Titre: Title:	Non-Euclidean plate theory for the Simulation, Optimization and Automation of shot peen forming
Auteur: Author:	Vladislav Sushitskii
Date:	2022
Type:	Mémoire ou thèse / Dissertation or Thesis
Référence: Citation:	Sushitskii, V. (2022). Non-Euclidean plate theory for the Simulation, Optimization and Automation of shot peen forming [Thèse de doctorat, Polytechnique Montréal]. PolyPublie. https://publications.polymtl.ca/10270/

URL de PolyPublie: PolyPublie URL:	https://publications.polymtl.ca/10270/
Directeurs de recherche: Advisors:	Frederick Gosselin, & Martin Lévesque
Programme: Program:	PhD.

POLYTECHNIQUE MONTRÉAL
affiliée à l'Université de Montréal

Non-Euclidean plate theory for the simulation, optimization and automation of shot peen forming

VLADISLAV SUSHITSKII
Département de génie mécanique

Thèse présentée en vue de l'obtention du diplôme de *Philosophiæ Doctor*
Génie mécanique

Avril 2022

POLYTECHNIQUE MONTRÉAL
affiliée à l'Université de Montréal

Cette thèse intitulée :

Non-Euclidean plate theory for the simulation, optimization and automation of shot peen forming

présentée par **Vladislav SUSHITSKII**
en vue de l'obtention du diplôme de *Philosophiæ Doctor*
a été dûment acceptée par le jury d'examen constitué de :

Farbod KHAMENEIFAR, président
Frédérick GOSSELIN, membre et directeur de recherche
Martin LÉVESQUE, membre et codirecteur de recherche
Étienne MARTIN, membre
Emmanuelle ROUHAUD, membre externe

ACKNOWLEDGEMENTS

I would like to sincerely thank my research supervisors, Prof. Frédérick Gosselin and Prof. Martin Lévesque, for offering me this opportunity and for providing me with all necessary instruments to succeed. Many times throughout these studies I was surprised with an excellent correlation between this project and my previous research experience. And this is not a lucky coincidence but rather a merit of Frédérick's and Martin's exceptional proficiency in human resources management. They were not only *supervising* the activities, but they were also *helping* to break the deadlocks, to formulate the objectives and to accomplish them. With no exception, I could always count on my supervisors if I needed any kind of scientific, administrative or financial support.

As soon as we discovered the works of Prof. Wim van Rees from the Massachusetts Institute of Technology (MIT) in modeling of mechanics of plates and shells, Frédérick immediately proposed to start a full-scale collaboration. I am grateful to Wim for fully supporting this intention, and I am especially indebted to Frédérick and Wim for organization of my internship at the MIT, which was financed by The Aluminium Research Centre – REGAL. Our shoulder to shoulder work with Wim gave a great impetus to our project and advanced me on a professional level.

I am grateful to my fellow colleagues from the Laboratory for Multiscale Mechanics (LM^2), with whom I exchanged many ideas and shared great experiences. In particular, I would like to thank Pierre Faucheux for his mentoring and for his constant will to help throughout the project. I would like to express my gratitude to Dr. Hong Yan Miao for her administrative support, for her participation in planning of the experiments and for her help in preparation of the articles. A significant part of the experimental work was also accomplished by the LM^2 interns Nada Abbassi and Jovan Kostenov, and by the LM^2 member Olivier Duchesne. In addition, the experiments could not have been realized without the support of the Aerospace Technology Center (CTA) team. In this connection, I would like to thank Jeremy Elsek-Valois, Pierre-Olivier Dubois and Raphaël Paradis for their professionalism.

Finally, I gratefully acknowledge the financial support provided by The Natural Sciences and Engineering Research Council of Canada (NSERC), by The Fonds de Recherche du Québec – Nature et Technologies (FRQNT) and The Ministère de l'Économie et de l'Innovation du Québec, by The Aluminium Research Centre – REGAL, and by Aerosphere Inc. I also owe to Aerosphere Inc. for discovering all particularities of the industrial peen forming process.

RÉSUMÉ

La mise en forme par grenaillage est utilisée dans l'industrie aérospatiale pour façonner de larges plaques métalliques, telles que des revêtements d'ailes ou des segments de réservoirs de carburant. Le procédé consiste à traiter une plaque avec un jet de billes rigides (grenailles) qui sont projetées à travers une buse mobile. Un tel traitement induit une déformation plastique dans la couche extérieure du matériau, ce qui fait plier la plaque. Le jet de grenailles est appliqué localement selon un motif de grenaillage, et des différents motifs de grenaillage entraînent des différentes formes courbées.

Nous avons développé un logiciel de simulation de la mise en forme par grenaillage. Le logiciel prédit l'effet d'application d'un motif donné et calcule un motif optimal qui mène à la forme cible. En d'autres mots, le logiciel résout les problèmes direct et inverse respectivement. Le logiciel est basé sur la théorie des plaques non euclidiennes qui fournit des instruments théoriques pour décrire la déformation des structures minces. De plus, le logiciel utilise le concept d'eigenstrains permettant de modéliser efficacement des charges induites par le grenaillage sans simuler chaque impact. Ensemble, la théorie des plaques non-Euclidiennes et le concept d'eigenstrains fournissent une formulation numérique concise pour le logiciel de simulation. Le solveur du problème inverse implémenté de cette manière montre une vitesse de calcul et une précision élevées, ce qui rend le logiciel applicable industriellement.

Un autre avantage de notre solveur du problème inverse est sa capacité d'opérer avec plusieurs régimes de grenaillage. Cette caractéristique améliore la précision de la mise en forme. Lorsque le motif est calculé, notre logiciel le divise en segments traités uniformément, ce qui est requis par les conditions de grenaillage pratiques. La segmentation est effectuée soit avec un algorithme de regroupement (grouping) soit avec un algorithme de partitionnement (clustering). L'algorithme de regroupement est utilisé lorsque les régimes de grenaillage sont fixes et leur ajustement est limité, tandis que l'algorithme de partitionnement calcule automatiquement les régimes de grenaillage optimaux. Les deux algorithmes sont suivis d'un algorithme de filtrage qui corrige les erreurs de segmentation locales. Le solveur du problème inverse ainsi que les algorithmes de regroupement, de partitionnement et de filtrage ont été testés numériquement à l'aide de 200 cas de tests générés aléatoirement.

En s'appuyant sur notre logiciel de simulation, nous avons automatisé le procédé de la mise en forme par grenaillage. À savoir, nous avons développé un flux de travaux comprenant toutes les étapes nécessaires pour la mise en forme automatisée d'une plaque métallique à l'aide d'un robot de grenaillage. Dans cette thèse, nous décrivons comment calibrer le

logiciel de simulation, comment programmer automatiquement le robot en fonction du motif de grenaillage, comment appliquer efficacement le motif prescrit et comment évaluer la qualité de la mise en forme. Lorsque le logiciel est calibré, l'intervention humaine n'est nécessaire que pour l'installation et la désinstallation du composant traité. Le flux de travaux est ajustable pour toutes les formes cibles et, de plus, il est économiquement optimisé. Nous présentons à la fois les détails théoriques et pratiques, donc cette thèse est un guide complet pour l'automatisation du procédé.

Le flux de travaux automatisé a été appliqué pour façonner des plaques d'aluminium de forme libre et pour fabriquer le revêtement d'une aile d'avion modèle. Dans chaque cas, nous caractérisons l'erreur entre la forme simulée et la forme développée, nous révélons les causes de cette erreur et nous suggérons des pistes d'amélioration.

ABSTRACT

Shot peen forming is an industrial process for shaping thin and large metal panels. It is widely used in the aerospace industry to shape, for example, wing skins or fuel tank segments. The process consists in treating the plate with a stream of rigid shot that are projected through a moving nozzle. Such a treatment induces plastic deformation in the outer layer of material and makes the plate bend. The shot stream is applied locally according to the peening pattern, and different peening patterns lead to different curved shapes.

We developed a peen forming simulation software that predicts the effect of applying a given pattern and computes an optimal peening pattern that leads to the target shape, i.e., solves respectively the forward and the inverse problems. The software is based on the theory of non-Euclidean plates, which provides efficient theoretical instruments to describe deformation of thin structures. In addition, the software uses the eigenstrain approach that allows to rapidly model the peening-induced loads without simulating every impact. Together, the theory of non-Euclidean plates and the eigenstrain approach provide a concise numerical formulation for the simulation software. The inverse problem solver implemented in this way shows high computation speed and precision, which makes the software industrially applicable.

Another advantage of our inverse problem solver is its ability to operate with multiple peening regimes. This feature enhances the forming precision. When the pattern is computed, our software splits it into uniformly treated segments, which is required by practical peening conditions. The segmentation is done using either the grouping or the clustering algorithm. The grouping algorithm is used when the peening regimes are fixed and their adjustment is restricted, while the clustering algorithm automatically computes optimal peening regimes. Both algorithms are followed by the filtering algorithm that corrects local segmentation errors. The inverse problem solver along with the grouping, the clustering and the filtering algorithms were tested numerically using 200 randomly generated test cases.

Relying on our simulation software, we have automated the peen forming process. Namely, we developed a workflow including all necessary steps for the automated shaping of a metal plate with a shot peening robot. We describe how to calibrate the simulation software, how to automatically program the robot in accordance with the peening pattern, how to efficiently apply the prescribed pattern and how to evaluate the quality of shaping. When the software is calibrated, the human intervention is necessary only for installation and deinstallation of the treated component. The workflow fits for all target shapes, and, moreover, it is economically optimized. We present both theoretical and practical details, so this thesis is a complete

guide for the process automation.

The automated workflow was applied to shape freeform aluminum panels and the skin of a model airplane wing. In each case, we characterize the error between the simulated and the practically developed shapes, we reveal the causes of this error and we suggest avenues for further improvement.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF SYMBOLS AND ACRONYMS

Symbols

A_c	Current local area
A_{tar}	Target local area
A_{total}	Total area of the plate in its initial configuration
A_{top}	Area of the treated segments on the top side of the plate
A_{bot}	Area of the treated segments on the bottom side of the plate
a_h^S	Arc height of an Almen strip at the saturation point, which defines the peening intensity
$\mathbf{a}$	2×2 matrix defining the local first fundamental form
$(\mathbf{a}_{r,t}, \mathbf{a}_{r,b})$	Two first fundamental forms defining the bilayer rest configuration
$(\mathbf{a}_{r,t}^{ortho}, \mathbf{a}_{r,b}^{ortho})$	Two first fundamental forms defining the bilayer rest configuration for the case of local in-plane orthotropic growth
$(\mathbf{a}_{r,t}^{new}, \mathbf{a}_{r,b}^{new})$	Two first fundamental forms defining the adjusted bilayer rest configuration
$(\mathbf{a}_{rf,t}\ , \mathbf{a}_{rf,b})$	Two first fundamental forms defining the bilayer rest configuration obtained on the final iteration
$\mathbf{b}$	2×2 matrix defining the local second fundamental form
$(\mathbf{a}_c, \mathbf{b}_c)$	The first and the second fundamental forms defining the monolayer current configuration
$(\mathbf{a}_c^{new}, \mathbf{b}_c^{new})$	The first and the second fundamental forms defining the monolayer current configuration, which was obtained by assignment of the adjusted rest configuration and subsequent deformation
$(\mathbf{a}_f, \mathbf{b}_f)$	The first and the second fundamental forms defining the monolayer final configuration
$(\mathbf{a}_{init}, \mathbf{b}_{init})$	The first and the second fundamental forms defining the monolayer initial configuration
$(\mathbf{a}_r, \mathbf{b}_r)$	The first and the second fundamental forms defining the monolayer rest configuration
$(\mathbf{a}_r^{new}, \mathbf{b}_r^{new})$	The first and the second fundamental forms defining the adjusted monolayer rest configuration
$(\mathbf{a}_r^{ortho}, \mathbf{b}_r^{ortho})$	The first and the second fundamental forms defining the monolayer rest configuration for the case of local in-plane orthotropic growth

