisciplinary optimization*, 37 (2009), pp. 217–237.
- [275] G. I. ROZVANY, M. ZHOU, AND T. BIRKER, *Generalized shape optimization without homogenization*, *Structural optimization*, 4 (1992), pp. 250–252.
- [276] W. RUDIN, *Functional Analysis*, International series in pure and applied mathematics, Tata McGraw-Hill, 1974.
- [277] ———, *Functional Analysis*, International series in pure and applied mathematics, McGraw-Hill, 1991.
- [278] S. J. RUUTH AND B. MERRIMAN, *A simple embedding method for solving partial differential equations on surfaces*, *Journal of Computational Physics*, 227 (2008), pp. 1943–1961.
- [279] Y. SAAD AND M. H. SCHULTZ, *Gmres: A generalized minimal residual algorithm for solving nonsymmetric linear systems*, *SIAM Journal on scientific and statistical computing*, 7 (1986), pp. 856–869.
- [280] M. SAHRAOUI AND M. KAVIANY, *Slip and no-slip velocity boundary conditions at interface of porous, plain media*, *International Journal of Heat and Mass Transfer*, 35 (1992), pp. 927–943.
- [281] G. SAPIRO, *Geometric partial differential equations and image analysis*, Cambridge university press, 2006.
- [282] N. SASAKI, T. KITAMORI, AND H.-B. KIM, *Ac electroosmotic micromixer for chemical processing in a microchannel*, *Lab on a Chip*, 6 (2006), pp. 550–554.
- [283] S. A. SAUTER AND C. SCHWAB, *Boundary element methods*, *Boundary Element Methods*, (2011), pp. 183–287.
- [284] H. SCHAEFFER, *Topological vector spaces*, *mcmillan, new york, 1966*, MR, 33, p. 1689.
- [285] A. SCHIELA AND J. ORTIZ, *Second order directional shape derivatives of integrals on submanifolds*, *Mathematical Control & Related Fields*, 11 (2021), p. 653.
- [286] U. SCHRAMM, H. THOMAS, AND M. ZHOU, *Manufacturing considerations and structural optimization for automotive components*, tech. rep., SAE Technical Paper, 2002.
- [287] V. H. SCHULZ, *A Riemannian view on shape optimization*, *Found. Comput. Math.*, 14 (2014), pp. 483–501.
- [288] M. W. SCROGGS, I. A. BARATTA, C. N. RICHARDSON, AND G. N. WELLS, *Basix: a runtime finite element basis evaluation library*, *Journal of Open Source Software*, 7 (2022), p. 3982.
- [289] S. SELVAKUMAR, K. ARULSHRI, K. PADMANABAN, AND K. SASIKUMAR, *Design and optimization of machining fixture layout using ann and doe*, *The International Journal of Advanced Manufacturing Technology*, 65 (2013), pp. 1573–1586.
- [290] O. SÉRO-GUILLAUME AND J. MARGERIT, *Modelling forest fires. part i: a complete set of equations derived by extended irreversible thermodynamics*, *International Journal of Heat and Mass Transfer*, 45 (2002), pp. 1705–1722.
- [291] J. A. SETHIAN, *A fast marching level set method for monotonically advancing fronts.*, proceedings of the National Academy of Sciences, 93 (1996), pp. 1591–1595.
- [292] ———, *Fast marching methods*, *SIAM review*, 41 (1999), pp. 199–235.

- 
- 
- [293] J. A. SETHIAN, *Level set methods and fast marching methods: evolving interfaces in computational geometry, fluid mechanics, computer vision, and materials science*, vol. 3, Cambridge university press, 1999.
- [294] U. SHAFI, R. MUMTAZ, J. GARCÍA-NIETO, S. A. HASSAN, S. A. R. ZAIDI, AND N. IQBAL, *Precision agriculture techniques and practices: From considerations to applications*, Sensors, 19 (2019), p. 3796.
- [295] J. J. SHIRRON AND I. BABUŠKA, *A comparison of approximate boundary conditions and infinite element methods for exterior helmholtz problems*, Computer Methods in Applied Mechanics and Engineering, 164 (1998), pp. 121–139.
- [296] C.-W. SHU AND S. OSHER, *Efficient implementation of essentially non-oscillatory shock-capturing schemes*, Journal of computational physics, 77 (1988), pp. 439–471.
- [297] O. SIGMUND, *A 99 line topology optimization code written in matlab*, Structural and multidisciplinary optimization, 21 (2001), pp. 120–127.
- [298] O. SIGMUND AND K. MAUTE, *Topology optimization approaches*, Structural and Multidisciplinary Optimization, 48 (2013), pp. 1031–1055.
- [299] C. K. SOH AND J. YANG, *Fuzzy controlled genetic algorithm search for shape optimization*, Journal of computing in civil engineering, 10 (1996), pp. 143–150.
- [300] J. SOKOŁOWSKI AND A. ZOCHOWSKI, *On the topological derivative in shape optimization*, SIAM Journal on Control and Optimization, 37 (1999), pp. 1251–1272.
- [301] J. SOKOŁOWSKI AND J.-P. ZOLÉSIO, *Introduction to shape optimization*, Springer, 1992.
- [302] R. J. SPITERI AND S. J. RUUTH, *A new class of optimal high-order strong-stability-preserving time discretization methods*, SIAM Journal on Numerical Analysis, 40 (2002), pp. 469–491.
- [303] M. SPIVAK, *A comprehensive introduction to differential geometry*, vol. 4, Publish or Perish, Incorporated, 1975.
- [304] J. V. STAFFORD, *Implementing precision agriculture in the 21st century*, Journal of agricultural engineering research, 76 (2000), pp. 267–275.
- [305] E. STEPHAN AND W. WENDLAND, *A boundary integral equation method for three-dimensional crack problems in elasticity*, Mathematical methods in the applied sciences, 8 (1986), pp. 609–623.
- [306] E. P. STEPHAN, *Boundary integral equations for magnetic screens in  $\mathbb{R}^3$* , Proceedings of the Royal Society of Edinburgh Section A: Mathematics, 102 (1986), pp. 189–210.
- [307] ———, *Boundary integral equations for screen problems in  $\mathbb{R}^3$* , Integral Equations and Operator Theory, 10 (1987), pp. 236–257.
- [308] H. A. STONE AND S. KIM, *Microfluidics: basic issues, applications, and challenges*, American Institute of Chemical Engineers. AIChE Journal, 47 (2001), p. 1250.
- [309] J. STRAIN, *Semi-lagrangian methods for level set equations*, Journal of Computational Physics, 151 (1999), pp. 498–533.
- [310] K. STURM, *On shape optimization with non-linear partial differential equations*, Technische Universitaet Berlin (Germany), 2015.
- [311] ———, *A structure theorem for shape functions defined on submanifolds*, Interfaces and Free Boundaries, 18 (2016), pp. 523–543.
- [312] A. L. SULLIVAN, *Wildland surface fire spread modelling, 1990–2007. 2: Empirical and quasi-empirical models*, International Journal of Wildland Fire, 18 (2009), pp. 369–386.
- [313] K. SVANBERG, *The method of moving asymptotes—A new method for structural optimization*, International journal for numerical methods in engineering, 24 (1987), pp. 359–373.
- 
-

- [314] L. A. SZOLGA AND R. T. OPRA, *Robotic arm for biological probe tubes handling*, in 2021 IEEE 27th International Symposium for Design and Technology in Electronic Packaging (SIITME), IEEE, 2021, pp. 317–321.
- [315] P. TANG, F. QIU, H. ZHANG, AND Y. YANG, *Phase separation patterns for diblock copolymers on spherical surfaces: A finite volume method*, Physical Review E, 72 (2005), p. 016710.
- [316] L. TIAN, C. B. MACDONALD, AND S. J. RUUTH, *Segmentation on surfaces with the closest point method*, in 2009 16th IEEE International Conference on Image Processing (ICIP), IEEE, 2009, pp. 3009–3012.
- [317] S. TOWNSEND AND H. A. KIM, *A level set topology optimization method for the buckling of shell structures*, Structural and Multidisciplinary Optimization, 60 (2019), pp. 1783–1800.
- [318] E. A. TRÄFF, O. SIGMUND, AND N. AAGE, *Topology optimization of ultra high resolution shell structures*, Thin-Walled Structures, 160 (2021), p. 107349.
- [319] G. TURK, *Generating textures on arbitrary surfaces using reaction-diffusion*, Acm Siggraph Computer Graphics, 25 (1991), pp. 289–298.
- [320] B. VELICHKOV, *Existence and regularity results for some shape optimization problems*, Springer, 2015.
- [321] P. VICHARE, A. NASSEHI, AND S. T. NEWMAN, *Unified representation of fixtures: clamping, locating and supporting elements in cnc manufacture*, International Journal of Production Research, 49 (2011), pp. 5017–5032.
- [322] C. VILLANI, *Optimal transport: old and new*, vol. 338, Springer, 2009.
- [323] E. WADBRO AND M. BERGGREN, *Topology optimization of an acoustic horn*, Computer methods in applied mechanics and engineering, 196 (2006), pp. 420–436.
- [324] C. WANG, Z. ZHAO, M. ZHOU, O. SIGMUND, AND X. S. ZHANG, *A comprehensive review of educational articles on structural and multidisciplinary optimization*, Structural and Multidisciplinary Optimization, 64 (2021), pp. 2827–2880.
- [325] M. Y. WANG, X. WANG, AND D. GUO, *A level set method for structural topology optimization*, Computer methods in applied mechanics and engineering, 192 (2003), pp. 227–246.
- [326] S. WANG, K. M. LIM, B. C. KHOO, AND M. Y. WANG, *An extended level set method for shape and topology optimization*, Journal of Computational Physics, 221 (2007), pp. 395–421.
- [327] W. WANG, H. RIVARD, AND R. ZMEUREANU, *Floor shape optimization for green building design*, Advanced Engineering Informatics, 20 (2006), pp. 363–378.
- [328] A. WITKIN AND M. KASS, *Reaction-diffusion textures*, in Proceedings of the 18th annual conference on Computer graphics and interactive techniques, 1991, pp. 299–308.
- [329] D. WOLTER, *Spatial representation and reasoning for robot mapping: a shape-based approach*, vol. 48, Springer Science & Business Media, 2008.
- [330] J. WU, O. SIGMUND, AND J. P. GROEN, *Topology optimization of multi-scale structures: a review*, Structural and Multidisciplinary Optimization, (2021), pp. 1–26.
- [331] S. XI, H. CHEN, B. ZHANG, B.-I. WU, AND J. A. KONG, *Route to low-scattering cylindrical cloaks with finite permittivity and permeability*, Physical Review B, 79 (2009), p. 155122.
- [332] Q. XIA AND T. SHI, *Topology optimization of compliant mechanism and its support through a level set method*, Computer Methods in Applied Mechanics and Engineering, 305 (2016), pp. 359–375.
- [333] Q. XIA, M. Y. WANG, AND T. SHI, *A level set method for shape and topology optimization of both structure and support of continuum structures*, Computer Methods in Applied Mechanics and Engineering, 272 (2014), pp. 340–353.

- 
- 
- [334] W. XUDONG, W. Z. SHEN, W. J. ZHU, J. N. SORENSSEN, AND C. JIN, *Shape optimization of wind turbine blades*, Wind Energy: An International Journal for Progress and Applications in Wind Power Conversion Technology, 12 (2009), pp. 781–803.
- [335] X. YAN, J. ZHU, M. KUANG, AND X. WANG, *Aerodynamic shape optimization using a novel optimizer based on machine learning techniques*, Aerospace Science and Technology, 86 (2019), pp. 826–835.
- [336] R. YANG, C.-H. CHUANG, X. CHE, AND C. SOTO, *New applications of topology optimisation in automotive industry*, International Journal of Vehicle Design, 23 (2000), pp. 1–15.
- [337] Z. YANG AND X. HUANG, *An acoustic cloaking design based on topology optimization*, The Journal of the Acoustical Society of America, 152 (2022), pp. 3510–3521.
- [338] Q. YE, Y. GUO, S. CHEN, N. LEI, AND X. D. GU, *Topology optimization of conformal structures on manifolds using extended level set methods (x-lsm) and conformal geometry theory*, Computer Methods in Applied Mechanics and Engineering, 344 (2019), pp. 164–185.
- [339] T. YOKOZEKI, S.-I. TAKEDA, T. OGASAWARA, AND T. ISHIKAWA, *Mechanical properties of corrugated composites for candidate materials of flexible wing structures*, Composites Part A: applied science and manufacturing, 37 (2006), pp. 1578–1586.
- [340] M. YULIN AND W. XIAOMING, *A level set method for structural topology optimization and its applications*, Advances in Engineering software, 35 (2004), pp. 415–441.
- [341] A. ZAHARESCU, E. BOYER, AND R. HORAUD, *Topology-adaptive mesh deformation for surface evolution, morphing, and multiview reconstruction*, IEEE Transactions on Pattern Analysis and Machine Intelligence, 33 (2010), pp. 823–837.
- [342] S.-E. ZERROUQ AND M. PIERRE, *Robust optimal ship hulls based on michell’s wave resistance*, Mathematical Methods in the Applied Sciences, 47 (2024), pp. 928–960.
- [343] D. ZHANG AND G. LU, *Review of shape representation and description techniques*, Pattern recognition, 37 (2004), pp. 1–19.
- [344] J.-B. ZHANG, G.-W. HE, AND F. LIU, *Electro-osmotic flow and mixing in heterogeneous microchannels*, Physical Review E, 73 (2006), p. 056305.
- [345] W. ZHANG, L. ZHAO, AND S. CAI, *Shape optimization of dirichlet boundaries based on weighted b-spline finite cell method and level-set function*, Computer Methods in Applied Mechanics and Engineering, 294 (2015), pp. 359–383.
- [346] W. ZHANG AND B. ZHU, *Optimization design for slip/no-slip configuration of hydrophobic sliding bearings using monte carlo search*, Tribology International, 178 (2023), p. 108034.
- [347] Y. ZHANG, H. CHEN, I. MEZIC, C. MEINHART, L. PETZOLD, AND N. MACDONALD, *Soi processing of a ring electrokinetic chaotic micromixer*, in Proc NSTI Nanotechnology Conference and Trade Show (Nanotech 2004), vol. 1, 2004, pp. 292–295.
- [348] Y.-T. ZHANG AND C.-W. SHU, *Eno and weno schemes*, in Handbook of numerical analysis, vol. 17, Elsevier, 2016, pp. 103–122.
- [349] H. ZHAO, *A fast sweeping method for eikonal equations*, Mathematics of computation, 74 (2005), pp. 603–627.
- [350] H.-K. ZHAO, S. OSHER, AND R. FEDKIW, *Fast surface reconstruction using the level set method*, in Variational and Level Set Methods in Computer Vision, 2001. Proceedings. IEEE Workshop on, IEEE, 2001, pp. 194–201.
- [351] Z.-C. ZHONG, S.-H. WEI, J.-P. WANG, C.-K. FENG, C.-S. CHEN, AND C.-H. YU, *Finite element analysis of the lumbar spine with a new cage using a topology optimization method*, Medical engineering & physics, 28 (2006), pp. 90–98.
- [352] B. ZHU, B. RAO, J. JIA, AND Y. LI, *Shape optimization of arch dams for static and dynamic loads*, Journal of structural engineering, 118 (1992), pp. 2996–3015.
- 
-

- [353] J.-H. ZHU, W.-H. ZHANG, AND L. XIA, *Topology optimization in aircraft and aerospace structures design*, Archives of computational methods in engineering, 23 (2016), pp. 595–622.
- [354] J.-P. ZOLÉSIO, *Un resultat d'existence de vitesse convergente dans des problemes d'identification de domaine.*, (1976).
- [355] ———, *The material derivative (or speed) method for shape optimization*, Optimization of distributed parameter structures, 2 (1981), pp. 1089–1151.

# Appendix A

## Supporting material

---

### A.1 Useful identities for the computation of shape derivatives

The following identities are obtained by use of Green's formulas ([Proposition C.1](#)) and are used to remove the volumetric terms from the shape derivative expression. We assume that  $\Omega \subset \mathbb{R}^n$  is a bounded Lipschitz domain.

**Proposition A.1.** *Let  $j \in C^1(\mathbb{R}^n)$ ,  $\gamma \in C^\infty(\Omega)$ ,  $u \in H^2(\Omega)$ , and  $\theta \in W^{1,\infty}(\mathbb{R}^d, \mathbb{R}^d)$ . Then all the following identities are valid:*

1.  $\int_{\Omega} j(\nabla u) \nabla \cdot \theta \, dx = \int_{\partial\Omega} j(\nabla u) \theta \cdot n \, ds - \int_{\Omega} \theta \cdot \nabla^2 u j'(\nabla u),$
2.  $\int_{\Omega} \nabla \theta^T \nabla u \cdot j'(\nabla u) \, dx = \int_{\partial\Omega} (\theta \cdot \nabla u) j'(\nabla u) \cdot n \, ds - \int_{\Omega} \nabla \cdot j'(\nabla u) \theta \cdot \nabla u \, dx - \int_{\Omega} \nabla^2 u j'(\nabla u) \cdot \theta \, dx,$
3.  $\int_{\Omega} (\nabla \cdot \theta) (\gamma \nabla u \cdot \nabla p) \, dx = \int_{\partial\Omega} \gamma \nabla u \cdot \nabla p \theta \cdot n \, ds - \int_{\Omega} \gamma \nabla (\nabla u \cdot \nabla p) \, dx - \int_{\Omega} \nabla u \cdot \nabla p \nabla \gamma \cdot \theta \, dx,$
4.  $\int_{\Omega} \nabla \theta (\gamma \nabla u) \cdot \nabla p \, dx = \int_{\partial\Omega} \theta \cdot \nabla p \gamma \frac{\partial u}{\partial n} \, ds - \int_{\Omega} \nabla^2 p \gamma \nabla u \cdot \theta \, dx,$
5.  $\int_{\Omega} \nabla \theta^T (\gamma \nabla u) \cdot \nabla p \, dx = \int_{\partial\Omega} \theta \cdot \nabla u \gamma \frac{\partial p}{\partial n} \, ds - \int_{\Omega} \nabla \cdot j'(\nabla u) \theta \cdot \nabla u \, dx - \int_{\Omega} \nabla^2 u (\gamma \nabla p) \cdot \theta \, dx.$

### A.2 Proofs

#### A.2.1 Proof of [Proposition 4.4](#)

*Proof.* For simplicity we set  $u_\varepsilon = u_{C,A,\varepsilon}$  solution to [\(4.82\)](#).

**Step 1.** We compute the Lagrangian derivative. For  $\theta \in \Theta$  with norm  $\|\theta\|_{W^{1,\infty}(\mathbb{R}^d, \mathbb{R}^d)} < 1$ , the function  $u_{\Omega_\theta, \varepsilon} \in H^1(\Omega_\theta)$  is the unique solution to the variational problem:

$$\begin{aligned} \forall v \in H^1(\Omega_\theta), \quad \int_{\Omega_\theta} \gamma \nabla u_{\Omega_\theta, \varepsilon} \cdot \nabla v \, dx + \int_{\partial\Omega_\theta} \frac{1}{\varepsilon} \left( h \left( \frac{d_{\Gamma_{A\theta}}^{\partial\Omega_\theta}}{\varepsilon} \right) + h \left( \frac{d_{\Gamma_{C\theta}}^{\partial\Omega_\theta}}{\varepsilon} \right) \right) u_{\Omega_\theta, \varepsilon} v \, ds = \\ \int_{\partial\Omega_\theta} \frac{1}{\varepsilon} h \left( \frac{d_{\Gamma_{A\theta}}}{\varepsilon} \right) u_{\text{in}} v \, ds, \end{aligned}$$

Let us introduce the transported mapping:

$$\begin{aligned} W^{1,\infty}(\mathbb{R}^d, \mathbb{R}^d) &\rightarrow H^1(\Omega) \\ \theta &\mapsto \bar{u}_\varepsilon(\theta) := u_{\Omega_\theta, \varepsilon} \circ (\text{Id} + \theta). \end{aligned}$$

Performing a change of variables (see [Corollaries C.1](#) and [C.2](#)) and choosing test functions of the form  $v \circ (\text{Id} + \theta)^{-1}$ ,  $v \in H^1(\Omega)$ , we obtain the following variational characterization for  $\overline{u_\varepsilon}(\theta)$ . For all  $v \in H^1(\Omega)$ :

$$\begin{aligned} & \int_{\Omega} A(\theta) \nabla \overline{u_\varepsilon}(\theta) \cdot \nabla v \, dx \\ & + \int_{\partial\Omega} \frac{1}{\varepsilon} \left( h \left( \frac{d_{\Gamma_{A\theta}}^{\partial\Omega} \circ (\text{Id} + \theta)}{\varepsilon} \right) + h \left( \frac{d_{\Gamma_{C\theta}}^{\partial\Omega} \circ (\text{Id} + \theta)}{\varepsilon} \right) \right) \det(I + \nabla\theta) |(I + \nabla\theta)^{-T} n| \overline{u_\varepsilon}(\theta) v \, ds \\ & = \int_{\partial\Omega} \frac{1}{\varepsilon} h \left( \frac{d_{\Gamma_{A\theta}}^{\partial\Omega} \circ (\text{Id} + \theta)}{\varepsilon} \right) \det(I + \nabla\theta) |(I + \nabla\theta)^{-T} n| u_{\text{in}} \circ (\text{Id} + \theta) v \, ds, \end{aligned}$$

where:

$$A(\theta) := \gamma \circ (\text{Id} + \theta) |\det(I + \nabla\theta)| (I + \nabla\theta)^{-1} (I + \nabla\theta)^{-T}.$$

The Fréchet differentiability of the map  $\theta \mapsto \overline{u_\varepsilon}(\theta)$  is typically established via the implicit function theorem (see [\[15, 184\]](#)) but for the sake of simplicity we omit this part of the proof. Differentiating both sides with respect to  $\theta$  at 0 yields the characterization for the ‘‘Lagrangian’’ derivative  $\dot{u}_{\Omega,\varepsilon}(\theta)$  of  $u_{\Omega,\varepsilon}$ , i.e. the derivative of the mapping  $\theta \mapsto \overline{u_\varepsilon}(\theta)$ , for all  $v \in H^1(\Omega)$ ,

$$\begin{aligned} & \int_{\Omega} \gamma \nabla \dot{u}_{\Omega,\varepsilon}(\theta) \cdot \nabla v \, dx + \int_{\partial\Omega} (h_{C,\varepsilon} + h_{A,\varepsilon}) \dot{u}_\varepsilon(\theta) v \, ds = \\ & - \int_{\Omega} (\nabla u_\varepsilon \cdot \nabla v) (\nabla \gamma \cdot \theta) \, dx - \int_{\Omega} (\nabla \cdot \theta \text{I} - \nabla\theta - \nabla\theta^T) \gamma \nabla u_\varepsilon \cdot \nabla v \, dx \\ & - \int_{\partial\Omega} (\nabla_{\partial\Omega} \cdot \theta) (h_{C,\varepsilon} + h_{A,\varepsilon}) u_\varepsilon v \, ds \\ & - \frac{1}{\varepsilon^2} \int_{\partial\Omega} \left( h' \left( \frac{d_{\Gamma_C}^{\partial\Omega}}{\varepsilon} \right) D'_C(0)(\theta) + h' \left( \frac{d_{\Gamma_A}^{\partial\Omega}}{\varepsilon} \right) D'_A(0)(\theta) \right) u_\varepsilon v \, ds \\ & + \int_{\partial\Omega} ((\nabla_{\partial\Omega} \cdot \theta) u_{\text{in}} + \nabla u_{\text{in}} \cdot \theta) h_{A,\varepsilon} v \, ds \\ & + \frac{1}{\varepsilon^2} \int_{\partial\Omega} h' \left( \frac{d_{\Gamma_A}^{\partial\Omega}}{\varepsilon} \right) D'_A(0)(\theta) u_{\text{in}} v \, ds, \end{aligned} \tag{A.1}$$

where we have used the following facts (see [\[184\]](#) for their statement and proof):

$$\begin{aligned} & \left. \frac{d}{d\theta} \right|_{\theta=0} \det(I + \nabla\theta) = \nabla \cdot \theta, \\ & \left. \frac{d}{d\theta} \right|_{\theta=0} \det(I + \nabla\theta) |(I + \nabla\theta)^{-T} n_{\partial\Omega}| = \nabla_{\partial\Omega} \cdot \theta. \end{aligned}$$

**Step 2.** Shape derivative of  $J_\varepsilon(\Omega)$ . We now calculate the derivative of the objective function  $J_\varepsilon(\Omega)$ ; for sufficiently small  $\theta \in W^{1,\infty}(\mathbb{R}^d, \mathbb{R}^d)$ , it holds:

$$\begin{aligned} J_\varepsilon(\Omega_\theta) & = \int_{\Omega} |\det(I + \nabla\theta)| j((\nabla u_{\Omega_\theta,\varepsilon}) \circ (\text{Id} + \theta)) \, dx \\ & = \int_{\Omega} |\det(I + \nabla\theta)| j((I + \nabla\theta)^{-T} \nabla \overline{u_\varepsilon}(\theta)) \, dx. \end{aligned}$$

Hence, taking derivatives in this formula, we obtain:

$$J'_\varepsilon(\Omega)(\theta) = \int_{\Omega} j'(\nabla u_\varepsilon) \cdot \nabla \dot{u}_{\Omega,\varepsilon}(\theta) \, dx + \int_{\Omega} j(\nabla u_\varepsilon) \nabla \cdot \theta \, dx - \int_{\Omega} \nabla\theta^T \nabla u_\varepsilon \cdot j'(\nabla u_\varepsilon) \, dx.$$

**Step 3.** We reformulate this derivative by using the adjoint state. Consider the weak formulation for the adjoint solution  $p_\varepsilon \in H^1(\Omega)$ , given by:

$$\forall v \in H^1(\Omega), \quad \int_{\Omega} \gamma \nabla p_\varepsilon \cdot \nabla v \, dx + \int_{\partial\Omega} (h_{C,\varepsilon} + h_{A,\varepsilon}) p_\varepsilon v \, ds = - \int_{\Omega} j'(\nabla u_\varepsilon) \cdot \nabla v \, dx.$$