$(\mathbf{a}_{tar}, \mathbf{b}_{tar})$	The first and the second fundamental forms defining the monolayer target configuration
$\mathcal{C}_c$	Point cloud describing the current configuration
$\mathcal{C}_c^{new}$	Point cloud describing the current configuration obtained by assignment of the adjusted rest configuration and subsequent deformation
$\mathcal{C}_f$	Point cloud describing the final configuration
$\mathcal{C}_{tar}$	Point cloud describing the target configuration
c_k	Numerically optimized coefficients used for reconstruction of the through-thickness eigenstrain profile $\gamma(z)$ with basis functions $\xi_k(z)$
D	Characteristic length of the plate
d_H	The Hausdorff distance
E_{BL}	Elastic energy of a bilayer
E_{ML}	Elastic energy of a monolayer
$\vec{e}_0$, $\vec{e}_1$, $\vec{e}_2$	Edge vectors defining three edges of a triangular element
$\vec{e}_0^{\,r}$, $\vec{e}_1^{\,r}$, $\vec{e}_2^{\,r}$	Edge vectors of a triangular element in the rest configuration
F_p	Stretching force induced by the peening treatment
H_c	Current local mean curvature
H_{tar}	Target local mean curvature
h	The plate thickness
h_*	Thickness of the expanding layer
(h_*^t, h_*^b)	Thicknesses of the top and bottom layers of a trilayer respectively
$\mathbf{I}$	The identity matrix
K	Number of triangular elements in the model
`k`	Number of clusters during the `k`-means clustering
k_A	Ratio between the target local area and the current local area
k_H	Ratio between the target local mean curvature and the current local mean curvature
l_x, l_y	Dimensions of the rectangle formed by the supports of the Almen gage
M	Maximal number of iterations for the inverse problem resolution
M_p	Bending moment induced by the peening treatment
$\vec{m}$	Mapping defining position of each point of the mid-surface in $\mathbb{R}^3$ based on its coordinates in the domain U
N	Number of predefined peening regimes
$\vec{n}$	Unit normal vector
$\vec{n}_0$, $\vec{n}_1$, $\vec{n}_2$	Edge-directors defining normals to the three edges of a triangular element

$\vec{P}$	Distributed force
$\vec{P}_{max}$	Maximal value of the distributed force
$\mathscr{R}$	Rest configuration
$\mathscr{R}_{tar}$	Rest configuration leading to the target shape
$\vec{r}$	Radius-vector defining position of a point in space
$\mathbf{S}$	The shape operator
$\mathbf{S}_c$	Current shape operator
$\mathbf{S}_r$	Rest shape operator
$\mathbf{S}_r^{new}$	Adjusted rest shape operator
$\mathbf{S}_{tar}$	Target shape operator
t_{sat}	Exposure time needed for an Almen strip to develop the arc height a_h^S
U	Domain of $\mathbb{R}^2$ englobing the plate mid-surface in coordinates (x, y)
U_1	Sub-domain of U
u	Deflection of a shot peened coupon measured with the Almen gage
u_{avg}	Average Deflection of four shot peened coupons
$\vec{v}_0$, $\vec{v}_1$, $\vec{v}_2$	Position vectors defining three vertices of a triangular element
(x, y)	Lagrangian curvilinear coordinates parametrizing the plate mid-surface
Y	Young's modulus
z	Through-thickness coordinate
Δz	Deflection of the target shape
Γ	Total eigenstrain
Γ_1	The first eigenstrain moment
Γ_B	Dimensionless peening-induced bending load
$\overline{\Gamma}_B$	Dimensionless peening-induced bending load rescaled by the squared proportion between the characteristic length of the plate D and its thickness h
$\gamma(z)$	Symmetrical through-thickness eigenstrain profile
$\gamma^{el}(z)$	Symmetrical through-thickness elastic strain profile
$\gamma^{res}(z)$	Symmetrical through-thickness residual strain profile
δ	The upper threshold for $\lvert k_H \rvert$
$\boldsymbol{\varepsilon}$	Eigenstrain tensor induced by the peening treatment
$\boldsymbol{\varepsilon}^t$	Eigenstrain tensor induced by the peening treatment in the top layer of a bilayer
$\boldsymbol{\varepsilon}^b$	Eigenstrain tensor induced by the peening treatment in the bottom layer of a bilayer
$\boldsymbol{\varepsilon}^{el}$	Elastic strain tensor

$\boldsymbol{\varepsilon}^{res}$	Residual strain tensor
ε_*	Eigenstrain induced in the expanding layer
$\Delta\varepsilon$	Difference between ε^t and ε^b
$(\varepsilon^{(1)}, \varepsilon^{(2)})$	Eigenstrain characterizing a peening regime
$(\varepsilon^{(1)}_{e,i}, \varepsilon^{(2)}_{e,i})$	Eigenstrain contribution made by element e that serves for adjusting regime i during clustering
$(\varepsilon^t, \varepsilon^b)$	Eigenstrain induced in the top and bottom layers of a bilayer respectively
$(\varepsilon^t_*, \varepsilon^b_*)$	Eigenstrain induced in the top and bottom layers of a trilayer respectively
$(\varepsilon^t_{avg}, \varepsilon^b_{avg})$	The average of the orthotropic eigenstrain in two principal directions on the top and bottom layers respectively
$(\varepsilon^{cen,t}, \varepsilon^{cen,b})$	Coordinates of a group centroid on the plane $(\varepsilon^t, \varepsilon^b)$
$(\varepsilon^{rf,t}, \varepsilon^{rf,b})$	Eigenstrain prescribed to the top and bottom layers by $(\mathbf{a}_{rf,t}\,, \mathbf{a}_{rf,b})$
(θ^t, θ^b)	Angles between the local principal strain direction and the x axis on the top and bottom layers respectively
κ	Curvature measured in the center of an Almen-strip sized coupon
$\kappa^c_x, \kappa^{tar}_x$	Local curvatures in the x-direction for the current and the target shapes respectively
$\kappa^c_{avg}, \kappa^{tar}_{avg}$	Averaged values of the local curvatures in the x-direction for the current and the target shapes respectively
λ_1, λ_2	Lagrange multipliers
μ	Coefficient of proportionality between the dimensionless curvature κh and the dimensionless load Γ_B
ν	Poisson's ratio
$\xi_k(z)$	Basis functions used to numerically reconstruct the through-thickness eigenstrain profile $\gamma(z)$
τ	Threshold for the dimensionless error Ω used as a stop condition for the iterative inverse problem solver
ρ_e	Peening regime assigned to element e
ρ_{e1}, ρ_{e2}, ρ_{e3}	Peening regimes assigned to the three neighbors of element e
$\boldsymbol{\sigma}$	Residual stress tensor
$\sigma_\gamma(z)$	Symmetrical through-thickness residual stress profile
σ_u	The standard deviation in deflection in a set of four shot peened coupons

φ_0, φ_1, φ_2	Angles of inclination of the edge-director vectors $\vec{n}_0$, $\vec{n}_1$, $\vec{n}_2$ with respect to the average of the adjacent face normals
χ	Eigenstrain anisotropy coefficient
χ^{new}	The iteratively adjusted eigenstrain anisotropy coefficient
χ_L	Eigenstrain anisotropy coefficient in the rolling direction of an aluminum plate
χ_T	Eigenstrain anisotropy coefficient in the direction transverse to the rolling direction
χ_*	The average of the absolute values of χ_L and χ_T
ψ	Difference between the areas of segments treated on the top and bottom sides divided by the total area of the plate
Ω	Dimensionless error: Hausdorff distance d_H between the final and the target configurations divided by the square root of the total area of the plate in its initial configuration A_{total}
Ω_z	Dimensionless error: Hausdorff distance d_H between the final and the target configurations divided by deflection of the target shape Δz

Abbreviations

CA	Cellular Automata
CAD	Computer-Aided Design
CMM	Coordinate-Measuring Machine
CST	Constant-Strain Triangle
DCS	Discrete Cosserat Shell
DKS	Discrete Koiter Shell
DOF	Degree of Freedom
FEM	Finite Element Method
GPMC equations	Gauss–Peterson–Mainardi–Codazzi equations
LSCM algorithm	Least Squares Conformal Mapping algorithm
L-BFGS algorithm	Limited-memory Broyden–Fletcher–Goldfarb–Shanno algorithm
L-direction	Rolling direction of an aluminum plate
T-direction	Direction transverse to the rolling direction of an aluminum plate
XRD	X-Ray Diffraction

CHAPTER 1 INTRODUCTION

Fabricating of aircrafts and space missiles requires forming large aluminum plates with high precision. These plates constitute, for example, wing skins, fuselage shells or fuel tank segments. The outlined components often have a nonuniform double curvature, meaning that they are locally curved in two directions and that the local radii of curvature vary over the plate. A cost-effective, yet efficient, shaping method that delivers such forms is called shot peen forming. This treatment consists in projecting a stream of rigid shot towards the surface of the plate. The shot stream plastically deforms the surface and induces a convex curvature on the treated side [1]. The stream width is smaller than the plate size, and the treatment can be applied to both sides of the plate with a variable intensity. This means that the developed curvatures can be locally controlled by altering the peened segments and the peening parameters. Consequently, peen forming does not require any dies, which is the main advantage of this shaping method when compared to the conventional methods such as rolling, pressing, or bending. In terms of cost-efficiency, the absence of dies is especially advantageous when shaping large plates in moderate amounts, which is the case of aerospace projects.

Numerical simulation of shot peen forming is necessary to ensure its precision and time efficiency. The simulation of this process involves the resolution of the *forward* and of the *inverse* problems [2]. The forward problem resolution means computing the deformation resulting from a predefined peening treatment. The inverse problem resolution means defining a necessary treatment to achieve a predefined target shape. A numerical inverse problem resolution is crucial for industrial companies to design a new peen forming procedure. Without it, design of the procedure is a trial and error process, which may last for several weeks or months and leads to the deterioration of many components. Under these conditions, the time and resources spent on the design depend on the craft experience of the specialists in charge and on the complexity of the target shape. This limits, in particular, the size of the shaped components. Therefore, the absence of a numerical inverse problem solver diminishes the economic benefits of shot peen forming.

The publicly available inverse problem solvers lack precision and computational efficiency, which hinders their industrial implementation. The early solvers proposed constant relation between the peening-induced loads and the developed curvature, which is only the case if the induced rotations are small [3, 4]. This limitation was later overcome with the help of the *eigenstrain* approach [5, 6]. In the peen forming case, it involves formulating the peening-

induced loads in terms of plastic strains (eigenstrains), which stay constant regardless of the rotations magnitude. This allows to represent the peened plate as a shell subjected to internal loads and to model deformations resulting from these loads using the finite element method. However, the computation speed of the existing eigenstrain-based solvers is not high enough to make them applicable for large industrial parts. For example, generation of the training examples for the eigenstrain-based neural network presented in Ref. [6] takes 20 hours for a plate meshed with 1024 rectangular elements.

The eigenstrain approach bridges the gap between shot peen forming and the theory of non-Euclidean plates. This theory considers the reconfiguration of plates and shells subjected to internal non-elastic strains that make their metric non-Euclidean. The theory is based on geometrical relations between the surface metric and curvature, which explain, for example, why a flat paper cannot be wrapped over a sphere. The researchers working in this field have overcome the issue of excessive computational burden needed for the inverse problem resolution [7]. Namely, they have developed an analytical approach to the inverse problem resolution, which requires a negligible amount of time with respect to cutting edge numerical methods used in the peen forming context [5, 6].