Selecting  $v = \dot{u}_\varepsilon(\theta)$  as a test function results in the following expression:

$$\int_{\Omega} \gamma \nabla p_\varepsilon \cdot \nabla \dot{u}_\varepsilon(\theta) \, dx + \int_{\partial\Omega} (h_{C,\varepsilon} + h_{A,\varepsilon}) p_\varepsilon \dot{u}_\varepsilon(\theta) \, ds = - \int_{\Omega} j'(\nabla u_\varepsilon) \cdot \nabla \dot{u}_\varepsilon(\theta) \, dx.$$

On the other hand, opting for  $v = p_\varepsilon$  as a test function in (A.1) and combining it with the previous equation leads to the volumetric expression of the shape derivative:

$$\begin{aligned} J'(\Omega)(\theta) &= \int_{\Omega} j(\nabla u_{\Omega,\varepsilon}) \nabla \cdot \theta \, dx - \int_{\Omega} \nabla \theta^T \nabla u_\varepsilon \cdot j'(\nabla u_\varepsilon) \, dx + \int_{\Omega} (\nabla u_\varepsilon \cdot \nabla p)(\nabla \gamma \cdot \theta) \, dx \\ &\quad + \int_{\Omega} (\nabla \cdot \theta I - \nabla \theta - \nabla \theta^T) \gamma \nabla u_\varepsilon \cdot \nabla p_\varepsilon \, dx \\ &\quad - \int_{\partial\Omega} ((\nabla_{\partial\Omega} \cdot \theta) u_{\text{in}} + \nabla u_{\text{in}} \cdot \theta) h_{A,\varepsilon} p_\varepsilon \, ds \\ &\quad + \int_{\partial\Omega} (\nabla_{\partial\Omega} \cdot \theta) (h_{C,\varepsilon} + h_{A,\varepsilon}) u_\varepsilon p_\varepsilon \, ds \\ &\quad + \frac{1}{\varepsilon^2} \int_{\partial\Omega} \left( h' \left( \frac{d_{\Gamma_C}}{\varepsilon} \right) D'_C(0)(\theta) + h' \left( \frac{d_{\Gamma_A}}{\varepsilon} \right) D'_A(0)(\theta) \right) u_\varepsilon p_\varepsilon \, ds \\ &\quad - \frac{1}{\varepsilon^2} \int_{\partial\Omega} h' \left( \frac{d_{\Gamma_A}}{\varepsilon} \right) D'_A(0)(\theta) u_{\text{in}} p_\varepsilon \, ds, \end{aligned} \tag{A.2}$$

**Step 4.** Derivation of the surface expression. Following multiple applications of integration by parts (refer to [Appendix A.1](#)), one can verify the cancellation of the volumetric terms, resulting in the following expression for the shape derivative:

$$\begin{aligned} J'_\varepsilon(\Omega)(\theta) &= \int_{\partial\Omega} (j(\nabla u_\varepsilon) - \gamma \nabla u_\varepsilon \cdot \nabla p_\varepsilon) \theta \cdot n_{\partial\Omega} \, ds - (I_\varepsilon^1 - I_\varepsilon^2 + I_\varepsilon^3) \\ &\quad + \frac{1}{\varepsilon^2} \int_{\partial\Omega} \left( h' \left( \frac{d_{\Gamma_C}^{\partial\Omega}}{\varepsilon} \right) D'_C(0)(\theta) + h' \left( \frac{d_{\Gamma_A}^{\partial\Omega}}{\varepsilon} \right) D'_A(0)(\theta) \right) u_\varepsilon p_\varepsilon \, ds \\ &\quad - \frac{1}{\varepsilon^2} \int_{\partial\Omega} h' \left( \frac{d_{\Gamma_A}^{\partial\Omega}}{\varepsilon} \right) D'_A(0)(\theta) u_{\text{in}} p_\varepsilon \, ds, \end{aligned}$$

where

$$\begin{aligned} I_\varepsilon^1 &:= \int_{\partial\Omega} ((\nabla_{\partial\Omega} \cdot \theta) u_{\text{in}} + \nabla u_{\text{in}} \cdot \theta) h_{A,\varepsilon} p_\varepsilon \, ds, \\ I_\varepsilon^2 &:= \int_{\partial\Omega} (\nabla_{\partial\Omega} \cdot \theta) (h_{C,\varepsilon} + h_{A,\varepsilon}) u_\varepsilon p_\varepsilon \, ds, \\ I_\varepsilon^3 &:= \int_{\partial\Omega} \left( \theta \cdot \nabla p_\varepsilon \gamma \frac{\partial u_\varepsilon}{\partial n_{\partial\Omega}} \, ds + \theta \cdot \nabla u_\varepsilon \left( \gamma \frac{\partial p_\varepsilon}{\partial n_{\partial\Omega}} - j'(\nabla u_\varepsilon) \cdot n_{\partial\Omega} \right) \right) \, ds. \end{aligned}$$

The integration of this expression towards tangential terms can be accomplished by employing [Proposition C.2](#) and decomposing:

$$\begin{aligned} \nabla_{\partial\Omega} u_{\text{in}} &= \nabla u_{\text{in}} - \frac{\partial u_{\text{in}}}{\partial n_{\partial\Omega}} n_{\partial\Omega}, \\ \nabla_{\partial\Omega} u_\varepsilon &= \nabla u_\varepsilon - \frac{\partial u_\varepsilon}{\partial n_{\partial\Omega}} n_{\partial\Omega}, \\ \nabla_{\partial\Omega} p_\varepsilon &= \nabla p_\varepsilon - \frac{\partial p_\varepsilon}{\partial n_{\partial\Omega}} n_{\partial\Omega}. \end{aligned} \tag{A.3}$$

Firstly, tangentially integrating the first term of  $I_\varepsilon^1$  by parts gives:

$$\begin{aligned} I_\varepsilon^1 &= \int_{\partial\Omega} \kappa h_{A,\varepsilon} u_{\text{in}} p_\varepsilon \theta \cdot n_{\partial\Omega} \, ds + \int_{\partial\Omega} h_{A,\varepsilon} p_\varepsilon \frac{\partial u_{\text{in}}}{\partial n_{\partial\Omega}} \theta \cdot n_{\partial\Omega} \, ds \\ &\quad - \int_{\partial\Omega} u_{\text{in}} p_\varepsilon \nabla_{\partial\Omega} h_{A,\varepsilon} \cdot \theta \, ds - \int_{\partial\Omega} h_{A,\varepsilon} u_{\text{in}} \nabla_{\partial\Omega} p_\varepsilon \cdot \theta \, ds. \end{aligned}$$

Next, integrating  $I_\varepsilon^2$  yields:

$$\begin{aligned} I_\varepsilon^2 &= \int_{\partial\Omega} \kappa(h_{A,\varepsilon} + h_{C,\varepsilon})u_\varepsilon p_\varepsilon \theta \cdot n_{\partial\Omega} \, ds \\ &\quad - \int_{\partial\Omega} (h_{A,\varepsilon} + h_{C,\varepsilon})p_\varepsilon \nabla_{\partial\Omega} u_\varepsilon \cdot \theta \, ds - \int_{\partial\Omega} (h_{A,\varepsilon} + h_{C,\varepsilon})u_\varepsilon \nabla_{\partial\Omega} p_\varepsilon \cdot \theta \, ds \\ &\quad - \int_{\partial\Omega} \nabla_{\partial\Omega}(h_{A,\varepsilon} + h_{C,\varepsilon}) \cdot \theta \, u_\varepsilon p_\varepsilon \, ds. \end{aligned}$$

Additionally, we can substitute the definitions of  $\gamma \frac{\partial u_\varepsilon}{\partial n_{\partial\Omega}}$  and  $\gamma \frac{\partial p_\varepsilon}{\partial n_{\partial\Omega}}$  into  $I_\varepsilon^3$  and decompose them tangentially:

$$\begin{aligned} I_\varepsilon^3 &= \int_{\partial\Omega} h_{A,\varepsilon} u_{\text{in}} \left( \nabla_{\partial\Omega} p_\varepsilon + \frac{\partial p_\varepsilon}{\partial n_{\partial\Omega}} n_{\partial\Omega} \right) \cdot \theta \, ds \\ &\quad - \int_{\partial\Omega} (h_{A,\varepsilon} + h_{C,\varepsilon}) u_\varepsilon \left( \nabla_{\partial\Omega} p_\varepsilon + \frac{\partial p_\varepsilon}{\partial n_{\partial\Omega}} n_{\partial\Omega} \right) \cdot \theta \, ds - \int_{\partial\Omega} (h_{A,\varepsilon} + h_{C,\varepsilon}) p_\varepsilon \left( \nabla_{\partial\Omega} u_\varepsilon + \frac{\partial u_\varepsilon}{\partial n_{\partial\Omega}} n_{\partial\Omega} \right) \cdot \theta \, ds \end{aligned}$$

Computing now  $I_\varepsilon^1 - I_\varepsilon^2 + I_\varepsilon^3$  yields:

$$\begin{aligned} I_\varepsilon^1 - I_\varepsilon^2 + I_\varepsilon^3 &= \int_{\partial\Omega} \kappa h_{A,\varepsilon} u_{\text{in}} p_\varepsilon \theta \cdot n_{\partial\Omega} \, ds - \int_{\partial\Omega} \kappa (h_{A,\varepsilon} + h_{C,\varepsilon}) u_\varepsilon p_\varepsilon \theta \cdot n_{\partial\Omega} \, ds \\ &\quad + \int_{\partial\Omega} h_{A,\varepsilon} p_\varepsilon \frac{\partial u_{\text{in}}}{\partial n_{\partial\Omega}} \theta \cdot n_{\partial\Omega} \, ds - \int_{\partial\Omega} (h_{A,\varepsilon} + h_{C,\varepsilon}) p_\varepsilon \frac{\partial u_\varepsilon}{\partial n_{\partial\Omega}} \theta \cdot n_{\partial\Omega} \, ds \\ &\quad + \int_{\partial\Omega} h_{A,\varepsilon} u_{\text{in}} \frac{\partial p_\varepsilon}{\partial n_{\partial\Omega}} \theta \cdot n_{\partial\Omega} \, ds - \int_{\partial\Omega} (h_{A,\varepsilon} + h_{C,\varepsilon}) u_\varepsilon \frac{\partial p_\varepsilon}{\partial n_{\partial\Omega}} \theta \cdot n_{\partial\Omega} \, ds \\ &\quad - \int_{\partial\Omega} u_{\text{in}} p_\varepsilon \nabla_{\partial\Omega} h_{A,\varepsilon} \cdot \theta \, ds + \int_{\partial\Omega} u_\varepsilon p_\varepsilon \nabla_{\partial\Omega} (h_{A,\varepsilon} + h_{C,\varepsilon}) \cdot \theta \, ds. \end{aligned}$$

Recall that by (4.82) we have:

$$\gamma \frac{\partial u_\varepsilon}{\partial n_{\partial\Omega}} = h_{A,\varepsilon} u_{\text{in}} - (h_{C,\varepsilon} + h_{A,\varepsilon}) u_\varepsilon \quad \text{on } \partial\Omega,$$

hence we can identify  $\gamma \frac{\partial u_\varepsilon}{\partial n_{\partial\Omega}}$  in the sum to see:

$$\begin{aligned} I_\varepsilon^1 - I_\varepsilon^2 + I_\varepsilon^3 &= \int_{\partial\Omega} \kappa \gamma \frac{\partial u_\varepsilon}{\partial n_{\partial\Omega}} h_{A,\varepsilon} p_\varepsilon \theta \cdot n_{\partial\Omega} \, ds \\ &\quad + \int_{\partial\Omega} h_{A,\varepsilon} p_\varepsilon \frac{\partial u_{\text{in}}}{\partial n_{\partial\Omega}} \theta \cdot n_{\partial\Omega} \, ds - \int_{\partial\Omega} (h_{A,\varepsilon} + h_{C,\varepsilon}) p_\varepsilon \frac{\partial u_\varepsilon}{\partial n_{\partial\Omega}} \theta \cdot n_{\partial\Omega} \, ds \\ &\quad + \int_{\partial\Omega} \gamma \frac{\partial u_\varepsilon}{\partial n_{\partial\Omega}} \frac{\partial p_\varepsilon}{\partial n_{\partial\Omega}} \theta \cdot n_{\partial\Omega} \, ds \\ &\quad - \int_{\partial\Omega} u_{\text{in}} p_\varepsilon \nabla_{\partial\Omega} h_{A,\varepsilon} \cdot \theta \, ds + \int_{\partial\Omega} u_\varepsilon p_\varepsilon \nabla_{\partial\Omega} (h_{A,\varepsilon} + h_{C,\varepsilon}) \cdot \theta \, ds. \end{aligned}$$

Substituting back in results in the expression taking the form:

$$\begin{aligned} J'(\Omega)(\theta) &= \int_{\partial\Omega} (j(\nabla u_\varepsilon) - \gamma \nabla u_\varepsilon \cdot \nabla p_\varepsilon) \theta \cdot n_{\partial\Omega} \, ds - \int_{\partial\Omega} \kappa \gamma \frac{\partial u_\varepsilon}{\partial n_{\partial\Omega}} h_{A,\varepsilon} p_\varepsilon \theta \cdot n_{\partial\Omega} \, ds \\ &\quad - \int_{\partial\Omega} h_{A,\varepsilon} p_\varepsilon \frac{\partial u_{\text{in}}}{\partial n_{\partial\Omega}} \theta \cdot n_{\partial\Omega} \, ds + \int_{\partial\Omega} (h_{A,\varepsilon} + h_{C,\varepsilon}) p_\varepsilon \frac{\partial u_\varepsilon}{\partial n_{\partial\Omega}} \theta \cdot n_{\partial\Omega} \, ds - \int_{\partial\Omega} \gamma \frac{\partial u_\varepsilon}{\partial n_{\partial\Omega}} \frac{\partial p_\varepsilon}{\partial n_{\partial\Omega}} \theta \cdot n_{\partial\Omega} \, ds \\ &\quad + \int_{\partial\Omega} u_{\text{in}} p_\varepsilon \nabla_{\partial\Omega} h_{A,\varepsilon} \cdot \theta \, ds - \int_{\partial\Omega} u_\varepsilon p_\varepsilon \nabla_{\partial\Omega} (h_{A,\varepsilon} + h_{C,\varepsilon}) \cdot \theta \, ds \\ &\quad + \frac{1}{\varepsilon^2} \int_{\partial\Omega} \left( h' \left( \frac{d_{\Gamma C}^{\partial\Omega}}{\varepsilon} \right) D'_C(0)(\theta) + h' \left( \frac{d_{\Gamma A}^{\partial\Omega}}{\varepsilon} \right) D'_A(0)(\theta) \right) u_\varepsilon p_\varepsilon \, ds \\ &\quad - \frac{1}{\varepsilon^2} \int_{\partial\Omega} h' \left( \frac{d_{\Gamma A}^{\partial\Omega}}{\varepsilon} \right) D'_A(0)(\theta) u_{\text{in}} p_\varepsilon \, ds. \end{aligned}$$

We can further simplify the last three lines of this new expression. Via uses of [Lemma 2.5](#) and [Lemma 2.3](#) for the region  $\Gamma_A$  (and, respectively, for the region  $\Gamma_C$ ) we can see that:

$$\begin{aligned} & \nabla_{\partial\Omega} h_{A,\varepsilon} \cdot \theta - \frac{1}{\varepsilon^2} h' \left( \frac{d_{\Gamma_A}^{\partial\Omega}}{\varepsilon} \right) D'_A(0)(\theta) \\ &= -\frac{1}{\varepsilon^2} h' \left( \frac{d_{\Gamma_A}^{\partial\Omega}}{\varepsilon} \right) \frac{\log_x(p_{\Sigma_A}(x))}{d_{\Gamma_A}^{\partial\Omega}(x)} \cdot \theta - \frac{1}{\varepsilon^2} h' \left( \frac{d_{\Gamma_A}^{\partial\Omega}}{\varepsilon} \right) D'_A(0)(\theta) \\ &= \frac{1}{\varepsilon^2} h' \left( \frac{d_{\Gamma_A}^{\partial\Omega}}{\varepsilon} \right) \left( \theta(p_{\Sigma_A}) \cdot n_{\Sigma_A}(p_{\Sigma_A}) - \int_0^{d_G^S(p)} \theta(\sigma(t)) \cdot \mathbb{I}(\sigma'(t), \sigma'(t)) dt \right), \end{aligned}$$

which, upon substitution, and after imposing  $\theta \cdot n_{\partial\Omega} = 0$  leads to the desired expression in [Proposition 4.4](#).  $\blacksquare$

## A.2.2 Proof of [Proposition 4.6](#)

*Sketch of proof.* We give the main outline of the proof since it's very similar to the proof of [Proposition 4.4](#). We assume that  $\theta = 0$  outside of the box delimited by  $\Gamma$  and denote  $j(u) = |u|^2 = \bar{u}u$ .

**Step 1.** We compute the Lagrangian derivative. For  $\theta \in \Theta$  with norm  $\|\theta\|_{W^{1,\infty}(\mathbb{R}^d, \mathbb{R}^d)} < 1$ , the function  $u_{\Omega_\theta} \in H^1(\Omega_\theta)$  is the unique solution to the variational problem:

$$\forall v \in H^1(D \setminus \bar{\Omega}_\theta; \mathbb{C}), \quad \int_{D \setminus \bar{\Omega}_\theta} \gamma \nabla u_{\Omega_\theta} \cdot \bar{\nabla} v \, dx - \omega^2 \int_{D \setminus \bar{\Omega}_\theta} u_{\Omega_\theta} \bar{v} \, dx - \left( i\omega - \frac{1}{R} \right) \int_{\Gamma_E} u_{\Omega_\theta} \bar{v} \, ds - \frac{i\omega}{z} \int_{(\Gamma_R)_\theta} u_{\Omega_\theta} \bar{v} \, ds = \int_{\partial\Omega_\theta} \gamma \frac{\partial f}{\partial n_{\partial\Omega_\theta}} \bar{v} \, ds + \frac{i\omega}{z} \int_{(\Gamma_R)_\theta} f \bar{v} \, ds,$$

where  $\bar{v}$  is the conjugate of  $v$ , and  $\bar{\nabla} v$  is the conjugate of its gradient. After a few calculations we arrive at the following characterization for the Lagrangian derivative.

$$\begin{aligned} & \forall v \in H^1(D \setminus \bar{\Omega}_\theta; \mathbb{C}), \\ & \int_{D \setminus \bar{\Omega}} \gamma \nabla \hat{u}_\Omega(\theta) \cdot \bar{\nabla} v \, dx - \omega^2 \int_{D \setminus \bar{\Omega}} \hat{u}_\Omega(\theta) \bar{v} \, dx + \left( i\omega - \frac{1}{R} \right) \int_{\Gamma_E} \hat{u}_\Omega(\theta) \bar{v} \, ds - \frac{i\omega}{z} \int_{\Gamma_R} \hat{u}_\Omega(\theta) \bar{v} \, ds = \\ & - \int_{\Omega} (\nabla \gamma \cdot \theta) (\nabla u_\Omega \cdot \bar{\nabla} v) \, dx - \int_{\Omega} (\nabla \cdot \theta \mathbf{I} - \nabla \theta - \nabla \theta^T) \gamma \nabla u_\Omega \cdot \bar{\nabla} v \, dx \\ & + \omega^2 \int_{D \setminus \bar{\Omega}} (\nabla \cdot \theta) u_\Omega \bar{v} \, dx + \frac{i\omega}{z} \int_{\Gamma_R} (\nabla_{\partial\Omega} \cdot \theta) u_\Omega \bar{v} \, ds + \frac{i\omega}{z} \int_{\Gamma_R} \nabla_{\partial\Omega} \cdot (f\theta) \bar{v} \, ds \\ & + \frac{d}{d\theta} \Big|_{\theta=0} \left[ \int_{\partial\Omega} (\gamma \circ (\text{Id} + \theta)) \left( \frac{\partial f}{\partial n_{\partial\Omega}} \circ (\text{Id} + \theta) \right) |\text{com}(I + \nabla \theta) n_{\partial\Omega}| \bar{v} \, ds \right], \end{aligned} \tag{A.4}$$

**Step 2.** Shape derivative of  $J(\Omega)$ . We now calculate the derivative of the objective function  $J(\Omega)$ ; for sufficiently small  $\theta \in W^{1,\infty}(\mathbb{R}^d, \mathbb{R}^d)$ , it holds:

$$J'(\Omega)(\theta) = \int_{\Omega} (\nabla \cdot \theta) j(u_\Omega) \, dx + \text{Re} \left( \int_{\Omega} j'(u_\Omega) \bar{\hat{u}}_\Omega(\theta) \, dx \right)$$

**Step 3.** We reformulate this derivative by using the adjoint state. The weak formulation for  $p \in H^1(\Omega; \mathbb{C})$  is:

$$\forall v \in H^1(D \setminus \bar{\Omega}; \mathbb{C}), \quad \int_{D \setminus \bar{\Omega}} \gamma \nabla p \cdot \bar{\nabla} v \, dx - \omega^2 \int_{D \setminus \bar{\Omega}} p \bar{v} \, dx - \left( -i\omega - \frac{1}{R} \right) \int_{\Gamma_E} p \bar{v} \, ds + \frac{i\omega}{z} \int_{\Gamma_R} p \bar{v} \, ds = - \int_{\Omega} j'(u_\Omega) \bar{v} \, dx,$$

Taking the complex conjugate on both sides and choosing  $v = \hat{u}_\Omega(\theta)$ , results in

$$\int_{D \setminus \bar{\Omega}} \gamma \nabla \hat{u}_\Omega(\theta) \cdot \bar{\nabla} p \, dx - \omega^2 \int_{D \setminus \bar{\Omega}} \hat{u}_\Omega(\theta) \bar{p} \, dx - \left( i\omega - \frac{1}{R} \right) \int_{\Gamma_E} \hat{u}_\Omega(\theta) \bar{p} \, ds - \frac{i\omega}{z} \int_{\Gamma_R} \gamma \hat{u}_\Omega(\theta) \bar{p} \, ds = - \int_{\Omega} \hat{u}_\Omega(\theta) \overline{j'(u_\Omega)} \, dx.$$

Now we can recognize the left hand side of the equation above, by choosing  $v = p$  in (A.4), so that,

$$\begin{aligned} - \int_{\Omega} \dot{u}_{\Omega}(\theta) \overline{j'(u_{\Omega})} dx &= - \int_{\Omega} (\nabla \gamma \cdot \theta) (\nabla u_{\Omega} \cdot \overline{\nabla p}) dx - \int_{\Omega} (\nabla \cdot \theta I - \nabla \theta - \nabla \theta^T) \gamma \nabla u_{\Omega} \cdot \overline{\nabla p} dx \\ &\quad + \omega^2 \int_{D \setminus \overline{\Omega}} (\nabla \cdot \theta) u_{\Omega} \overline{p} dx + \frac{i\omega}{z} \int_{\Gamma_R} (\nabla_{\partial\Omega} \cdot \theta) u_{\Omega} \overline{p} ds + \frac{i\omega}{z} \int_{\Gamma_R} \nabla_{\partial\Omega} \cdot (f\theta) \overline{p} ds \\ &\quad + \frac{d}{d\theta} \Big|_{\theta=0} \left[ \int_{\partial\Omega} (\gamma \circ (\text{Id} + \theta)) \left( \frac{\partial f}{\partial n_{\partial\Omega}} \circ (\text{Id} + \theta) \right) |\text{com}(I + \nabla \theta) n_{\partial\Omega}| \overline{p} ds \right]. \end{aligned}$$

**Step 4.** Derivation of the surfacic form. At this point one could integrate by parts towards an expression of the form:

$$\int_{\partial\Omega} v_{\Omega} \theta \cdot n_{\partial\Omega} ds, \quad \text{for some } v_{\Omega},$$

but since we already know this to be the case for the integrals over  $\Omega$  and  $\partial\Omega$ , which cancel when we set  $\theta \cdot n_{\partial\Omega} = 0$ , we look at the terms that depend on  $\Gamma_R$ . From one side we have:

$$\int_{\Gamma_R} (\nabla_{\partial\Omega} \cdot \theta) u_{\Omega} \overline{p} ds = \int_{\partial\Gamma_R} u_{\Omega} \overline{p} \theta \cdot n_{\partial\Gamma_R} d\sigma + \int_{\Gamma_R} -\nabla_{\partial\Omega}(u\overline{p}) + \kappa u \overline{p} \theta \cdot n_{\partial\Omega} ds.$$

For the second integral depending on  $\Gamma_R$ , we find:

$$\int_{\Gamma_R} \nabla_{\partial\Omega} \cdot (f\theta) \overline{p} ds = \int_{\partial\Gamma_R} f \overline{p} \theta \cdot n_{\partial\Gamma_R} d\sigma + \int_{\Gamma_R} (-\nabla_{\partial\Omega} \overline{p} \cdot \theta + \kappa \overline{p} \theta \cdot n_{\partial\Omega}) ds$$

In this manner, after setting  $\theta \cdot n_{\partial\Omega} = 0$ , we can eventually deduce:

$$\int_{\Omega} j'(u_{\Omega}) \overline{\dot{u}_{\Omega}(\theta)} dx = \frac{i\omega}{z} \int_{\partial\Gamma_R} \overline{(u_{\Omega} + f)p} \theta \cdot n_{\partial\Gamma_R} d\sigma.$$

As a consequence,

$$\text{Re} \left( \int_{\Omega} j'(u_{\Omega}) \overline{\dot{u}_{\Omega}(\theta)} dx \right) = -\frac{\omega}{z} \text{Im} \left( \int_{\partial\Gamma_R} \overline{(u_{\Omega} + f)p} \theta \cdot n_{\partial\Gamma_R} d\sigma \right),$$

whence the result. ■

# Appendix B

## Differentiability

---

### B.1 The Fréchet derivative

The Fréchet derivative is the generalization of the classical derivative to Banach spaces. For a function  $f : V \rightarrow W$ , where  $V$  and  $W$  are Banach spaces, the Fréchet derivative at a point  $u \in V$  is a bounded linear operator  $L : V \rightarrow W$  such that  $L$  provides the best linear approximation to  $f$  near  $x_0$ .

**Definition B.1 (Fréchet differentiability).** *Let  $V$  and  $W$  be two Banach spaces. Let  $f$  be defined on a neighborhood of  $u \in V$  with values in  $W$ . We say  $f$  is **Fréchet differentiable** at  $u$  if there exists a continuous linear mapping  $L : V \rightarrow W$  such that*

$$f(u + w) = f(u) + L(w) + o(w), \quad \text{with } \lim_{w \rightarrow 0} \frac{\|o(w)\|_W}{\|w\|_V} = 0 \quad (\text{B.1})$$

We call  $f'(u) := L$ , the **Fréchet derivative** of  $f$  at  $u$ .