The peening treatment is applied according to the peening *pattern*, which constitutes the inverse problem solution. The pattern indicates the treated segments of the plate and the set of peening parameters prescribed to each segment. A set of peening parameters is called the peening *regime*. Each peening regime induces a different local deflection. Consequently, the forming of a complexly curved shape requires multiple regimes in the pattern. The segments treated with different regimes must be strictly delimited because the peening equipment does not allow to gradually vary the peening parameters during treatment. However, the existing inverse problem solvers either do not split the pattern into segments [3, 4, 8] or operate with only one predefined regime [5, 6]. In the former case, the pattern must be manually split into segments during post-processing. In the latter case, the optimal regime must be determined through trial and error computations. Hence, an efficient inverse problem solver must compute optimal regimes and split the pattern into segments automatically. We can formulate the division of a pattern into segments as a *clustering* problem, which can be resolved with an appropriate clustering algorithm [9].

A high repeatability of the peen forming process is only ensured if it is applied using an automated equipment. Such equipment typically includes a programmable robotic arm with a peening nozzle attached to its extremity, which is presented in Figure 1.1 a). However, the efficient use of the automated equipment requires additional tools and a well established workflow. Thus, the equipment must be controlled by a numerical inverse problem solver, so

that the trial and error process is avoided. The peening pattern computed by the solver must be automatically transmitted to the robotic arm because manual programming of the robotic arm is time consuming. If the solver formulates the pattern in terms of eigenstrains, then a calibration strategy must be determined to relate the actual peening parameters and the eigenstrain. Another calibration must be done to ensure that the peened segments are treated uniformly and that the rest of the plate is not affected by peening. Finally, the prescribed pattern must be reproduced in the least amount of time, and the treated component must be optimally fixed during treatment. A complete workflow encompassing all these details is not available in the modern literature. The industrial leaders state to have automated the peen forming process but keep details secret [10, 11]. Consequently, companies are still applying peen forming manually, as it is illustrated in Figure 1.1 b).

This thesis focuses on three subjects: the simulation of shot peen forming, the segmentation of the peening pattern and the automation of the peening process. The thesis is structured as follows. Chapter 2 reviews achievements that are currently made in the mentioned domains and examines limitations of the proposed studies. Chapter 3 summarizes the literature review and defines the research objectives. Chapters 4-6 present the achievements made in the framework of this research. Each of these three Chapters is structured as an independent article. Chapter 4 describes the simulation software, which is based on the eigenstrain approach and on the theory of non-Euclidean plates. It also presents results of the numerical validation of the inverse problem solver. Chapter 5 describes the strategy for the segmentation of a peening pattern that automatically computes optimal peening regimes. Chapter 6 describes the automated shot peen forming workflow, which was applied for the experimental validation of the simulation software. Chapter 7 compares the applied modeling methods to similar techniques. Finally, Chapter 8 reviews the findings and provides recommendations for future work.

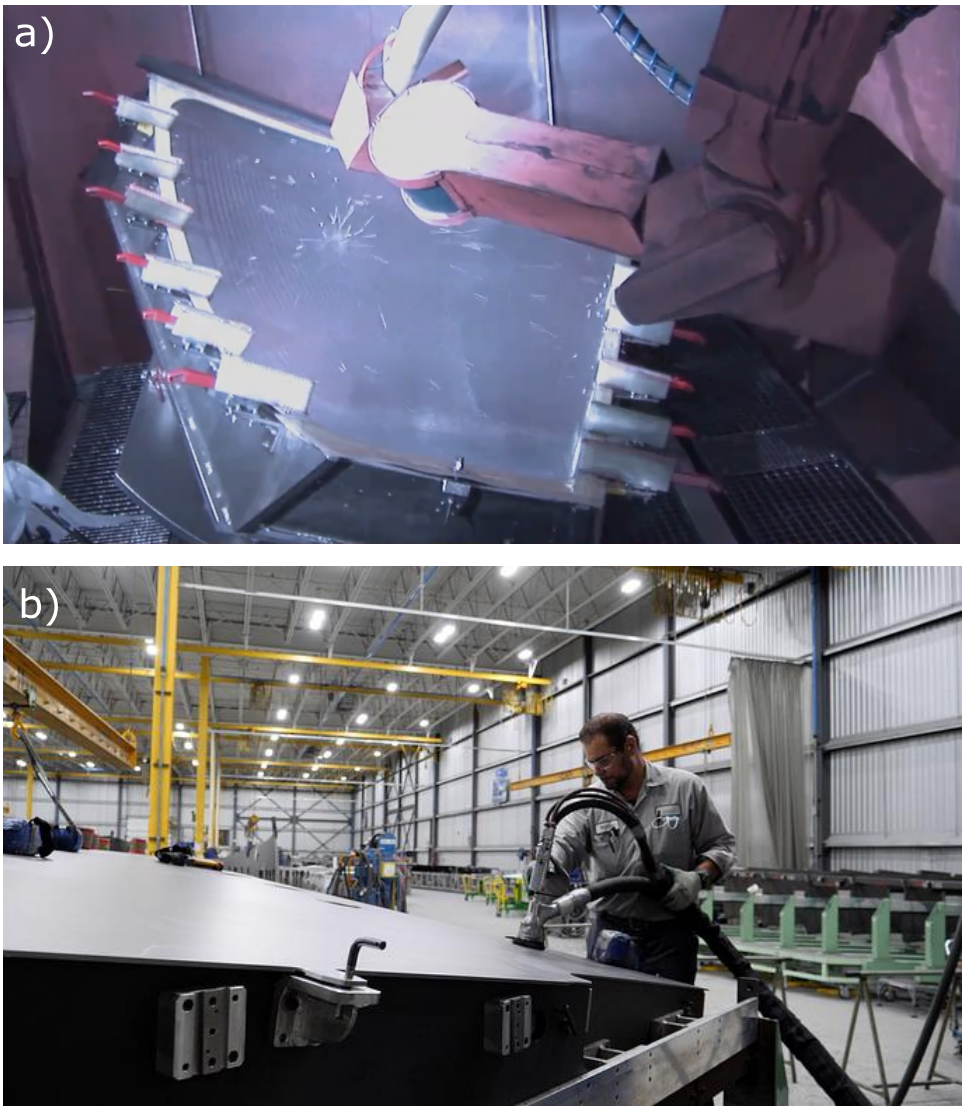

Figure 1.1 Automated and manual shot peen forming. a) A robotic arm equipped with a peening nozzle, which is projecting a stream of shot towards the component. © KSA Kugelstrahlzentrum Aachen GmbH. Reproduced with permission. b) Manual peen forming process. The peening nozzle is moved by the operator, and the reflected particles are absorbed by vacuum surrounding the nozzle. © SONACA Montreal. Reproduced with permission.

CHAPTER 2 LITERATURE REVIEW

We start the literature review with a primer on the shot peen forming process and its mechanics. This brings us to the eigenstrain approach that serves for modeling of the peening-induced loads. In this context, we examine the existing eigenstrain-based methods for the forward problem resolution. We then pass to the theory of non-Euclidean plates, that provides efficient theoretical tools for analysis of the mechanics of plates subjected to eigenstrain. We show that the theoretical and numerical results obtained in this field are applicable for the peen forming simulation. The last three sections of the literature review are dedicated to the aspects of the peen forming process that have a potential for improvement. The first aspect is the numerical inverse problem resolution. We perform a critical analysing the existing methods and examine the improvements that can be brought by the theory of non-Euclidean plates in this domain. The second aspect is segmentation of the peening pattern, which is necessary to make the inverse problem solution applicable in practice with a finite number of treatments. The third aspect is the shot peen forming automation. A conventional method for automation of this process does not exist, so we explore achievements done in this field so far.

2.1 Mechanics of shot peen forming

The peening particles (shot) have spherical form with a diameter of less than 4 mm and are made of steel, glass or ceramics [12]. The shot velocity and their size is set high enough to induce a "fully plastic" material response [13]. Thus, each impact creates a spherical dimple and induces plastic strain in the radial directions around the dimple [14]. The peening nozzle projects numerous shot at a time, and a multitude of impacts plastically deform the whole outer layer of material in the treated segment [15]. This effect is equivalent to stretching of the outer layer, which creates a bending moment and a stretching force influencing the whole structure. Thinner pieces such as metal plates succumb to these loads, i.e., they bend and stretch [16]. This effect underlies shot peen forming.

2.1.1 Characterization of the shot peening treatment

There are two key parameters that describe a shot peening treatment: *intensity* and *coverage*. The intensity is measured by means of a standardized procedure, which is based on the effect of bending of metal plates caused by peening. The procedure involves the peening of Almen

strips – small 76×19 mm sized coupons made of SAE 1070 steel. The thickness of the Almen strips depends on the anticipated intensity and is classified into three types: A, N and C. The strips are clamped and peened with the same parameters but for different exposure times t. When released from clamping, the treated strips adopt an arc-shape. The strips treated for a longer time have a larger deflection (arc height), which is measured using a special device - an Almen gage [17]. The arc height a_h of each strip is then plotted on a coordinate plane as a function of the exposure time, and a trendline named *saturation curve* is traced across the points. The main objective of the intensity measurement is to find the *saturation point* (t_{sat}, a_h^S) on the saturation curve, such that the point $(2t_{sat}, 1.1a_h^S)$ also lies on the saturation curve. The arc height a_h^S serves as the peening intensity value [18]. Figure 2.1 illustrates the saturation curve and the saturation point.

The peening coverage c is defined as a ratio of the impacted area to the total area of the treated specimen. The coverage depends on the exposure time and is usually determined "by observation at 10-30X magnification" [19]. Due to stochastic nature of impacts, the coverage of exactly 100% is not realizable in practice, so 98% is defined as full coverage. The coverage values beyond 98% are achieved by extending the time needed for the full coverage in the corresponding proportion. For example, a coverage of 196% is considered to be reached when a part is peened twice longer than it is necessary to obtain 98% coverage [19].

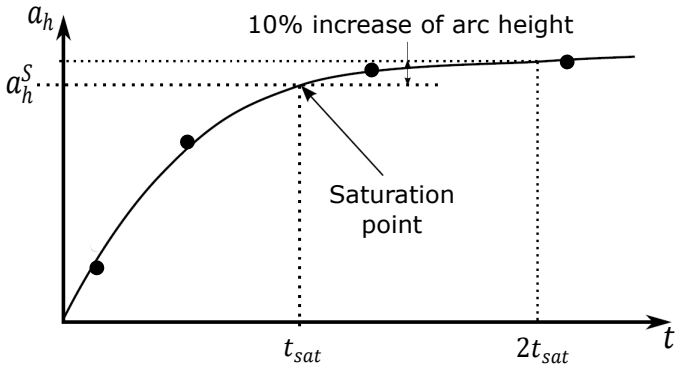

Figure 2.1 The saturation curve serving for the determination of the peening intensity. In this example, four Almen strips were treated with a different exposure time t. Their arc heights a_h were measured and traced as points on a coordinate plane (t, a_h). The saturation curve was then traced as a trendline. The saturation point (t_{sat}, a_h^S) was determined based on the condition that the point $(2t_{sat}, 1.1a_h^S)$ belongs to the saturation curve. The arc height a_h^S defines the peening intensity. Adapted from Ref. [18] with permission from A. Gariépy.