The Fréchet derivative satisfies the usual rules of calculus, including:

1. **Linearity:** If  $f$  and  $g$  are Fréchet differentiable and  $\alpha$  and  $\beta$  are scalars, then:

$$D(\alpha f + \beta g)(x) = \alpha Df(x) + \beta Dg(x).$$

2. **Product rule:** If  $f, g : X \rightarrow Y$  are Fréchet differentiable, then their product (in the case of Banach algebras) satisfies:

$$D(fg)(x) = Df(x)g(x) + f(x)Dg(x).$$

3. **Chain rule:** If  $f : U \rightarrow V$  and  $g : V \rightarrow W$  are Fréchet differentiable, then the composition  $g \circ f$  is Fréchet differentiable, and:

$$D(g \circ f)(x) = Dg(f(x)) \circ Df(x).$$

For more information on Fréchet differentiability, we refer to the standard books [277, 244]

### B.2 Differentials and gradients

Let  $H$  be a Hilbert space with inner product  $a(\cdot, \cdot)$  and norm  $\|u\|_H := a(u, u)^{1/2}$ . The Fréchet derivative of a function  $F : H \rightarrow \mathbb{R}$  at  $u \in H$  is the bounded linear map:

$$F'(u) \in H^*, \quad v \mapsto F'(u)(v),$$

satisfying the first-order expansion:

$$F(u+v) = F(u) + F'(u)(v) + o(v), \quad \frac{|o(v)|}{\|v\|_H} \rightarrow 0 \text{ as } v \rightarrow 0.$$

By the Riesz representation theorem,  $H^*$  is identified with  $H$ , allowing us to define the gradient  $g \in H$  as the unique element such that:

$$\forall v \in H, \quad a(g, v) = F'(u)(v).$$

If  $\tilde{a}(\cdot, \cdot)$  is another equivalent inner product on  $H$ , the corresponding gradient  $\tilde{g} \in H$  satisfies:

$$\forall v \in H, \quad \tilde{a}(\tilde{g}, v) = F'(u)(v),$$

where  $\tilde{g}$  generally differs from  $g$ , and gradients may be defined with respect to  $\tilde{a}$  instead of  $a$ .

Now, let  $(V, \|\cdot\|_V)$  be a Banach space, and  $F : V \rightarrow \mathbb{R}$  a differentiable function. The Fréchet derivative  $F'(u)$  at  $u \in V$  is still defined via the first-order expansion, but without a gradient, as Banach spaces generally lack an inner product. However, if  $H \subset V$  is a Hilbert space continuously embedded in  $V$ , meaning:

$$\forall v \in H, \quad \|v\|_V \leq C\|v\|_H,$$

the derivative  $F'(u)(v)$  induces a bounded map on  $H$ , and by the Riesz theorem, there exists  $g \in H$  such that:

$$\forall v \in H, \quad a(g, v) = F'(u)(v).$$

Conversely, if  $F : V \rightarrow \mathbb{R}$  is differentiable and  $V$  is embedded in a larger Hilbert space  $H$ , the derivative  $F'(u)$  may extend to  $H$ , providing a broader regularity for  $F$ .

### B.3 Jacobi's formula

Jacobi's formula is a mathematical result in matrix theory that provides the derivative of the determinant of a matrix. Specifically, it expresses how the determinant of a matrix changes when the matrix itself changes with respect to some parameter, usually time. The formula is useful in differential equations, control theory, and other areas of applied mathematics.

**Theorem B.1 (Jacobi's formula).** *Let  $A$  be a differentiable map from the real numbers to  $\mathbb{R}^{n \times n}$ , then:*

$$\frac{d}{dt} \det A(t) = \text{tr} \left( \text{adj}(A(t)) \frac{dA(t)}{dt} \right) = (\det A(t)) \text{tr} \left( A(t)^{-1} \frac{dA(t)}{dt} \right) \quad (\text{B.2})$$

### B.4 Differentiation through a minimum

This section is excerpted from [116] and the reader is referred to it for further reference. Consider a functional

$$G : [0, \tau] \times X \rightarrow \mathbb{R}$$

for some  $\tau > 0$  and some set  $X$ . For each  $t$  in  $[0, \tau]$ , define

$$g(t) \stackrel{\text{def}}{=} \inf \{ G(t, x) : x \in X \},$$

$$X(t) \stackrel{\text{def}}{=} \{ x \in X : G(t, x) = g(t) \}.$$

The objective is to characterize the limit

$$dg(0) \stackrel{\text{def}}{=} \lim_{t \rightarrow 0} \frac{g(t) - g(0)}{t}$$

when  $X(t)$  is not empty for  $0 \leq t \leq \tau$ . When  $X(t) = \{x^t\}$  is a singleton,  $0 \leq t \leq \tau$ , and the derivative

$$\dot{x} = \lim_{t \rightarrow 0} \frac{x^t - x^0}{t}$$

of  $x$  is known, then it is easy to obtain  $dg(0)$  under appropriate differentiability of the functional  $G$  with respect to  $t$  and  $x$ . When  $\dot{x}$  is not readily available or when the sets  $X(t)$  are not singletons, this direct approach fails or becomes very intricate.

We present a theorem that gives an explicit expression for  $dg(0)$ , the derivative of the minimum of the functional  $G$  with respect to  $t$  at  $t = 0$ . Its originality is that the differentiability of  $x^t$  is replaced by a continuity assumption on the set-valued function and the existence of the partial derivative of the functional  $G$  with respect to the parameter  $t$ . In other words, this technique does not require a priori knowledge of the derivative  $\dot{x}$  of the minimizing elements  $x^t$  with respect to  $t$ .

**Theorem B.2 (Differentiation through a minimum).** *Let  $X$  be an arbitrary set,  $\tau > 0$  and  $G : [0, \tau] \times X \rightarrow \mathbb{R}$  a functional. Denote, for every  $t \in [0, \tau]$ ,  $g(t) := \inf_X G(t, \cdot)$ . Assume that the four conditions below are fulfilled:*

1. *For every  $t \in [0, \tau]$ , the set  $X(t) := \{x \in X \mid G(t, x) = \inf_X G(t, \cdot)\}$  is nonempty.*
2.  *$G$  is differentiable with respect to  $t$  at every  $(t, x) \in [0, \tau] \times X$ .*
3. *For every  $x \in X(0)$ , the map  $t \mapsto \frac{\partial G}{\partial t}(t, x)$  is upper semicontinuous at  $t = 0$ .*

## B.5 Implicit function theorem

The crucial element in proving the differentiability of a function over the domain is the implicit function theorem.

**Theorem B.3 (Implicit function theorem).** *Let  $\Theta, E, F$  be Banach spaces,  $\mathcal{V} \subset \Theta, U \subset E$  be open sets. Let  $p \geq 1$ , and  $\mathcal{F} : \mathcal{V} \times U \rightarrow G$  be a function of class  $C^p$ . For a given point  $(\theta_0, u_0) \in \mathcal{V} \times U$  where  $\mathcal{F}(\theta_0, u_0) = 0$ , assume that the partial differential  $D_u \mathcal{F}(\theta_0, u_0) : F \rightarrow G$  is a linear isomorphism. Then there exists an open subset  $\mathcal{V}_0 \subset \mathcal{V}$  of  $\theta_0$  in  $\Theta$ , and a mapping  $g : \mathcal{V}_0 \rightarrow U$  such that:*

- i)  $g(\theta_0) = u_0$ ,
- ii) for  $\theta \in \mathcal{V}_0$ , the equation  $\mathcal{F}(\theta, u) = 0$  has  $u = g(\theta)$  for unique solution in  $E$ ,
- iii) The mapping  $g$  is of class  $C^p$ .

## B.6 The space $W^{1,\infty}$

**Definition B.2 (Lipschitz diffeomorphism).** *The set  $\mathcal{L}$  defined by*

$$\mathcal{L} := \{T : \mathbb{R}^d \rightarrow \mathbb{R}^d \mid (T - \text{Id}) \in W^{1,\infty}(\mathbb{R}^d; \mathbb{R}^d) \text{ and } (T^{-1} - \text{Id}) \in W^{1,\infty}(\mathbb{R}^d; \mathbb{R}^d)\} \quad (\text{B.3})$$

*is called the space of Lipschitz diffeomorphisms.*

# Appendix C

## Useful integration formulas

---

### C.1 Green's identities

The following are simple consequences of the usual Green's identities.

**Proposition C.1.** *Let  $\Omega$  be a domain with smooth boundary. They are valid for deformations  $\theta \in W^{1,\infty}(\mathbb{R}^d, \mathbb{R}^d)$ , functions  $w \in H^1(\Omega)$ , and vector fields  $a, b \in H^1(\Omega)^d$ .*

$$\int_{\Omega} w \nabla \cdot \theta \, dx = \int_{\partial\Omega} w \theta \cdot n \, ds - \int_{\Omega} \theta \cdot \nabla w \, dx, \quad (\text{C.1})$$

and

$$\int_{\Omega} \nabla \theta a \cdot b \, dx = \int_{\partial\Omega} (a \cdot n)(\theta \cdot b) \, ds - \int_{\Omega} (\operatorname{div} a)(b \cdot \theta) \, dx - \int_{\Omega} (\nabla b) a \cdot \theta \, dx. \quad (\text{C.2})$$

**Corollary C.1.** *Let  $\Omega \subset \mathbb{R}^d$  be a bounded, Lipschitz domain, and let  $\phi$  be a diffeomorphism from  $\mathbb{R}^d$  onto itself. Then, a measurable function  $f : \mathbb{R}^d \rightarrow \mathbb{R}$ , belongs to  $L^1(\phi(\partial\Omega))$  if and only if  $f \circ \phi \in L^1(\partial\Omega)$ , and:*

$$\int_{\phi(\partial\Omega)} f \, dx = \int_{\Omega} |\det(\nabla\phi)| f \circ \phi \, dx, \quad (\text{C.3})$$

where  $\nabla\phi$  is the Jacobian matrix of  $\phi$ .

**Corollary C.2.** *Let  $\Omega \subset \mathbb{R}^d$  be a bounded, Lipschitz domain, and let  $\phi$  be a diffeomorphism from  $\mathbb{R}^d$  onto itself. Then, a measurable function  $f : \mathbb{R}^d \rightarrow \mathbb{R}$ , belongs to  $L^1(\phi(\partial\Omega))$  if and only if  $f \circ \phi \in L^1(\partial\Omega)$ , and:*

$$\int_{\phi(\partial\Omega)} f \, dx = \int_{\partial\Omega} \det(\nabla\phi) |(\nabla\phi)^{-T} n| f \circ \phi \, dx, \quad (\text{C.4})$$

where  $\nabla\phi$  is the Jacobian matrix of  $\phi$ .

### C.2 Tangential calculus

**Proposition C.2.** *Let  $\Omega \subset \mathbb{R}^d$  be a smooth bounded domain with boundary  $\partial\Omega$ . Let  $u \in H^1(\partial\Omega)^d$  and  $\theta \in H^1(\partial\Omega)^d$ . Then:*

$$\int_{\partial\Omega} u \nabla_{\partial\Omega} \cdot \theta \, ds = \int_{\partial\Omega} (-\theta \cdot \nabla_{\partial\Omega} u + \kappa u \theta \cdot n) \, ds \quad (\text{C.5})$$

# Appendix D

## Riemannian geometry

---

In this section, we provide the necessary geometric preliminaries for understanding the contents of [Chapter 2](#). For a comprehensive understanding of differential geometry, we recommend referring primarily to the manuscripts [\[219, 220\]](#). Additionally, we suggest the following references specifically focused on Riemannian geometry [\[122, 80\]](#) and differential geometry in general [\[121, 303\]](#).

### D.0.1 Riemannian manifolds

We begin our review with the definition of a tangent space.

**Definition D.1 (Tangent space).** For every point  $p \in M$ , a **tangent vector at  $p$**  is a linear map  $v : C^\infty(M) \rightarrow \mathbb{R}$  that is a **derivation at  $p$** , meaning that for all  $f, g \in C^\infty(M)$  it satisfies the product rule:

$$v(fg) = f(p)vg + g(p)vf . \tag{D.1}$$

The set of all tangent vectors at  $p$  is denoted by  $T_pM$  and called the **tangent space at  $p$** .

We shall not explain how or why it makes sense to utilize this notion of tangent vectors as derivations, since [\[220\]](#) already provides an excellent introduction and discussion on this topic. However, it proves useful to mention that any member of  $T_pM$  can be regarded as a pair  $(p, v)$  where  $p$  represents a point on the manifold and  $v$  a vector direction.

**Definition D.2 (Riemannian manifold).** A **Riemannian manifold** is a pair  $(M, g)$  where  $M$  is a smooth manifold and  $g$  is a 2-tensor field whose value  $g_p$  at each point  $p \in M$  is an inner product on the tangent space  $T_pM$ .

We will consistently assume that all manifolds under consideration are Riemannian, and any submanifold will be equipped with its induced metric. At each point  $p \in M$ , the inner product  $g_p$  on the tangent space  $T_pM$  will be denoted as  $u \cdot v := g_p(u, v)$  for any  $u, v \in T_pM$ , assuming the choice of metric is clear. We define the *length* or *norm* of a vector  $v \in T_pM$  as  $|v| := \sqrt{v \cdot v}$ .