2.1.2 The strain and the stress in the treated segment

Consider a flat metal plate peened uniformly at the full coverage. The surface of the plate is covered with numerous overlapping dimples, and this induced surface roughness can influence the fatigue life both positively and negatively, depending on the specific case [20,21]. However, the surface roughness is insignificant in terms of shot peen forming. Indeed, this process influences the shape of the whole plate, so the small variations of plastic strain over the surface can be neglected in case of full coverage [22]. Consequently, we assume that the generated plastic strain is homogeneous in the in-plane directions. Moreover, the shear components are zero due to translational invariance [23]. However, the plastic strain varies in the through-thickness direction, so the the plastic strain tensor $\boldsymbol{\varepsilon}$ has the following form:

$$\boldsymbol{\varepsilon}(z) = \begin{pmatrix} \varepsilon_{xx}(z) & 0 & 0 \\ 0 & \varepsilon_{yy}(z) & 0 \\ 0 & 0 & \varepsilon_{zz}(z) \end{pmatrix}, \tag{2.1}$$

where x and y are Lagrangian coordinates that follow the plate mid-surface, and z is the through-thickness Lagrangian coordinate. The thickness of the plastically affected outer layer depends on the peening parameters and on the treated material. For example, in the case of aluminum, the layer thickness is typically less than 2.5 mm [2]. The components $\varepsilon_{xx}(z)$ and $\varepsilon_{yy}(z)$ are positive inside this layer and equal zero further from the surface. At the same time, $\varepsilon_{zz} = -(\varepsilon_{xx} + \varepsilon_{yy})$ due to volume conservation.

The plate bends and stretches *elastically* to compensate the loads induced by the plastically deformed layer. The elastic strain provoked in this way is denoted by $\boldsymbol{\varepsilon}^{el}$. Same as $\boldsymbol{\varepsilon}$, the $\boldsymbol{\varepsilon}^{el}$ is uniform along the surface of the uniformly treated segment and depends only on z. The peening-induced strains are small [22], so the resulting strain $\boldsymbol{\varepsilon}^{res}$ may be decomposed as [23, 24]:

$$\boldsymbol{\varepsilon}^{res} = \boldsymbol{\varepsilon} + \boldsymbol{\varepsilon}^{el}. \tag{2.2}$$

If the treated material is isotropic and free of initial stresses, then $\varepsilon_{xx}(z) = \varepsilon_{yy}(z) = \gamma(z)$, $\varepsilon^{el}_{xx}(z) = \varepsilon^{el}_{yy}(z) = \gamma^{el}(z)$ and $\varepsilon^{res}_{xx}(z) = \varepsilon^{res}_{yy}(z) = \gamma^{res}(z)$. Figure 2.2 illustrates the general shape of the profiles $\gamma(z)$, $\gamma^{el}(z)$ and $\gamma^{res}(z)$.

The deformed structure stays residually stressed. The linear elasticity theory relates the residual stress $\boldsymbol{\sigma}$ and the elastic strain $\boldsymbol{\varepsilon}^{el}$ as:

$$\begin{cases} \sigma_{xx} = \frac{Y}{1-\nu^2}\left(\varepsilon^{el}_{xx} + \nu\varepsilon^{el}_{yy}\right), \\ \sigma_{yy} = \frac{Y}{1-\nu^2}\left(\varepsilon^{el}_{yy} + \nu\varepsilon^{el}_{xx}\right), \end{cases} \tag{2.3}$$

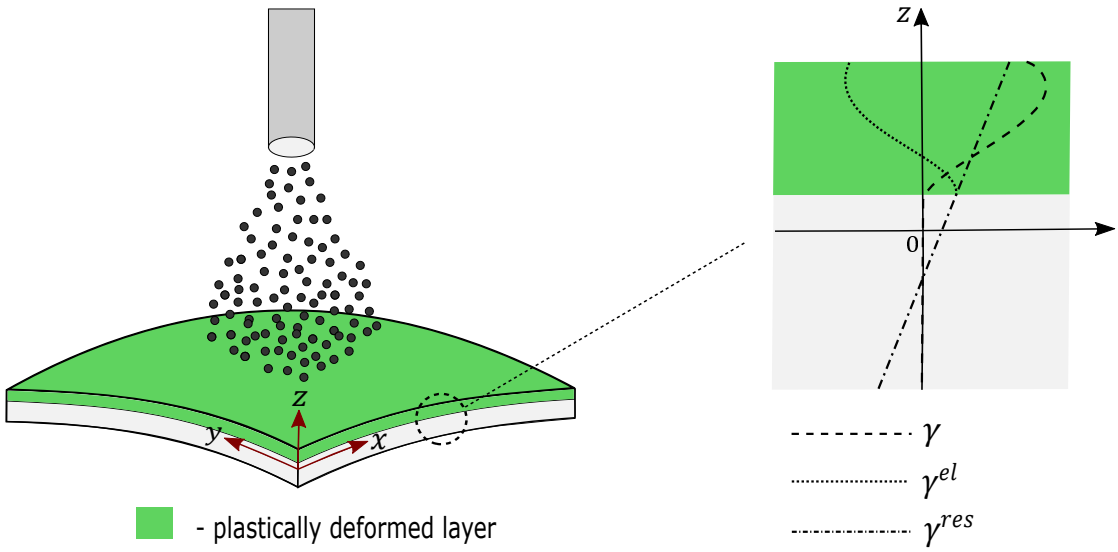

Figure 2.2 Strains in an isotropic plate that was bent by means of a uniform shot peening treatment. The Lagrangian coordinates x and y follow the plate mid-surface, and the Lagrangian coordinate z points in the through-thickness direction. The strains are distributed uniformly along x and y but vary along z. The plastic strain $\gamma(z)$ is induced by the treatment. The elastic strain $\gamma^{el}(z)$ springs back in response to the loads created by $\gamma(z)$. The resulting strain $\gamma^{res}(z)$ is the sum of $\gamma(z)$ and $\gamma^{el}(z)$. Under the Kirchhoff-Love straight normals assumption, the through-thickness profile of $\gamma^{res}(z)$ is a straight line.

where Y is the Young's modulus and ν is the Poisson's ratio of the treated material. All the other components of the residual stress tensor are equal to zero. The non-diagonal components equal to zero due to absence of the shear strains. The fact that $\sigma_{zz} = 0$ follows from the equilibrium equations. Thus, the equilibrium requires $div(\boldsymbol{\sigma}) = 0$ at each point, so the profile $\sigma_{zz}(z)$ is constant. Moreover, the presence of free upper and lower surfaces requires $\sigma_{zz} = 0$ on the surfaces. Consequently, $\sigma_{zz}(z) = 0$ for every z (Ref. [25], Appendix D). For the case of an isotropic non-prestressed material, Equations 2.3 reduce to:

$$\sigma_{xx}(z) = \sigma_{yy}(z) = \sigma_{\gamma}(z) = \frac{Y}{1-\nu}\gamma^{el}(z). \tag{2.4}$$

An approximate expression for the form of the profile $\sigma_{\gamma} = f(z)$ is presented in Ref. [26]. It was developed theoretically and adjusted using empirical results presented in Ref. [27].

2.1.3 The force and the moment induced by the treatment

Now let us consider a clamped plate that underwent a uniform treatment but stays perfectly flat and unstretched due to the clamping. In this case $\boldsymbol{\varepsilon}^{res} = \mathbf{0}$ and $\boldsymbol{\varepsilon}^{el} = -\boldsymbol{\varepsilon}$. For simplicity,

we consider an isotropic material, which is free of initial stresses. The stretching force F_p and the bending moment M_p induced by the treatment are of the same magnitude as the force and the moment acting on the clamping support [16]:

$$F_p = -\int_{-\frac{h}{2}}^{\frac{h}{2}} \sigma_\gamma(z)dz, \tag{2.5}$$

$$M_p = -\int_{-\frac{h}{2}}^{\frac{h}{2}} \sigma_\gamma(z)zdz, \tag{2.6}$$

where h denotes the plate thickness. The displacements are equal to zero in this case, so using Eqn. 2.4 and taking into account that $\gamma^{el}(z) = -\gamma(z)$ we express $\vec{F}$ and $\vec{M}$ in terms of $\gamma(z)$ as:

$$F_p = \frac{Y}{1-\nu}\int_{-\frac{h}{2}}^{\frac{h}{2}} \gamma(z)dz = \frac{Y}{1-\nu}\Gamma, \tag{2.7}$$

$$M_p = \frac{Y}{1-\nu}\int_{-\frac{h}{2}}^{\frac{h}{2}} \gamma(z)zdz = \frac{Y}{1-\nu}\Gamma^1. \tag{2.8}$$

Here, Γ denotes the total plastic strain and Γ^1 stands for the first plastic strain moment [2].

2.2 The eigenstrain approach in the modeling of shot peen forming

The cause for deformation of a shot peened plate is the plastic strain $\boldsymbol{\varepsilon}$ induced by the treatment. Consequently, a natural way to compute the deformed shape of the component is to input $\boldsymbol{\varepsilon}$ in the structure and to compute the elastic springback, i.e, to find $\boldsymbol{\varepsilon}^{el}$ and $\boldsymbol{\varepsilon}^{res}$. The plastic strain $\boldsymbol{\varepsilon}$ fits definition of eigenstrain, which is a term encompassing all nonelastic strains [28]. Moreover, $\boldsymbol{\varepsilon}$ is the only type of eigenstrains present in a shot peened component.

2.2.1 Implementation of an eigenstrain model

The resulting strain $\boldsymbol{\varepsilon}^{res}$ depends only on Γ and Γ^1. These quantities imply integrating over the plate thickness, so different through-thickness eigenstrain profiles $\boldsymbol{\varepsilon}(z)$ can induce the same $\boldsymbol{\varepsilon}^{res}(z)$, provided that Γ and Γ^1 induced by these profiles are equal. Several authors use this rule and input an idealized eigenstrain profile in their simulations [2, 29]. The concept of the idealized eigenstrain profile is illustrated in Figure 2.3. Such a profile simplifies the problem formulation and accelerates the computations. It should be noted that different $\boldsymbol{\varepsilon}(z)$ induce different elastic strain profiles $\boldsymbol{\varepsilon}^{el}(z)$ and thus different residual stress profiles $\boldsymbol{\sigma}(z)$. Given that the main objective of the peen forming simulation is to relate the peening pattern and the resulting deformation, this circumstance is usually ignored.

One strategy for idealization of the eigenstrain profile is homogenization of the eigenstrain through the whole plastically affected layer (Figure 2.3 c), which results in the "*step*" eigenstrain profile:

$$\gamma_{st}(z) = \begin{cases} \varepsilon_* & \text{for} \quad \left(\frac{h}{2} - h_*\right) < z < \frac{h}{2}, \\ 0 & \text{for} \quad -\frac{h}{2} < z < \left(\frac{h}{2} - h_*\right). \end{cases} \tag{2.9}$$

Here, ε_* denotes the eigenstrain assigned to the expanding layer and h_* denotes its thickness. If the plate is treated from both sides, then it takes the form of a *trilayer*:

$$\gamma_{tri}(z) = \begin{cases} \varepsilon_*^t & \text{for} \quad \left(\frac{h}{2} - h_*^t\right) < z < \frac{h}{2}, \\ 0 & \text{for} \quad \left(h_*^b - \frac{h}{2}\right) < z < \left(\frac{h}{2} - h_*^t\right), \\ \varepsilon_*^b & \text{for} \quad -\frac{h}{2} < z < \left(h_*^b - \frac{h}{2}\right). \end{cases} \tag{2.10}$$

Here, h_*^t and h_*^b denote the thickness of the top and bottom layers respectively, while ε_*^t and ε_*^b denote the eigenstrain assigned to the top and bottom layers. If the same peening regimes are simultaneously applied from the top and bottom sides, then $\varepsilon_*^t = \varepsilon_*^b$ and $h_*^t = h_*^b$.