Regarding notation, the symbol  $M$  typically represents the ambient manifold (without boundary) in which all embedded submanifolds reside. A smooth *coordinate chart* on an open subset  $U \subset M$  will be denoted by  $(U, \phi)$ , where  $\phi : U \rightarrow V \subset \mathbb{R}^n$  is a smooth map. We use upper indices for coordinates, so if  $p \in M$ , its coordinates are written as  $\phi(p) = (x^1, \dots, x^n)$ . Curves will typically be denoted as  $\gamma : I \rightarrow M$  or  $\sigma : I \rightarrow M$ , where  $I$  is a subset of  $\mathbb{R}$ .

Furthermore, we use  $C^\infty(M)$  to denote the vector space of all smooth functions from  $M$  to  $\mathbb{R}$ , and  $C^\infty(M, N)$  to denote the set of all smooth maps from a smooth manifold  $M$  to another smooth manifold  $N$ . Additionally, it is important to note that if  $M$  is an  $n$ -dimensional smooth manifold, we have a natural way to represent any tangent vector  $v \in T_pM$  using the so-called *coordinate vectors*.

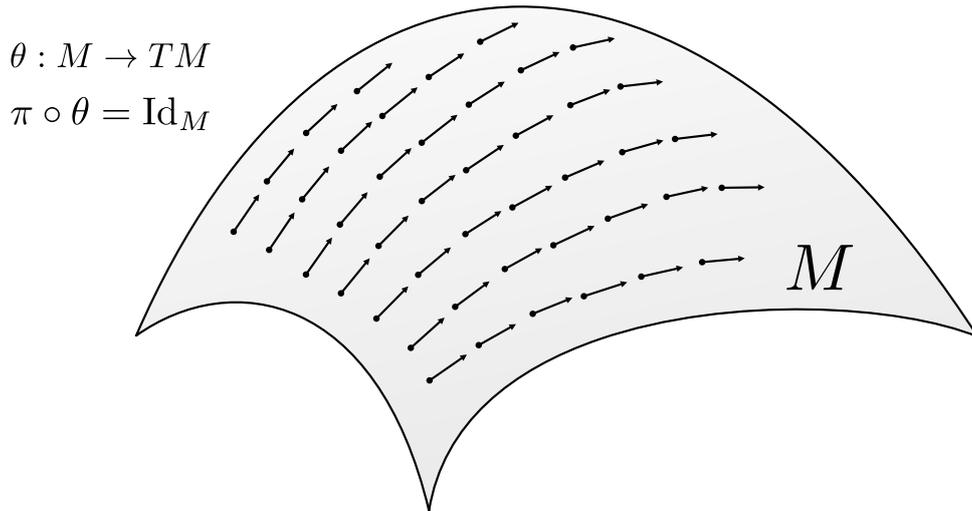


Figure D.1: Example of a vector field  $\theta$  on  $M$ . The value of a vector field  $\theta(p)$  can be regarded as a pair  $(p, v)$ , which satisfies the relation  $\pi(\theta(p)) = p$ .

**Definition D.3.** Let  $M$  be an  $n$ -dimensional smooth manifold, and let  $(U, \phi)$  be coordinate chart, whose coordinate functions are given by  $(x^1, \dots, x^n)$ . We define the **coordinate vectors**  $\partial/\partial x^i|_p$  by:

$$\frac{\partial}{\partial x^i} \Big|_p f = \frac{\partial}{\partial x^i} \Big|_{\phi(p)} (f \circ \phi^{-1}) . \quad (\text{D.2})$$

In particular, these vectors form a basis of  $T_p M$ .

Now we move on to the tangent bundle. The tangent bundle is the disjoint union of the tangent spaces of the  $n$  dimensional manifold  $M$ . In general, the tangent bundle  $TM$  is a particular example of a *vector bundle* and can be made into a  $2n$  dimensional manifold via the natural topology and smooth structure (see for instance Proposition 3.18 in [220]). Furthermore, the notion of tangent bundle, allows us to define the concept of a vector field as a section of the former.

**Definition D.4 (Tangent bundle).** The **tangent bundle  $TM$  of  $M$** , denoted  $TM$ , is the disjoint union of the tangent spaces at all points of  $M$ :

$$TM := \coprod_{p \in M} T_p M . \quad (\text{D.3})$$

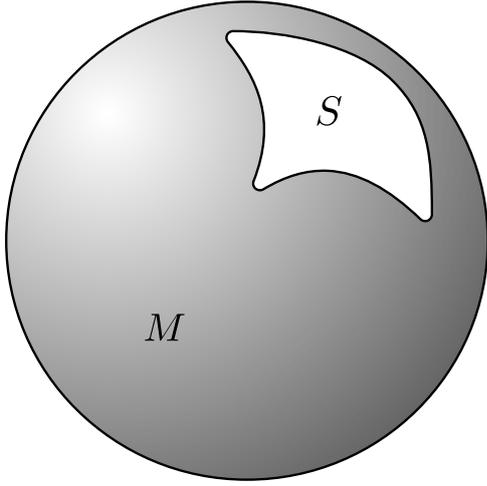
**Definition D.5 (Vector field).** A **smooth section  $\theta$  of  $TM$**  is called a **smooth vector field**. That is, a smooth map  $\theta : M \rightarrow TM$  such that  $\pi \circ \theta = \text{Id}_M$ , where  $\pi : TM \rightarrow M$  is the projection mapping of  $TM$ , i.e.  $\forall p \in M, \theta(p) \in T_p M$ . The set of all smooth vector fields on  $M$  is denoted by  $\mathfrak{X}(M)$ . In addition, the **zero section  $\mathbf{0}_M$**  is defined as the smooth vector field such that  $\mathbf{0}_M(p) = 0 \in T_p M$  for all  $p \in M$ .

**Definition D.6.** Given a smooth curve  $\gamma : I \rightarrow M$ , we define a **smooth vector field along  $\gamma$**  as a smooth map  $\theta : I \rightarrow TM$  such that  $\theta(t) \in T_{\gamma(t)} M$ . We denote the set of all smooth vector fields along  $\gamma$  by  $\mathfrak{X}(\gamma)$ .

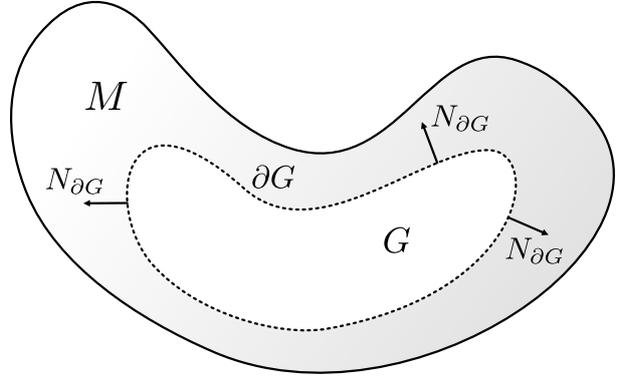
In general, one can think of sections as “graphs” of vectors on a manifold. In other words, the relation  $\pi \circ \theta = \text{Id}_M$  expresses that the arrow  $\theta(p)$  shall have  $p$  as its base point.

**Definition D.7 (Local frame).** A **local frame** for  $TM$  is an ordered  $n$ -tuple  $(E_1, \dots, E_n)$  of local sections over an open set  $U$  such that  $(E_1(p), \dots, E_n(p))$  forms a basis of  $T_p M$  for every  $p \in M$ .

Among local frames, we have in particular the notion of *adapted orthonormal frame*, which is particularly useful in practice.



(a) Example of hypersurface  $S$  embedded in the ambient manifold  $M$ .



(b) Example of a regular domain  $G$ , with boundary  $\partial G$ , embedded in the ambient manifold  $M$ .

**Definition D.8 (Adapted orthonormal frame).** If  $(M, g)$  is an  $n$ -dimensional Riemannian manifold and  $S$  is an  $m$ -dimensional submanifold with its induced metric, an orthonormal local frame  $(E_1, \dots, E_n)$  for  $M$  on an open subset  $U \subset M$  is said to be **adapted to  $S$**  if the first  $m$  vector fields  $E_1, \dots, E_m$  are tangent to  $S$ .

**Lemma D.1.** Let  $(M, g)$  be an  $n$ -dimensional Riemannian manifold, and let  $S$  be a smooth embedded submanifold with induced metric. For any  $p \in S$ , there exists a neighborhood  $U$  of  $p$  in  $M$  and a smooth orthonormal frame for  $M$  on  $U$  that is adapted to  $S$ .

We will be often dealing with hypersurfaces and regular domains embedded in an ambient space.

**Definition D.9 (Hypersurface).** Let  $(M, g)$  be a smooth manifold with metric  $g$ . A **hypersurface  $S$  in  $M$**  is a submanifold of codimension 1 and we refer to  $M$  as the ambient manifold.

**Definition D.10 (Regular domain).** A **regular domain  $G$  of  $M$**  is a subset of  $M$  which is also a closed embedded codimension 0 submanifold with boundary.

Given a regular domain  $G$  embedded in the ambient manifold  $M$ , we will often need to talk about the concept of an outward-pointing unit normal vector field  $n_{\partial G}$ . Let us first define the space which will correspond to the codomain of the normal vector field.

**Definition D.11 (Normal Bundle).** Let  $(M, g)$  be a smooth Riemannian manifold and  $S \subset M$  a smooth submanifold with or without boundary in  $M$ . A vector  $n \in T_p M$  is **normal to  $S$**  if  $w \cdot n = 0$  for every  $w \in T_p S$ . The space of all vectors normal to  $S$  at  $p$ , called the **normal space at  $p$** , is a subspace of  $T_p M$  and is denoted  $N_p S = (T_p S)^\perp$ . The set

$$NS = \coprod_{p \in S} N_p S \quad (\text{D.4})$$

is called the **normal bundle of  $S$** .

With this definition in place, we can define the unit normal vector field.

**Theorem D.1 (Outward-pointing unit normal).** If  $(M, g)$  is a smooth Riemannian manifold with boundary, the normal bundle to  $\partial M$  is a smooth rank-1 vector bundle over  $\partial M$ , and there is a unique smooth outward-pointing unit normal vector field  $N_{\partial M}$  along all of  $\partial M$ .

---



---

**Remark D.1.** *In general, it is possible to extend the previous result to the case of a hypersurface  $S$ . Indeed, at every  $p \in M$  we can consider any adapted orthonormal frame  $(E_1, \dots, E_{n+1})$  on a neighborhood  $U$  of  $p$  in  $M$ , which furnishes us with two choices of unit normal vectors;  $\pm E_{n+1}$ . Utilizing this fact we can deduce that for a sufficiently small neighborhood of  $p$ , we can always choose some smooth unit normal vector field along  $S$ . If both  $M$  and  $S$  are orientable we can use the orientation to choose a smooth unit normal vector field along all of  $S$ , but in general this might or might not be possible.*

## D.0.2 Connections and covariant derivatives

A *connection* in differential geometry is a tool used to compare and relate vectors in the tangent spaces of a manifold at different points. It allows for the definition of how vectors change as they move along a curve on the manifold. Connections are essential for defining concepts like parallel transport, covariant derivatives, and curvature. Essentially, a connection provides a set of formal rules for differentiating vector fields on manifolds.

**Definition D.12 (Connection).** *Let  $TM$  be the tangent vector bundle of  $M$ . A **connection on  $M$**  is a map:*

$$\nabla : \mathfrak{X}(M) \times \mathfrak{X}(M) \rightarrow \mathfrak{X}(M) \quad (\text{D.5})$$

$$(\theta, \eta) \mapsto \nabla_{\theta}\eta, \quad (\text{D.6})$$

which satisfies:

1.  $\nabla_{\theta}\eta$  is linear over  $C^{\infty}(M)$  in  $\theta$ , i.e.

$$\nabla_{f_1\theta_1+f_2\theta_2}\eta = f_1\nabla_{\theta_1}\eta + f_2\nabla_{\theta_2}\eta, \quad (\text{D.7})$$

for any  $f_1, f_2 \in C^{\infty}(M)$  and  $\theta, \eta \in \mathfrak{X}(M)$ .

2.  $\nabla_{\theta}\eta$  is linear over  $\mathbb{R}$  in  $\eta$ , i.e.

$$\nabla_{\theta}(a_1\eta_1 + a_2\eta_2) = a_1\nabla_{\theta}\eta_1 + a_2\nabla_{\theta}\eta_2, \quad (\text{D.8})$$

for  $a_1, a_2 \in \mathbb{R}$  and  $\eta_1, \eta_2 \in \mathfrak{X}(M)$ .

3.  $\nabla$  satisfies the product rule, i.e.

$$\nabla_{\theta}(f\eta) = f\nabla_{\theta}\eta + (\theta f)\eta. \quad (\text{D.9})$$

Furthermore, we call  $\nabla_{\theta}\eta$  the **covariant derivative of  $\eta$  in the direction  $\theta$** .

In flat, Euclidean space, the derivative of a vector field is straightforward, as we can simply subtract vectors at different points. However, in curved spaces or manifolds, vectors at different points lie in different tangent spaces, so we need a way to compare them that considers the manifold's curvature. This is where the covariant derivative comes in.