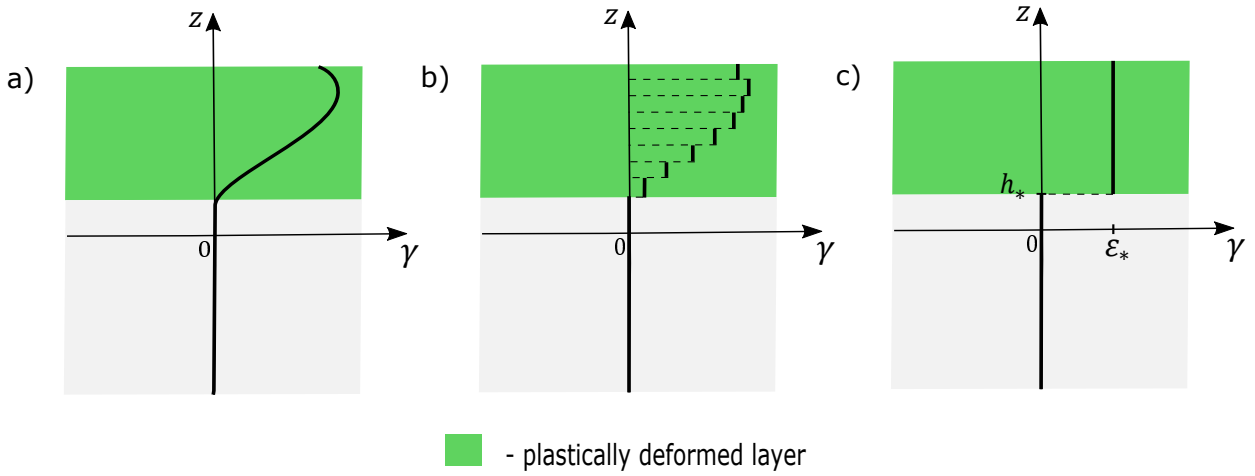

Figure 2.3 Different eigenstrain profiles used as input for the springback simulation. a) A continuous eigenstrain profile induced by shot peening. b) A slightly idealized (discretized) eigenstrain profle. The plastically deformed layer is split into 8 sublayers. Each sublayer expands uniformly through its thickness, but the expansion magnitude is different for each layer. c) The fully idealized eigenstrain profile. The plastically deformed layer is represented as a single layer expanding uniformly through its thickness. Its thickness is denoted by h_*, and the assigned eigenstrain is denoted by ε_*. All three profiles lead to the same deformed shape because they induce the same total eigenstrain Γ and the first eigenstrain moment Γ^1. The profile idealization reduces the number of variables in the model and accelerates the springback simulation.

Otherwise, the plate can be represented as a *bilayer* consisting of two equally thick layers (Ref. [25], Chapter 4):

$$\gamma_{bi}(z) = \begin{cases} \varepsilon^t & \text{for} \quad 0 < z < \frac{h}{2}, \\ \varepsilon^b & \text{for} \quad -\frac{h}{2} < z < 0. \end{cases} \tag{2.11}$$

If the treatment is applied from the top side, then $\varepsilon^t > 0$ and $\varepsilon^b < 0$. Moreover, $|\varepsilon^t| > |\varepsilon^b|$, so that Γ is positive. In the case of the bottom side treatment, the inequalities are opposite. When the treatment is applied from both sides using the same regime, then $\varepsilon^t = \varepsilon^b > 0$.

2.2.2 Relating shot peening parameters with eigenstrain

A set of peening parameters is called a peening *regime*. Every combination of a peening regime and a treated material induces different eigenstrain. Hence, to simulate peen forming using the eigenstrain approach, one has to first determine this relation. The first steps in this direction were done using theoretical considerations. Thus, the depth of the plastically deformed layer was expressed in terms of the shot velocity, their radii and the treated material properties [29,30]. The deformed layer thickness was assumed to be the same as the thickness of the plastically deformed segment induced by a single indentation. This value was obtained by combining solutions for the Hertz and the Boussinesq problems under the assumption that the impact is normal to the surface. This simplified formulation implies no superposition between the impacts and does not provide the eigenstrain magnitude.

An accurate determination of the trough-thickness eigenstrain profile is challenging, and it can only be done by means of numerical simulations or experimental measurement. The main complexity lies in the stochastic nature of the shot projection [31]. Thus, in practice not all impacts are normal to the surface, and the impacts often superimpose with each other. The impacts harden and heat the material thus altering its response to the subsequent indentations [32]. The material hardening, however, is not always proportional to the number of impacts [33]. In addition, the shot are not perfectly spherical [12], and their projection speed may fluctuate due to imperfections in the shot peening system.

Another challenge lies in the anisotropic mechanical properties of aluminum plates that vary depending on the fabrication process [34]. For example, a flat plate produced by rolling contains anisotropic residual stresses [35], and it also demonstrates plastic anisotropy [36]. For this reason, a symmetrical impact induces different through-thickness eigenstrain profiles in the rolling and in the transversal directions [34]. In the context of shot peen forming, the phenomenon of *eigenstrain anisotropy* was examined in Ref. [35].

Numerical impact simulations

One of the first numerical impact simulations was performed by Han et al. [37, 38]. At first, the authors modeled a single impact, and then they simulated 32 consecutive batches of 16 impacts applied on a small representative volume of $8 \times 8 \times 4$ mm. The simulations provided the eigenstrain profiles $\gamma(z)$ for different peening conditions, but these profiles were not validated experimentally. It was observed that a 3D model gives more realistic profiles than a simplified 2D model. It was also noted that the profiles depend on the coefficient of friction between the shot and the treated material. Moreover, it was proved that $\sigma_\gamma(z)$ generated by a single shot is different from that generated by multiple shot. Therefore, the extrapolation of $\sigma_\gamma(z)$ and $\gamma(z)$ generated by one shot over the whole shot peened segment is inaccurate.

Another simulation of multiple impacts was perfomed by Wang et al. [39]. The authors modelled peening of a $20 \times 20 \times 4$ mm plate using the finite element method. They made use of the symmetry and simulated the treatment of a quarter of the plate with 100-800 impacts. This simulation clearly showed that the eigenstrain $\boldsymbol{\varepsilon}$ in the treated segment is organized in layers, or, mathematically speaking, that $\boldsymbol{\varepsilon}$ depends only on z. The same simulation was further improved by Kang et al. [40] using a denser mesh. The computed profile $\gamma(z)$ was compared to through-thickness microhardness measurements in experimentally treated specimens, that confirmed that $\gamma(z)$ was qualitatively well predicted. The simulations were also validated in terms of local deflections of the specimens, that were slightly underestimated.

A similar approach was later presented by Chen et al. [22]. The authors simulated the peening of type C Almen strip — a $76 \times 19 \times 2.39$ mm steel coupon commonly used for the determination of peening intensity. The strip was treated uniformly with arbitrarily distributed shot, which fully covered the strip. Different numbers of shot $(1200 - 6000)$ and different shot velocities $(50 - 75$ m/s) were considered. The simulations examined the behaviour only of a quarter of the strip, and the results were then extrapolated to the whole component. Figure 2.4 illustrates the simulation strategy and the strip in its deformed state. It was proven that, in the case of full coverage, the eigenstrain $\boldsymbol{\varepsilon}$ and the residual stress $\boldsymbol{\sigma}$ have a layered structure, which is visible in Figure 2.4 b). It also was noted that, under the described conditions, the simulated springback did not influence the eigenstrain, so $\boldsymbol{\varepsilon}$ was constant before and after the springback.

All impact simulations presented in this subsection were performed using the dynamic explicit analysis. The shot were modelled as undeformable spheres, and the impacted surface was densely meshed either with tetrahedral [39] or with hexahedral [22, 38, 40] volumetric finite elements. An accurate dynamic simulation required a dense mesh: the element size had to be

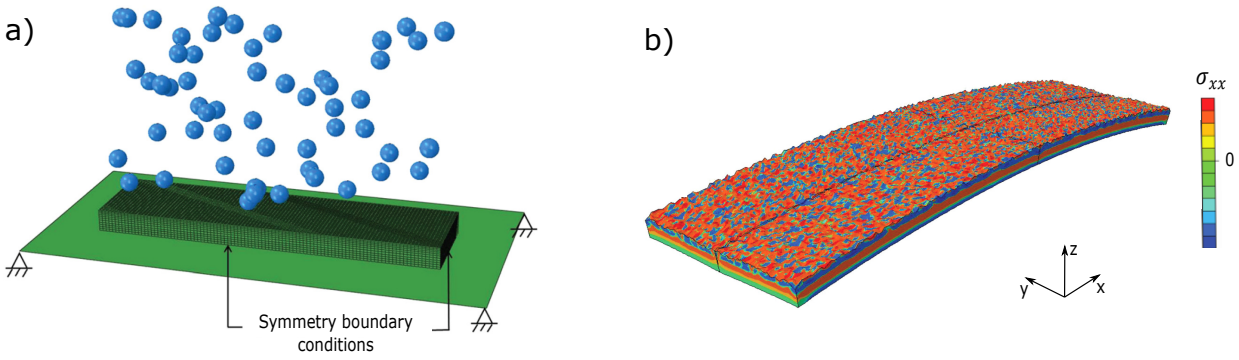

Figure 2.4 Numerical simulation of 6000 peening impacts impinging an Almen strip conducted by Chen et al. [22]. a) The researchers simulated impacting of a quarter of the strip with 1500 shot at the speed of 75 m/s. The symmetry boundary conditions were imposed to the internal boundaries. The strip was meshed with hexahedral finite elements and supported by a rigid surface. The shot were modeled as rigid spheres. b) The strip in its deformed state. The color indicates local residual stress σ_{xx} along the x-axis. In this research, (x, y, z) is the absolute coordinate system, and the strip is oriented along the x-axis. The figure demonstrates that despite local variations, the through-thickness residual stress profile has a layered form. Adapted from Ref. [22] with permission from the American Society of Mechanical Engineers.

ten times smaller than the shot diameter [37]. At least nine elements were used in the through-thickness direction as well [40]. This made such computations time-consuming. Thus, the model presented in Ref. [22] was composed of 84774 elements, and the simulation took 20 hours involving a supercomputer. The model presented in Ref. [40] was composed of 150 000 elements and it required 6 minutes to simulate each impact. A long computation time makes dynamic impact simulations unsuitable for industrial applications, unless a supercomputer is involved.

Direct experimental determination of eigenstrain

Given that the eigenstrain stays constant if the induced rotations are small [22], it can be determined on a thick shot peened specimen by means of X-ray diffraction (XRD). The corresponding procedures are described in Refs. [41] and [21]. Essentially, the XRD measures local $\gamma^{el}(z)$ profiles. The $\gamma(z)$ profiles can then be easily determined relying on Eqn. 2.1 and by assuming that the $\gamma^{res}(z)$ profile is a straight line [41]. The specimens presented in Ref. [21] did not undergo any springback, so, in this case, the eigenstrain was measured directly assuming that $\boldsymbol{\varepsilon}^{el} = -\boldsymbol{\varepsilon}$. The performed measurement allowed Zhang et al. [41] to conclude that the $\gamma^{res}(z)$ profiles do not depend on the thickness of the treated material, if the thickness is sufficiently large. The authors assume that this rule is valid for all thicknesses

that are at least twice larger than the depth of the plastically affected layer.

Although XRD proved its efficiency for theoretical research, its use for everyday eigenstrain measurement in industry is cumbersome. Thus, the process is time-consuming, and the result may depend on the material texture, the grain size and the net intensity. Also, the depth of penetration of the X-rays is very low (few microns), so one has to drill or cut the specimen to examine the eigenstrain distribution along the whole thickness. This may introduce a considerable error in the measurements [42]. Finally, the residual stress measurement requires costly equipment and highly-qualified staff.