**Theorem D.2 (Covariant derivatives along curves).** Let  $M$  be a smooth manifold and let  $\nabla$  be a connection on  $M$ . For each smooth curve,  $\gamma : I \rightarrow M$ , the connection determines a unique operator:

$$D_t : \mathfrak{X}(\gamma) \rightarrow \mathfrak{X}(\gamma) , \quad (\text{D.10})$$

called the **covariant derivative along  $\gamma(t)$** , satisfying the following properties:

1. *Linearity over  $\mathbb{R}$ ; if  $\theta, \eta \in \mathfrak{X}(\gamma)$  and  $a, b \in \mathbb{R}$ , then:*

$$D_t(a\theta + b\eta) = aD_t\theta + bD_t\eta . \quad (\text{D.11})$$

2. *The product rule; if  $\theta \in \mathfrak{X}(\theta)$ , then:*

$$D_t(f\theta) = f'\theta + fD_t\theta , \quad (\text{D.12})$$

for  $f \in C^\infty(I)$ .

3. *If  $\theta \in \mathfrak{X}(\gamma)$  can be extended to an open neighborhood  $U \subset M$ , then for every extension  $\tilde{\theta}$  of  $\theta$ :*

$$D_t\theta(t) = \nabla_{\gamma'(t)}\tilde{\theta} . \quad (\text{D.13})$$

**Definition D.13 (Lie bracket).** Let  $\theta, \eta \in \mathfrak{X}(M)$ . The map  $[\theta, \eta] : C^\infty(M) \rightarrow C^\infty(M)$  defined by:

$$[\theta, \eta]f := \theta(\eta f) - \eta(\theta f) \quad (\text{D.14})$$

is called the **Lie bracket of  $\theta$  and  $\eta$** .

**Theorem D.3 (Fundamental theorem of Riemannian geometry).** Let  $(M, g)$  be a Riemannian manifold. There exists a unique connection  $\nabla$  on  $M$  such that:

1. *It is compatible with  $g$ . That is, for all  $\theta, \eta, \zeta \in \mathfrak{X}(M)$ :*

$$\nabla_\theta(\eta \cdot \zeta) = (\nabla_\theta\eta) \cdot \zeta + \eta \cdot (\nabla_\theta\zeta) \quad (\text{D.15})$$

2. *It is symmetric. That is, for all  $\theta, \eta \in \mathfrak{X}(M)$ :*

$$\forall f \in C^\infty(M), (\nabla_\theta\eta - \nabla_\eta\theta) f = [\theta, \eta]f \quad (\text{D.16})$$

Such a connection is called the **Levi-Civita connection of  $g$** .

Let us mention that the previous theorem guarantees that we can always work with the Levi-Civita connection, as long as we have a metric  $g$  on  $M$ . From now on, we shall always make the assumption that we utilize the Levi-Civita connection without risk of confusion. Let us end this section with the definition of the so called *Christoffel symbols*.

**Proposition D.1 (Christoffel symbols).** Let  $E = (E_1, \dots, E_n)$  be a smooth local frame for  $TM$  on an open subset  $U \subset M$ . Then there exist  $n^3$  smooth function  $\Gamma_{i,j}^k : U \rightarrow \mathbb{R}$  called the **Christoffel symbols with respect to  $E$** , such that:

$$\nabla_{E_i}E_j = \sum_{k=0}^n \Gamma_{i,j}^k E_k . \quad (\text{D.17})$$

The Christoffel symbols describe how a vector field changes due to the curvature. In simpler terms, they correct the naive derivative to account for the curvature, ensuring that the derivative is intrinsic to the manifold.

### D.0.3 Geodesics, the exponential map, and tubular neighborhoods

Geodesics are the generalization of the concept of a “straight line” to curved spaces. In a mathematical context, particularly in differential geometry, a geodesic is defined as a curve that represents the shortest

distance between two points on a surface. Formally, it is a curve  $\gamma(t)$  in a Riemannian manifold such that the tangent vector to the curve remains parallel to itself when transported along the curve.

**Definition D.14 (Geodesic).** A smooth curve  $\gamma : I \rightarrow M$  is called a **geodesic** if for every  $t \in I$ ,  $D_t \gamma'(t) = 0$ , where  $D_t$  denotes the covariant derivative along the curve.

The next theorem establishes the existence and uniqueness of geodesics in a manifold.

**Theorem D.4 (Existence and uniqueness of geodesics).** For every  $p \in M$ ,  $v \in T_p M$ , and  $a \in \mathbb{R}$ , there exists an open interval  $I \subset \mathbb{R}$  containing  $t$  and a geodesic  $\gamma : I \rightarrow M$  satisfying  $\gamma(a) = p$  and  $\gamma'(a) = v$ .

Whenever we talk about a geodesic, we can utilize the fact that at each  $p$ , the vector  $v \in T_p M$  can be regarded as a pair  $v = (p, v_p)$ , which encodes the base point information and the direction in space, in order to simply denote  $\gamma_v$  for a geodesic which satisfies  $\gamma(a) = p$ ,  $\gamma'(a) = v$ . This notation is useful in the definition of the *exponential map*.

**Definition D.15.** Let  $(M, g)$  a Riemannian manifold. Define the subset  $\mathcal{D} \subset TM$  by

$$\mathcal{D} := \{v \in TM : \gamma_v \text{ is defined on } I\}, \quad (\text{D.18})$$

where  $[0, 1] \subset I$ . The **exponential map** is defined by:

$$\exp : \mathcal{D} \rightarrow M \quad (\text{D.19})$$

$$v \mapsto \gamma_v(1) . \quad (\text{D.20})$$

For each  $p \in M$ , the restriction of  $\mathcal{D}$  to  $T_p M$  is denoted  $\exp_p$  and called the **exponential map at  $p$** .

The exponential map connects the tangent space at a point to the manifold itself. It provides a way to translate a vector in the tangent space into a point on the manifold, following the direction and length specified by the vector.

**Remark D.2.** Note that in general the geodesics on a submanifold  $(S, \hat{g})$ , with its induced metric, of a Riemannian manifold  $(M, g)$  will not be equal in general. This is an important distinction to make and henceforth we will denote the submanifold exponential map by  $\widehat{\exp}$ , whenever talking about geodesics in  $S$ .

The exponential map enables the definition of a normal neighborhood and normal coordinates. In the context of a Riemannian manifold, a normal neighborhood is a special type of region around a point with certain advantageous properties. Specifically, it is an area where the exponential map is well-behaved, ensuring a one-to-one correspondence between points in the tangent space and points on the manifold.

**Definition D.16 (Star-shaped set).** A subset  $U$  of a vector space  $V$  is said to be **star-shaped** with respect to a point  $x \in V$  if for every  $y \in V$ , the line segment from  $x$  to  $y$  is contained in  $U$ .

**Definition D.17 (Normal neighborhood).** A neighborhood  $U$  of  $p \in M$  that is the diffeomorphic image under  $\exp_p$  of a star-shaped neighborhood  $V$  of  $0 \in T_p M$  is called a **normal neighborhood of  $p$** .

Normal coordinates, also known as geodesic normal coordinates or Riemannian normal coordinates, are a special type of coordinate system in the vicinity of a point on a Riemannian manifold. These coordinates simplify many calculations and provide a locally "flat" view of the manifold, making them particularly useful for studying the local geometry around a point.

**Definition D.18 (Normal coordinates).** Let  $p \in M$  and let  $(b_1, \dots, b_n)$  be an orthonormal basis for  $T_p M$ . Let  $B : \mathbb{R}^n \rightarrow T_p M$  be the basis isomorphism defined by  $B(x^1, \dots, x^n) = \sum_{i=1}^n x^i b_i$ . If  $U = \exp_p(V)$  is a normal neighborhood of  $p$ , then the smooth coordinates  $\phi : U \rightarrow \mathbb{R}^n$  defined by:

$$\phi = B^{-1} \circ (\exp_p|_V)^{-1} , \quad (\text{D.21})$$

are called **normal coordinates centered at  $p$** .

**Proposition D.2.** *Let  $p \in M$  and  $U$  a normal neighborhood of  $p$ . For every normal coordinate chart on  $U$  centered at  $p$ , the coordinate basis is orthonormal at  $p$ ; and for every orthonormal basis  $(b_1, \dots, b_n)$  for  $T_p M$ , there is a unique normal coordinate chart  $(U, x)$  on  $U$  such that  $\frac{\partial}{\partial x^i} \Big|_p = b_i$  for  $i = 1, \dots, n$ .*

**Proposition D.3 (Properties of normal coordinates).** *Let  $(U, x)$  be any normal coordinate chart centered at  $p \in M$ . Then:*

1.  $x(p) = 0$ .
2. For every  $v = \sum_{i=1}^n v^i \frac{\partial}{\partial x^i} \Big|_p$  the geodesic  $\gamma_v : I \rightarrow M$  starting at  $p$  with initial velocity  $v$  is represented in normal coordinates by the line:

$$\gamma_v(t) = (tv^1, \dots, tv^n) \tag{D.22}$$

for  $t \in I$  and as long as  $0 \in I$ .

3. The Christoffel symbols in these coordinates vanish at  $p$ .

Closely related and of great importance is the concept of a tubular neighborhood, which is heavily utilized in [Chapter 2](#). For a submanifold  $S$ , a tubular neighborhood provides a way to “thicken”  $S$  within the ambient manifold while maintaining a structured relationship between the points in the neighborhood and the submanifold. This concept is fundamental in differential geometry for studying the local geometry around submanifolds and has various applications in mathematical analysis and theoretical physics.

**Definition D.19 (Tubular neighborhood).** *Let  $S \subset M$  be an embedded submanifold,  $\pi : NS \rightarrow S$  the normal bundle of  $S$  in  $M$ , and define  $\mathcal{D}_S := \mathcal{D} \cap NS$ . A **tubular neighborhood** of  $S$  in  $M$  is an open subset  $U \subset M$  which is the diffeomorphic image under  $\exp|_{\mathcal{D}_S}$  of a subset  $V \subset \mathcal{D}_S$  of the form:*

$$V = \{(x, v) \in NS : |v| < \delta(x)\} , \tag{D.23}$$

for some positive continuous function  $\delta : S \rightarrow (0, +\infty)$ . If  $\delta(x) = \epsilon > 0$  for all  $x \in M$  then it is called an  **$\epsilon$ -tubular neighborhood of  $S$** .

The following theorem asserts that any sufficiently small neighborhood of a submanifold can be smoothly and uniquely mapped onto an open subset of its normal bundle. This theorem is particularly useful for studying the geometry and topology of submanifolds within a larger manifold.

**Theorem D.5 (Tubular neighborhood theorem).** *Let  $(M, g)$  be a Riemannian manifold. Every embedded submanifold of  $M$  has a tubular neighborhood in  $M$ , and every compact submanifold has an  $\epsilon$ -tubular neighborhood.*

Normal neighborhoods and tubular neighborhoods are related concepts in differential geometry that help in understanding the local structure of manifolds. Normal neighborhoods focus on the area around a single point, using the exponential map to relate the tangent space to the manifold. Tubular neighborhoods focus on the region around a submanifold, using the normal bundle to describe the neighborhood. Both concepts simplify the analysis of manifolds by providing well-behaved local structures.

## D.0.4 Curvature, families of curves, and Jacobi fields

Curvature is a concept in differential geometry that quantifies how a geometric object deviates from being flat or straight. It measures the amount by which a curve, surface, or manifold bends. There are several types of curvature, each applicable to different geometric contexts, such as curves, surfaces, and higher-dimensional spaces. The most general definition is the curvature endomorphism  $R$ . It measures how much the metric tensor deviates from being locally isometric to Euclidean space.

---



---

**Definition D.20 (Curvature endomorphism).** *The map defined by:*

$$R : \mathfrak{X}(M) \times \mathfrak{X}(M) \times \mathfrak{X}(M) \rightarrow \mathfrak{X}(M) \quad (\text{D.24})$$

$$(\theta, \eta, \zeta) \mapsto \nabla_\theta \nabla_\eta \zeta - \nabla_\eta \nabla_\theta \zeta - \nabla_{[\theta, \eta]} \zeta, \quad (\text{D.25})$$

*is called the **Riemann curvature endomorphism**.*

The previous definition is quite general, and the information which it contains is much more conveniently encoded into the following definition.

**Definition D.21 (Curvature tensor).** *The map defined by:*

$$\text{Riem} : \mathfrak{X}(M) \times \mathfrak{X}(M) \times \mathfrak{X}(M) \times \mathfrak{X}(M) \rightarrow \mathbb{R} \quad (\text{D.26})$$

$$(\theta, \eta, \zeta, \beta) \mapsto R(\theta, \eta) \zeta \cdot \beta, \quad (\text{D.27})$$

*is called the **Riemann curvature tensor**.*

The link between the curvature and geodesics can be established via the concept of variation through geodesics, which captures how geodesics in a Riemannian manifold change as some parameters defining them are varied. This concept introduces the variation vector field and Jacobi fields, providing insights into the curvature and geometric properties of the manifold. It is fundamental in the calculus of variations, understanding geodesic flows, and has important applications in fields like general relativity, where it helps describe geodesic deviation and tidal forces.

**Definition D.22.** *Given intervals  $I, J \subset \mathbb{R}$ , a smooth map  $\Gamma : J \times I \rightarrow M$  is called a **one-parameter family of curves**. If  $\gamma : [a, b] \rightarrow M$  is a smooth curve then a **variation of  $\gamma$**  is a family curves such that  $\Gamma(0, t) = \gamma(t)$  for every  $t \in I$ . If  $\Gamma$  is a variation  $\gamma$ , the **variation field of  $\Gamma$**  is the smooth vector field  $t \mapsto \frac{\partial}{\partial s} \Gamma(0, t)$  along  $\gamma$ . Additionally, if  $\gamma$  is a geodesic and for every  $s \in J$  the curve  $t \mapsto \Gamma(s, t)$  is a geodesic, then we say  $\Gamma$  is a **variation through geodesics**.*

In particular, smooth families of curves satisfy the following relation.

**Lemma D.2 (Symmetry lemma).** *Let  $\Gamma : J \times I \rightarrow M$  be a smooth family of curves in a Riemannian manifold. On every rectangle  $R_i := J \times [a_{i-1}, a_i] \subset I$ , we have:*

$$\forall (s, t) \in R_i, \quad D_s \frac{\partial}{\partial t} \Gamma(s, t) = D_t \frac{\partial}{\partial s} \Gamma(s, t). \quad (\text{D.28})$$

We now state the link between the curvature tensor and geodesics.

**Theorem D.6 (Jacobi equation).** *Let  $(M, g)$  be a Riemannian manifold, let  $\gamma$  be a geodesic in  $M$ , and let  $J$  be a vector field along  $\gamma$ . If  $J$  is the variation field of a variation through geodesics, then  $J$  satisfies the following equation called the **Jacobi equation**:*

$$D_t^2 J + R(J, \gamma') \gamma' = 0, \quad (\text{D.29})$$

*where  $R$  is the Riemann curvature endomorphism. A smooth vector field along a geodesic that satisfies the Jacobi equation is called a **Jacobi field**.*

In a nutshell, the Jacobi field is a vector field along a geodesic that describes how nearby geodesics deviate from one another, governed by the Jacobi equation involving the Riemann curvature tensor. Jacobi fields are crucial for understanding geodesic stability, the effect of curvature, and the occurrence of conjugate points, with significant applications in differential geometry and general relativity.

**Remark D.3.** *Let us comment on the physical interpretation of the Jacobi equation. It characterizes the precise behavior of how a rigid body is deformed while traveling along a geodesic path in the manifold. In physical terms, we can think of the curvature tensor as representing tidal forces that stretch or compress the rigid body in a gravitational field. We refer the reader to [76] for more interpretations and general knowledge on the Jacobi equation.*

### D.0.5 Distance functions

This section will briefly present some useful concepts and results of distance functions on a Riemannian manifold  $(M, g)$ . Furthermore, we will present some results concerning its differentiable properties. We consider all of our curves  $\gamma : [a, b] \rightarrow M$  to be smooth and define the *length of  $\gamma$*  to be:

$$L_g(\gamma) = \int_a^b |\gamma'(t)|_g dt . \quad (\text{D.30})$$

This notion of length allows us to define the concept of *Riemannian distance* between two points, which in turn will let us define the notion of a distance function on a manifold.

**Definition D.23 (Riemannian distance).** For each pair of points  $p, q \in M$ , we define the **Riemannian distance from  $p$  to  $q$** , denoted by  $d^M(p, q)$  to be the infimum of the lengths of all admissible curves from  $p$  to  $q$ . Mathematically,  $d^M(p, q) : M \times M \rightarrow \mathbb{R}$  is the function defined by:

$$d^M(p, q) := \inf_{\substack{\gamma : [a, b] \rightarrow M, \\ \gamma(a)=p, \gamma(b)=q}} L_g(\gamma) . \quad (\text{D.31})$$

**Definition D.24.** Suppose  $S \subset M$  is any subset. For each point  $x \in M$ , we define the **distance from  $x$  to  $S$**  to be:

$$d^M(x, S) = \inf_{p \in S} d(x, p) . \quad (\text{D.32})$$

# Appendix E

## Fréchet spaces, the Gateaux derivative, and continuous differentiability

---

This chapter clarifies the definition of Gateaux derivative and continuous differentiability when working on Fréchet spaces. The clarification and presentation of these concepts is necessary since we will be performing calculus on functions which belong to  $C^\infty(M)$ , which notably lack a norm and cannot be made into Banach spaces. The main reference for this section is the landmark work [178] but we also recommend any reference on topological vector spaces such as [212, 284]. We start with the definition of a *seminorm* and define our objects in terms of this elementary definition.

**Definition E.1 (Seminorm).** A *seminorm* on a vector space  $X$  is a real valued function  $\|\cdot\| : X \rightarrow \mathbb{R}$  such that:

1.  $\forall x \in X, \|x\| \geq 0$ ;
2.  $\forall x, y \in X, \|x + y\| \leq \|x\| + \|y\|$ ;
3.  $\forall c \in \mathbb{R}, \forall x \in X, \|cx\| = |c| \cdot \|x\|$

A collection of seminorms  $\{\|\cdot\|_k\}_{k \in \mathbb{N}}$  defines a unique topology such that a sequence  $x_i \rightarrow x$  if and only if  $\|f_i - f\|_k \rightarrow 0$  for all  $k \in \mathbb{N}$ .

A Fréchet space is a type of topological vector space that generalizes many of the properties of Banach spaces but without requiring a norm. Instead, it is defined using a countable family of seminorms.

**Definition E.2 (Fréchet space).** A *Fréchet space* is a vector space  $X$  which is:

1. *Metriizable.* Its topology is defined from a family of countable collection of seminorms  $\{\|\cdot\|_k\}_{k \in \mathbb{N}}$ .
2. *Hausdorff space.* For  $x \in X, x = 0$  when  $\|x\|_k = 0$  for all  $k \in \mathbb{N}$ .

More succinctly, a Fréchet space is a Hausdorff metriizable locally convex topological vector space.

In contrast to Banach spaces, a Fréchet space is defined by a countable family of seminorms rather than a single norm. This family of seminorms induces a topology that facilitates the convergence and continuity needed for the space's completeness. Unlike Banach spaces, Fréchet spaces do not necessarily have a norm, offering greater flexibility to accommodate spaces like the space of smooth functions. The locally convex and metriizable structure of Fréchet spaces makes them particularly well-suited for more complex analytical tasks involving infinite-dimensional spaces. Examples of Fréchet spaces are:

- *Banach spaces.* Every Banach space is a Fréchet space since the collection of seminorms is a singleton with its sole element the norm defined on the Banach space.

- *Countable product of Fréchet spaces.* Let  $\{X_i\}_{i \in \mathbb{N}}$  be a family of Fréchet spaces. The (countable) product  $X := \prod_{i=1}^{\infty} X_i$  is again a Fréchet space.
- $C^\infty(M, B)$ . The set of smooth functions defined on a compact manifold  $M$  which have as co-domain the Banach space  $B$  forms a vector space and can be made into a Fréchet space by defining the collection of seminorms:

$$\forall f \in C^\infty(M, B), \|f\|_k := \sum_{i=0}^k \sup_{p \in M} \left| \frac{\partial f}{\partial x_i}(p) \right|, \quad (\text{E.1})$$

where  $\frac{\partial f}{\partial x_i}$  denotes the  $i$ -th partial derivative of  $f$ . In particular the space  $C^\infty(M)$  forms a Fréchet space.

- $C^\infty(M, TM)$ . More generally, if  $M$  denotes a compact Riemannian manifold equipped with some Riemannian metric, and  $C^\infty(M, TM)$  is the set of smooth maps from  $M$  to its tangent bundle (taken as a manifold), we can define the following collection of seminorms:

$$\forall f \in C^\infty(M, TM), \|f\|_k := \sum_{i=0}^k \sup_{p \in M} |D^i f(p)|, \quad (\text{E.2})$$

where  $D^i f$  denotes the  $i$ -th covariant derivative of  $f$ .

A key observation regarding Fréchet spaces is that they can be viewed as a generalization of Banach spaces while still preserving enough structure for “typical operations” to remain meaningful under broad definitions. For instance, the definition of a Fréchet space as a local convex topological vector space is sufficient for the Hahn-Banach theorem to hold (see, for example, [62, 212, 284, 59, 276]). Moreover, Fréchet spaces provide enough structure to support differential calculus through the concept of the Gateaux derivative. This allows us to perform differential calculus on functions defined on Fréchet spaces.

**Definition E.3 (Gateaux derivative).** Let  $X$  and  $Y$  be Fréchet spaces,  $U \subset X$  an open subset, and  $F : U \rightarrow Y$  be a continuous (possibly non-linear) map. The **Gateaux derivative** of  $F$  at the point  $x \in U$  in the direction  $h \in X$  is defined by:

$$J'(x)(h) := \lim_{t \rightarrow 0} \frac{F(x + th) - F(x)}{t}. \quad (\text{E.3})$$

Furthermore, we say  $J$  is **Gateaux differentiable** at  $x$  in the direction  $h$  if the limit exists.

Additionally, the definition of the Gateaux derivative is well-matched with the following definition of a *partial derivative*. This definition of the partial derivative encapsulates the notion of taking the derivative with respect to a specific direction in one of the arguments.

**Remark E.1 (Partial derivatives).** Let  $X_i$  denote a Fréchet for  $i \in \{1, \dots, n\}$ ,  $X^n := \prod_{i=1}^n X_i$ , and  $Y$  another Fréchet space. We denote the  $i$ -th partial derivative of  $F : X^n \rightarrow Y$  at  $x = (x_1, \dots, x_n) \in X^n$  in the direction  $h$  by  $\frac{\partial F}{\partial x_i}(x)(h)$  and define it by:

$$\frac{\partial F}{\partial x_i}(x)(h) := \lim_{t \rightarrow 0} \frac{F(x_1, \dots, x_i + th, \dots, x_n) - F(x)}{t}. \quad (\text{E.4})$$

Equivalently, it is the Gateaux derivative of the map:

$$y \mapsto F(x_1, \dots, y, \dots, x_n), \quad (\text{E.5})$$

where the  $x_i$  are fixed.

Next, we recall the notion of *continuously differentiability*.

**Definition E.4 (Continuous differentiability).** We say that a map  $F : X \rightarrow Y$  is *continuously differentiable* (or  $C^1$ ) on  $U$  if the limit exists for all  $x \in U$ , all  $h \in X$  and, if  $J' : U \times X \rightarrow Y$  is continuous.

It is an important fact that by employing this definition, the Gateaux derivative of  $J'(x)$  at  $x \in X$  automatically becomes *linear* and *bounded*. We state the most important theorems, following [178].

---



---

**Theorem E.1 (Fundamental theorem of calculus).** *If  $J : U \subset X \rightarrow Y$  is a continuously differentiable map and if the path to  $h$  lies in  $U$ , then:*

$$J(x+h) - J(x) = \int_0^1 J'(x+th)(h) dt \quad (\text{E.6})$$

**Remark E.2.** *It is well-known that Fréchet differentiability (cf. e.g., [77, 7]) implies Gateaux differentiability in Banach spaces. Therefore, when considering a functional  $J : X \rightarrow Y$  between Banach spaces, demonstrating continuous differentiability in the Fréchet sense is sufficient to establish continuous differentiability in the Gateaux sense.*

The second most important theorem which we will often employ in our calculations is the linearity of the Gateaux derivative of a continuously differentiable map.

**Theorem E.2 (Linearity of Gateaux derivative).** *If  $J : U \subset X \rightarrow Y$  is continuously differentiable,  $x \in U$  and  $h, k \in X$ , and  $\alpha, \beta \in \mathbb{R}$  denote two scalars. Then:*

$$J'(x)(\alpha h + \beta k) = \alpha J'(x)(h) + \beta J'(x)(k) . \quad (\text{E.7})$$

This theorem, coupled together with the fundamental theorem of calculus, yields the chain rule for Gateaux derivatives.

**Theorem E.3 (Chain rule).** *If  $J : X \rightarrow Y$  and  $G : Z \rightarrow Y$  are continuously differentiable so is their composition  $J \circ G$  and*

$$(J \circ G)'(x)(h) = J'(G(x))(G'(x)(h)) . \quad (\text{E.8})$$

*Furthermore, if  $f : I \rightarrow X$  is a parametrized continuously differentiable curve and  $J : X \rightarrow Y$  is a continuously differentiable map, then  $J(f(t))$  is also a parametrized continuously differentiable curve such that:*

$$J(f(t))' = J'(f(t))f'(t) . \quad (\text{E.9})$$

Next, since we will often be working with functions whose multiple arguments operate on Fréchet spaces, it is important that we precise the decomposition of the *total derivative* into its partial derivatives.

**Theorem E.4 (Total derivative).** *Let  $J : X \times Y \rightarrow Z$  be a functional between Fréchet spaces and fix  $(x, y) \in X \times Y$ . The partial derivatives  $\frac{\partial J}{\partial x}(x, y) : X \rightarrow Z$  and  $\frac{\partial J}{\partial y}(x, y) : Y \rightarrow Z$  at  $(x, y)$  exist and are continuous if and only if the total derivative  $J'(x, y) : X \times Y \rightarrow Z$  at  $(x, y)$  exists and is continuous. In this case, we have:*

$$\forall (h, k) \in U, J'(x, y)(h, k) = \frac{\partial J}{\partial x}(x, y)(h) + \frac{\partial J}{\partial y}(x, y)(k) , \quad (\text{E.10})$$

*for a neighborhood  $U \subset X \times Y$ .*