Combination of modeling and experiments

Another strategy for the eigenstrain characterization involves experiments and an already implemented springback simulation model. For example, the eigenstrain profile $\boldsymbol{\varepsilon}(z)$ can be reconstructed from the measured residual stress profile $\boldsymbol{\sigma}(z)$. The residual stress, in turn, can be measured with cheaper and less time-consuming methods than XRD. For instance, this can be done using the hole drilling method [43]. The eigenstrain profile $\gamma(z)$ in this case is numerically optimized using the simulation model in the way that it generates the same residual stress profile as the measured one. The profile $\gamma(z)$ can be represented as the sum of basis functions $\xi_k(z)$ [2, 44]:

$$\gamma(z) = \sum_{k=1}^{N} c_k \xi_k(z), \tag{2.12}$$

where

$$\xi_k(z) = \begin{cases} \left(\frac{h}{2} - h_* - z\right)^{k+2} & \text{for} \quad \left(\frac{h}{2} - h_*\right) < z < \frac{h}{2}, \\ 0 \quad \text{for} \quad -\frac{h}{2} < z < \left(\frac{h}{2} - h_*\right). \end{cases} \tag{2.13}$$

The numerically optimized variables are the coefficients c_k and the thickness of the plastically deformed layer h_*.

If the springback simulation model relies on an idealized eigenstrain profile, then the determination of the real eigenstrain profile presents an unnecessary additional step and can be omitted. Indeed, the single-layer eigenstrain profile in this case is determined by only two variables: h_* and ε_*. If one of them is determined in advance, then the other can be calibrated experimentally. This strategy was adopted in Ref. [29]. The h_* value was determined as a function of the process parameters relying on the theoretical results of Al-Hassani et al. [30]. The ε_* value was adjusted to fit the simulated deflection of a $76 \times 19 \times 3$ mm aluminum coupon to the experimentally measured value. This strategy works without any

supplementary equipment or numerical models in addition to the shot peening machine and the eigenstrain-based springback simulation model.

The analytical relation for h_* presented in Ref. [30], however, is not exact. It is based on the depth of penetration of the plastic strain induced by one impact, but the simulations performed by Han et al. demonstrated that the plastic strain depth induced by multiple impacts is different [37]. Nevertheless, given that the bilayer deflection depends only on Γ^1, the parameter ε_* can be adjusted to accurately reproduce Γ^1 even if h_* is inaccurately estimated. In turn, the total eigenstrain Γ in this case is different from the real one, and thus the stretching of the specimen is not simulated correctly. This circumstance must be taken into account for the applications where an accurate simulation of the stretching is important.

2.2.3 Resolution of the forward problem using the eigenstrain approach

In the eigenstrain formulation, the forward problem consists in determining of the elastic springback induced by a predefined eigenstrain field.

Theoretical challenges

Shot peening induces small eigenstrain of the order of 10^{-3} [2, 35]. The elastic springback also implies small strains, but the arising displacements may be large [22]. At the same time, the treated plate stays residually stressed after the springback because its only stress-free configuration implies delamination. Large displacements and the absence of an integral stress-free state make standard linear elastic theories inappropriate for the forward problem resolution in a general case [45]. Thus, due to large displacements, the plate may exhibit a geometrically nonlinear behaviour. This means that the local elastic strain depends not only on the eigenstrain, but also on the geometry of the plate. For instance, Pezzula et al. showed that the directional curvature of a bilayer plate subjected to eigenstrain is different from the curvature of a beam cut out of this plate in the same direction [46] (see Figure 2.5). The standard equilibrium equations for the total forces and moments are unsuitable in this case either. They are formulated in terms of stresses, which are related to the elastic strains. However, the elastic strains are derived with respect to the virtual stress-free state that the plate can never adopt, and this factor alters the plate mechanics [47].

Numerical achievements

Notwithstanding the complexity of an analytical forward problem resolution, this problem can always be solved numerically using the finite element method in its shell [2, 48] or vol-

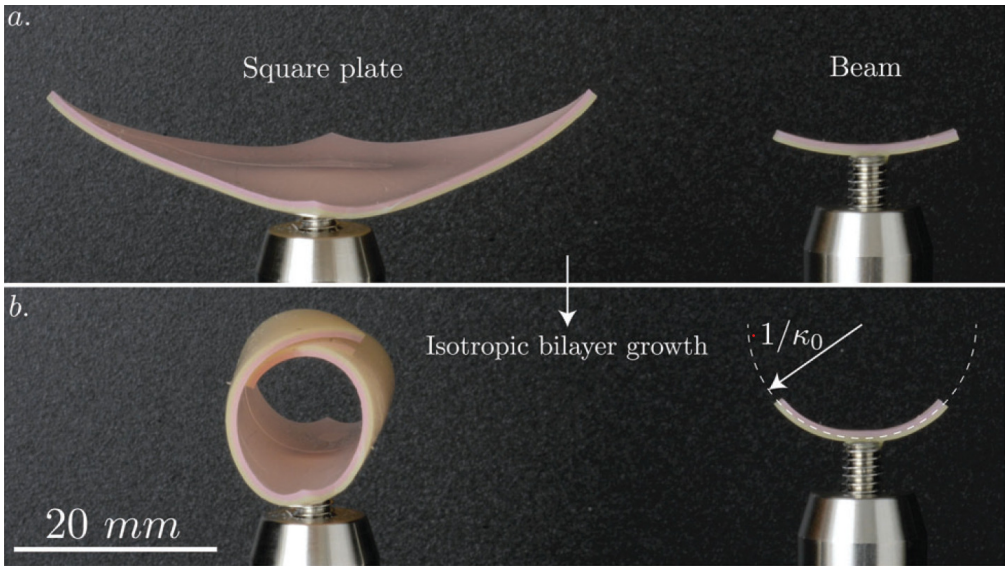

Figure 2.5 Bending of a bilayer plate and a bilayer beam subjected to eigenstrains. The two layers of the plate are made of two different polymer materials (PVS Zhermack Elite Double 32 and 8), and the beam is cut out of the plate. a) The initial state of the plate and the beam. b) The polymer chain flow induces eigenstrains, i.e., it makes the bottom layer expand uniformly. As a result, the plate adopts a cylindrical shape, and the beam bends. However, the curvature of the beam κ_0 is lower than the principal curvature of the cylinder. Source: [46]. Reproduced with permission from the Royal Society of Chemistry.

umetric [49, 50] formulation. The number and the density of integration points along the thickness can be varied thus allowing flexible simulation of different through-thickness eigenstrain profiles. The commercial FEM packages allow to efficiently model eigenstrain as an irreversible thermal expansion, which is mechanically equivalent to the plastic strain [51]. Otherwise, the eigenstrain can be induced by compression of the upper layer of material [52].

One of the first eigenstrain-based peen forming simulations is presented in Ref. [52]. The authors simulated a uniform treatment of a small $50 \times 10 \times 0.9$ mm strip and a 0.9 mm thick disc having 25 mm in diameter using volumetric finite elements. The eigenstrain was applied by fixing the nodes along the thickness and squeezing the outer surface of the material. After that, the nodes were released and the springback was computed. The proposed approach was innovative for the time of its creation, but it has two shortcomings. The first one was a complex relation between the squeezing force and the shot peening parameters, that was not examined precisely. The second one consists in a necessarily large number of elements along the thickness, which limits the approach to small components.

Another simulation involving volumetric finite elements was performed by Chaise et al. [49, 50]. The authors simulated a uniform treatment of $60 \times 18 \times 2$ mm coupons using thermal

expansions to reproduce the through-thickness eigenstrain profile. The simulations were reproduced experimentally, and the deflection of the coupons differed from the simulated one by less than 0.2 mm. Nevertheless, the use of volumetric finite elements made the springback computation expensive, so that the forward problem resolution for the $60 \times 18 \times 2$ mm specimen took "few hours".

Wang et al. [29] considered a similar problem: the researchers simulated the uniform treatment of a $76 \times 19 \times 3$ mm coupon. In this paper, the forward problem was solved using shell finite elements, and the through-thickness eigenstrain profile was idealized. The error in deflection between the simulations and the experiments was also of the order of 10^{-2} mm. The shell finite element formulation takes less computational time than the volumetric formulation. This study also shows that this formulation provides high accuracy if the eigenstrain is correctly calibrated in advance.

Several authors used the eigenstrain approach to simulate the peen forming of plates that are significantly bigger than the Almen strip. This was commonly done using shell finite elements and the thermal expansion approach. For instance, Levers et al. [48] simulated the forming of a wing skin panel. However, the paper only describes the concept without providing any validation details. More recently, Faucheux et al. [2] simulated a uniform one-side treatment applied on 1×1 m aluminum plates of different thicknesses: 5, 10 and 15 mm. The study showed that a uniformly peened plate adopts either spherical or cylindrical shape. Moreover, if the treatment is fixed, then this choice is determined by the plate thickness. Thus, the 5 mm thick plate morphed into a cylindrical patch, while the 15 mm thick plate adopted a spherical shape. This conclusion was also proven experimentally, which is illustrated in Figure 2.6.

In addition, Faucheux [25] and Miao et al. [35] simulated the uniform treatment of rectangular aluminum plates with various aspect ratios and dimensions, ranging from 254 to 1016 mm. The researchers examined the influence of material plastic anisotropy, of the initial stresses and of the additional prestress on the eigenstrain anisotropy. The eigenstrain anisotropy was included in these simulations using the coefficient χ implying the relation

$$\varepsilon_{xx} = (1 + \chi)\varepsilon_{yy}/(1 - \chi) \tag{2.14}$$

at any point of the treated segment. The simulations showed that χ is another factor that influences the transition between the spherical and the cylindrical shapes. It was demonstrated that, in the cylindrical regime, the rectangular plates always bend along their long direction in the case of $\chi = 0$. Otherwise, when $\chi \neq 0$, the bending direction is determined both by χ and by the aspect ratio of the plate.

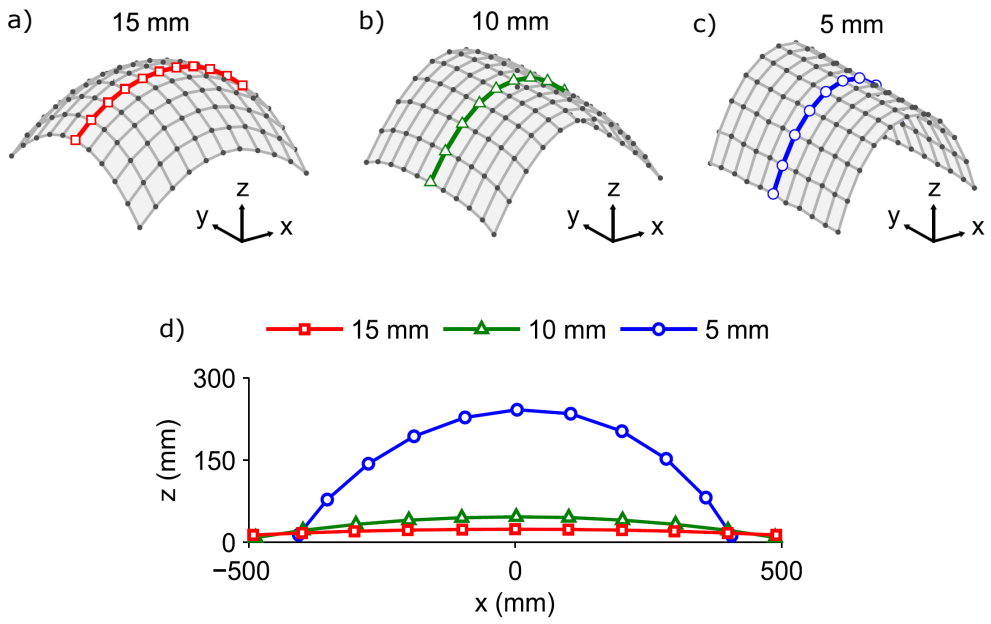

Figure 2.6 Deflection of uniformly treated 1 × 1 m 2024-T3 aluminum plates of different thicknesses described in Ref. [2]. The plates were treated at 80% coverage with Almen intensity of 0.57 mm C (Section 2.1.1). The 15 mm thick plate adopted a spherical shape (a), the 10 mm thick plate adopted a quasi-cylindrical shape with a slight curvature in the y-direction (b), and the 5 mm thick plate adopted a purely cylindrical shape (c), as was predicted by the numerical simulations. The deformation of the plates is magnified for illustration purposes. The developed shapes were scanned with a coordinate-measuring machine (CMM). Figure d) traces vertical cross-sectional deflection of the three plates. In this research, (x, y, z) is the absolute coordinate system. Adapted from Ref. [2] with permission from Elsevier Science & Technology Journals.

The experimental validation of the simulations presented in Refs. [2, 25, 35] demonstrated a good qualitative accordance, but a significant error was observed in terms of the predicted curvature. The authors explain it with the fact that the plates were free to deform during treatment. Thus, the progressive deformation of the plates created the effect of prestress, which altered the local eigenstrain. Consequently, the eigenstrain did not correspond to the simulated one. The influence of the prestress on the developed eigenstrain was further examined in detail in Ref. [53].

The shell finite elements along with the thermal expansion approach also provide enough versatility to simulate the application of more complex peening patterns. Each finite element can be subjected to the thermal expansion independently, so the peening pattern essentially denotes a set of eigenstrains prescribed to each element. With this, Miao et al. [5] simulated application of peening patterns that made 305 × 305 mm sized aluminum plates adopt a

cylindrical or a saddle shape. The experimental validation showed that the overall shape of each plate was predicted correctly, so the errors in deflection were of the order of 0.1 mm. In the worst case the error constituted 14% of the total deflection. Same as in the case of uniformly treated plates, the error was influenced by progressive deformation of the plates during treatment. In addition, the model did not take into account the anisotropy of eigenstrains induced in shot peened aluminum plates [25, 35], and this factor also increased the simulation error.

2.3 The theory of non-Euclidean plates

The eigenstrain is an internal load, and the elastic springback does not depend on its physical cause. This allows to solve the forward problem for shot peen forming using general theories describing the springback of elastic plates subjected to eigenstrain. Among those, the theory of non-Euclidean plates is specifically adapted to describe large displacements in the absence of an integral stress-free configuration. In fact, the latter condition gives the name to the theory: the plate is called non-Euclidean when its *rest*, i.e., stress-free configuration, is not embeddable in Euclidean space without the loss of integrity. The theory of non-Euclidean plates was initially developed and structured by Efrati and collaborators [45, 47, 54].

2.3.1 Fundamental principles

The theory of non-Euclidean plates adopts standard Kirchhoff-Love assumptions:

- straight lines normal to the mid-surface remain straight and normal to the mid-surface after deformation;
- the distance of any material point to the mid-surface remains unchanged after deformation.

This allows to associate the plate shape with the shape of its mid-surface. The mid-surface is parameterized with Lagrangian curvilinear coordinates $(x, y) \in U \subset \mathbb{R}^2$. The domain U determines the borders of the mid-surface. The position of each point of the mid-surface in 3D space is given the mapping $\vec{m} : U \mapsto \mathbb{R}^3$ [47]. The mid-surface shape is uniquely described using standard quantities coming from differential geometry: the first and the second fundamental forms. The former defines the local stretching and is defined by the 2×2 matrix $\mathbf{a}$:

$$\mathbf{a} = \begin{bmatrix} \partial_x \vec{m} \cdot \partial_x \vec{m} & \partial_x \vec{m} \cdot \partial_y \vec{m} \\ \partial_y \vec{m} \cdot \partial_x \vec{m} & \partial_y \vec{m} \cdot \partial_y \vec{m} \end{bmatrix}. \tag{2.15}$$

The latter defines the local curvature and is defined by the 2×2 matrix $\mathbf{b}$:

$$\mathbf{b} = \begin{bmatrix} \partial_{xx}\vec{m} \cdot \vec{n} & \partial_{xy}\vec{m} \cdot \vec{n} \\ \partial_{yx}\vec{m} \cdot \vec{n} & \partial_{yy}\vec{m} \cdot \vec{n} \end{bmatrix}, \tag{2.16}$$

where $\vec{n}$ stands for the local normal vector of a unit length. In these equations, ∂_x and ∂_y denote the partial derivatives $\partial/\partial x$ and $\partial/\partial y$ respectively, so $\partial_x\vec{m}$ and $\partial_y\vec{m}$ are the local tangent vectors. Figure 2.7 illustrates the concept of surface parameterization and the vectors involved.

The two fundamental forms uniquely describe a surface in Euclidean space under the condition that they are *compatible*. Mathematically, this means that they must satisfy the Gauss–Peterson–Mainardi–Codazzi (GPMC) equations, which may be found, for instance, in Ref. [55]. The introduction of eigenstrain is similar to the prescription of a new rest configuration to the plate, which is described by the two fundamental forms $(\mathbf{a}_r, \mathbf{b}_r)$. If $\mathbf{a}_r$ and $\mathbf{b}_r$ do not satisfy the GPMC equations, which is often the case, then the rest configuration is not embeddable in Euclidean space without the loss of integrity, so the plate is non-Euclidean. The *final* shape, i.e., the shape that the plate actually adopts, is described by compatible fundamental forms $(\mathbf{a}_f, \mathbf{b}_f)$. Hence, the forward problem in this formulation consists in the determination of $(\mathbf{a}_f, \mathbf{b}_f)$ as a function of $(\mathbf{a}_r, \mathbf{b}_r)$. This is done by minimizing the elastic energy functional E_{ML}, which is based on the Saint Venant - Kirchhoff hyperelastic material model [56], with respect to $(\mathbf{a}_f, \mathbf{b}_f)$ [7, 45]:

$$E_{ML} = \frac{1}{2}\int_U \left[\frac{h}{4}\left\|\mathbf{a}_r^{-1}\mathbf{a}_f - \mathbf{I}\right\|_e^2 + \frac{h^3}{12}\left\|\mathbf{a}_r^{-1}(\mathbf{b}_f - \mathbf{b}_r)\right\|_e^2\right]\sqrt{\det \mathbf{a}_r}\, dxdy, \tag{2.17}$$

where $\|\cdot\| = \alpha Tr^2(\cdot) + 2\beta Tr(\cdot^2)$ with $\alpha = Y\nu/(1-\nu^2)$ and $\beta = Y/(2+2\nu)$.

2.3.2 Theoretical results

An important theoretical achievement, which is applicable for the shot peen forming simulations, was made by van Rees and collaborators [7]. It was proven that every non-Euclidean plate having a rest configuration $(\mathbf{a}_r, \mathbf{b}_r)$ can be represented as a bilayer subjected to an *in-plane* eigenstrain on one or both layers. Such in-plane eigenstrain prescribes new rest first fundamental forms $(\mathbf{a}_{r,t},\ \mathbf{a}_{r,b})$ to the top and the bottom layers respectively. The plate is non-Euclidean if $\mathbf{a}_{r,t}$ and $\mathbf{a}_{r,b}$ are locally different because such rest configuration prescribes delamination. The two formulations of the rest configuration can be easily transformed one

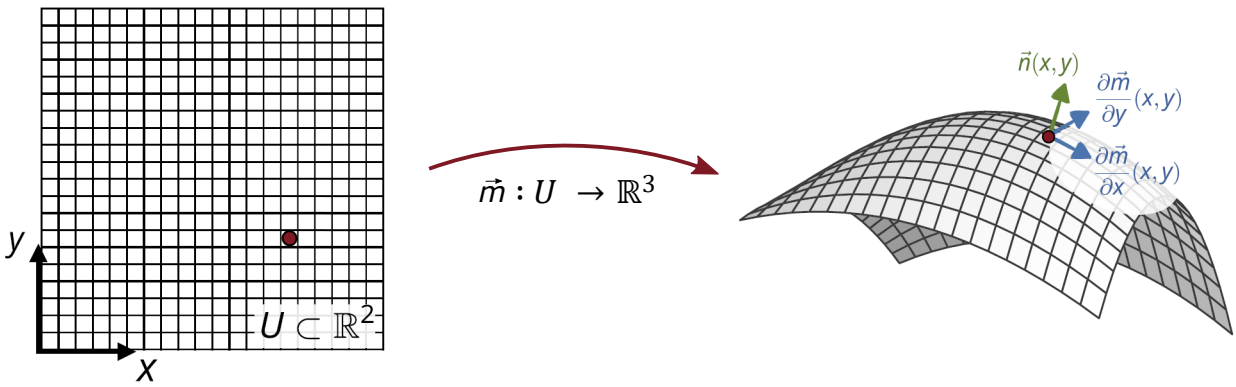

Figure 2.7 Surface pameterization. Each point of the surface is endowed with a fixed 2D coordinate (x, y), and the mapping $\vec{m} : U \mapsto \mathbb{R}^3$ defines position of the point in space. The mapping thus defines the field of local normals $\vec{n}(x, y)$ and the field of local tangent vectors $\partial\vec{m}(x, y)/\partial x$ and $\partial\vec{m}(x, y)/\partial y$. These quantities participate in the computation of the local fundamental forms $\mathbf{a}$ and $\mathbf{b}$ that uniquely define the shape of the surface. Adapted from Ref. [7] with permission from the Proceedings of the National Academy of Sciences.

into another [7]:

$$\begin{cases} \mathbf{a}_r = \dfrac{1}{2}\left(\mathbf{a}_{r,b} + \mathbf{a}_{r,t}\right), \\ \mathbf{b}_r = \dfrac{3}{4h}\left(\mathbf{a}_{r,b} - \mathbf{a}_{r,t}\right). \end{cases} \tag{2.18}$$

Inversely,

$$\begin{cases} \mathbf{a}_{r,t} = \mathbf{a}_r - \dfrac{2h}{3}\mathbf{b}_r, \\ \mathbf{a}_{r,b} = \mathbf{a}_r + \dfrac{2h}{3}\mathbf{b}_r. \end{cases} \tag{2.19}$$

Here, it is assumed that the bilayer consists of two equally thick layers. Such bilayer formulation of a non-Euclidean plate exactly corresponds to the idealized peening-induced eigenstrain profile (Eqn. 2.11). In this case, $\mathbf{a}_{r,t}$ and $\mathbf{a}_{r,b}$ are defined by ε^t and ε^b respectively. To find the final configuration $(\mathbf{a}_f\,, \mathbf{b}_f)$ induced by peening, one can transform $(\mathbf{a}_{r,t}\,, \mathbf{a}_{r,b})$ into $(\mathbf{a}_r\,, \mathbf{b}_r)$ using Equations 2.18 and then minimize E_{ML} with respect to $(\mathbf{a}_f\,, \mathbf{b}_f)$. Another solution would be to directly substitute $(\mathbf{a}_{r,t}\,, \mathbf{a}_{r,b})$ to the bilayer elastic energy functional E_{BL} formulated in Ref. [7] and to minimize it with respect to $(\mathbf{a}_f\,, \mathbf{b}_f)$.

In certain cases, the theory of non-Euclidean plates also allows to analytically describe the final shape of flat plates subjected to eigenstrain. Thus, Pezzulla et al. [57] described the curvature of a monolayer disc induced by isotropic swelling of its centre or of its outer part. The curvature depended on the swelling magnitude and on the ratio between the swelling and the passive areas. The analytical results showed good agreement with numerical

simulations and with experiments conducted using swelling elastomer sheets. It was observed that swelling of the central part induces a spherical shape, while swelling of the borders leads to a saddle shape. These experiments can be reproduced using peen forming by applying treatment from both sides on a metal disc in the segments that must undergo swelling.

Pezzulla et al. [46,58] also considered the case of a bilayer with an isotropically swelling upper layer. In terms of peen forming, this case corresponds to uniform treatment of a plate from one side (Eqn. 2.9). It was proved analytically that the plate in this case adopts either a spherical or a cylindrical shape depending on which shape minimizes the elastic energy (Eqn. 2.17). The authors derived relations for the curvature of the plate in both configurations. They also formulated the criterion of transition between the two configurations. Essentially, it was proven that a larger thickness promotes the adoption of a spherical shape, while a larger area of the plate, a larger Poisson's ratio and a larger eigenstrain magnitude promote the adoption of a cylindrical shape. This result is consistent with the simulations and experiments performed by Faucheux et al. in the field of shot peen forming [2].

2.3.3 Numerical results

The shell finite element method can be reformulated using the theory of non-Euclidean plates [59]. In this formulation, the eigenstrain is embedded in the rest configuration imposed on the plate, which eliminates the need to simulate its mechanical cause. The shell elements are triangular, and the fundamental forms are constant inside each element. Consequently, the integration in the expression for the global elastic energy functional E_{ML} (Eqn. 2.17) transforms into the sum over all elements of the mesh. The forward problem is solved through numerical minimization of E_{ML} with respect to the local final fundamental forms $(\mathbf{a}_f, \mathbf{b}_f)$, that, in turn, are defined by the positions of vertices and normals. As stated in the Appendix to Ref. [7], "in the context of finite element methods this numerical method is essentially a geometric reformulation of the constant-strain triangle (CST) for the membrane energy, and the Morley triangular element [60] for the bending energy [61, 62]".

A validation of this numerical method against theoretical results is presented in Ref. [63]. The authors solved the forward problem for several cases that admitted an analytical solution, such as a spherical cap, a hyperboloid cap or a cylindrical patch. The first and the second fundamental forms of these surfaces are well known, and they were imposed as the rest configuration $(\mathbf{a}_r, \mathbf{b}_r)$ to a flat unstretched square shell. This monolayer formulation of the rest configuration can be easily transformed into the bilayer case, which is closer to the shot peen forming simulation, by inverting Equations 2.18. In the considered cases, the rest fundamental forms were *compatible* because they described existing surfaces. Consequently,

the minimization of E_{ML} was supposed to provide $\mathbf{a}_f = \mathbf{a}_r$ and $\mathbf{b}_f = \mathbf{b}_r$. However, this was not exactly the case because, numerically, E_{ML} is minimized not with respect to $(\mathbf{a}_f, \mathbf{b}_f)$, but with respect to the positions of vertices and normals in the final mesh configuration. The final configurations were compared to the analytical shapes in terms of the maximal local discrepancy (the Hausdorff distance). The initial square shell had a unit area, and the Hausdorff distance for the three mentioned final shapes lied between 1×10^{-2} and 3.5×10^{-2} for a mesh consisting of 2686 vertices. The simulations also revealed that the error was slightly mesh-dependent but did not always decrease with a denser mesh.

The numerical implementation of the theory of non-Euclidean plates was also validated qualitatively against experiments with 4D printed shells [59, 63]. Such structures are printed flat, and, similar to shot peen forming, their deformation is caused by the eigenstrain. The eigenstrain is embedded in the structures during the printing phase and is "activated" with external stimuli, such as heat, humidity, or light [64]. For instance, van Rees et al. simulated the reconfiguration of 4D printed polymers that adopted complex shapes, such as a helicoid, a catenoid or a sombrero [59]. Each shell consisted of two layers containing oriented filaments that swelled when immersed in water. The technical details for printing of such structures are presented in Ref. [65]. The numerical simulations of van Rees et al. provided excellent visual consistency with the shapes developed in practice, which is illustrated in Figure 2.8. Nevertheless, the error was not quantified. In a similar fashion, Chen et al. [63] simulated the wrinkling of a polymer disc with radially varying eigenstrain activated in a hot bath [66] and of a bilayer pasta that was changing its shape during cooking [67]. The simulations also showed good qualitative accordance with experiments.

Finally, the numerical method was tested for the case of high eigenstrain of the order of 1. For example, van Rees et al. [7] applied it to simulate the reconfiguration of a flat disc into a human face and of a flat square into a 3D map of a mountainous landscape. In this connection, Chen et al. [63] also modeled wrapping of a flat disc into a sphere. Although these simulations were not experimentally validated, they showed that the algorithm is robust even in the case of a high eigenstrain magnitude. Also, they showed that in this case the reconfiguration must be simulated in several steps with the eigenstrain gradually increasing from zero up to the value prescribed by the rest configuration (see Appendix of Ref. [7]). The final configuration computed on each step is thus used as the initial guess for the next step. This process "guides" the optimization algorithm and reduces the chance for it to get stuck in a local minimum. However, an optimal strategy for the eigenstrain interpolation was not determined. This problem is of secondary importance for the shot peen forming simulations due to a low eigenstrain magnitude, but a multi-step simulation can still increase accuracy of the forward problem resolution.

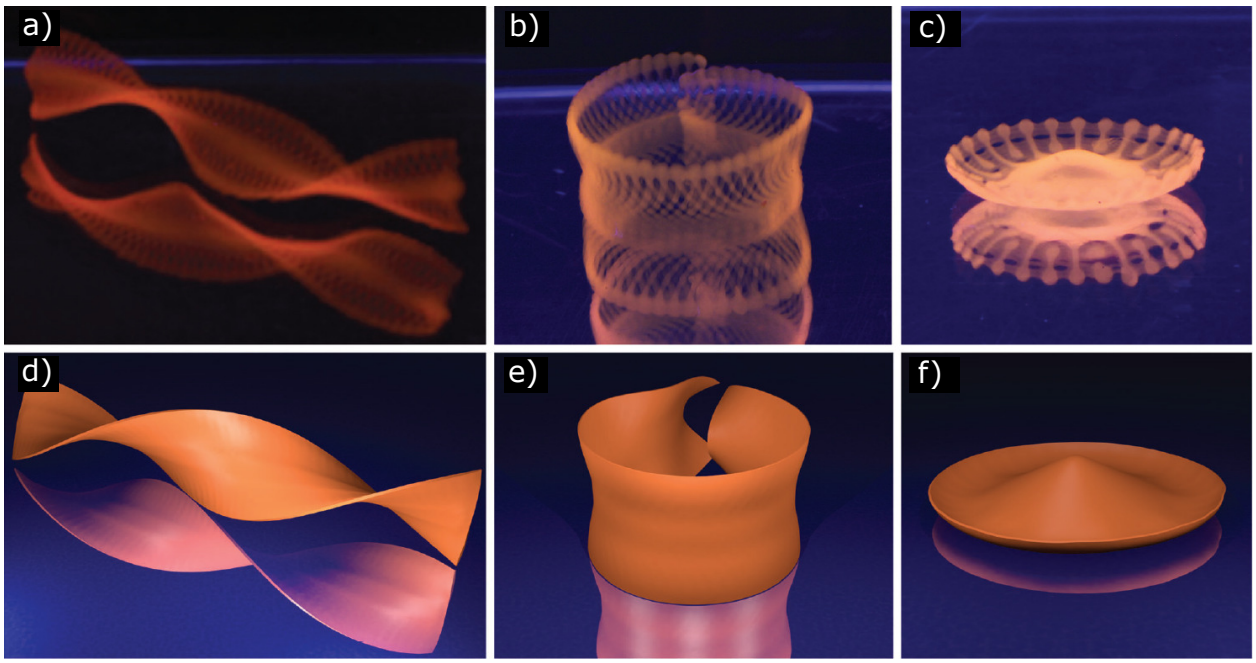

Figure 2.8 Comparison of the simulations and experiments conducted in the field of 4D printed non-Euclidean shells. The shells were printed flat and contained oriented filaments with embedded eigenstrain, which was released by humidity. The eigenstrain made the shells morph into a helicoid (a), a catenoid (b) or a sombrero (c). The numerical simulations (d-f) conducted using the geometric reformulation of the finite element method demonstrated good qualitative accordance with experiments. Adapted from Ref. [59] with permission from the Royal Society of Chemistry.

2.4 The inverse problem resolution

2.4.1 Early achievements

The first steps towards numerical resolution of the inverse problem were made by VanLuchene and Cramer [3]. The authors simulated the process using shell finite elements, and the effect of peening was reproduced with a stretching force and a bending moment applied to each element. The authors made an assumption that the deformed shapes can be represented as a linear combination of the deformed shapes induced by elementary loads. The relation between the loads and the actual peening parameters was calibrated experimentally [68]. With this, the inverse problem was resolved using a constrained least-squares algorithm, which optimized the sum of squared distances between the nodes of the target mesh and the current mesh with respect to the applied loads. Any experimental validation of the proposed approach is, however, not reported. The main drawback of this approach is the

representation of the peening-induced loads in terms of forces and moments. Thus, large peening-induced rotations provoke geometrically nonlinear behavior of the plate, which breaks the experimentally developed relations between these loads and the peening parameters [25].

The approach presented in Ref. [8] proposes a simpler optimization strategy. It involves a direct relation between the developed curvature and the peening parameters, thus bypassing the simulation stage. Hence, the curvature is numerically represented as a sum of two sine functions containing the peening parameters and six correction factors, that are calibrated experimentally. Although the method was validated for the case of freeform plates constituting a chair, the developed error was not quantified. A similar strategy was proposed by by Wang et al. [4], who developed a nonlinear relation between the arc height of a peened part and the exposure time. The local exposure times were thus optimized as a function of the local arc heights using a sequential quadratic programming algorithm [69]. Despite their simplicity, the proposed approaches are not suitable for the case of complex peening patterns, because the local curvatures are influenced by the curvatures in the neighboring zones. Moreover, the proposed relations may not hold for new geometries, thicknesses or materials.

Another strategy for the inverse problem resolution was reported by Essa et al. [70]. It consists in virtually flattening the target shape of the component by compressing it between two rigid surfaces and in observing the regions of compressive stress. According to the authors, the magnitude of the local compressive stress is proportional to the residual stress that must be developed by the peening treatment in the corresponding region. The residual stress, in turn, is related to the peening regimes through numerical impact simulation performed in advance. Same as the previously examined approaches, this approach loses its precision in the case of complex geometries and peening patterns. Unlike the eigenstrain, the residual stress evolves in a complex way during the reconfiguration of the component [25]. Consequently, the relation between the local peening-induced stresses and the final shape of the component has to be recalibrated for each particular peening case.

2.4.2 Eigenstrain-based approaches

In terms of the eigenstrain simulations, an efficient algorithm for the inverse problem resolution was proposed by Faucheux [25] and Miao et al. [5]. Similarly to Ref. [3] or [4], it is based on the numerical optimization of the sum of squared distances between the vertices of the target and the current meshes. However, the current mesh is computed on each iteration using the forward problem solver based on the shell finite element method, which is presented in Ref. [2]. The researchers use the trilayer formulation of the eigenstrain profile

(Eqn. 2.10), so the numerical optimizer locally adjusts ε_*^t and ε_*^b to fit the deflection to the target shape. Hence, the forward problem is solved on each iteration of the numerical optimization, that allows to take into account all nonlinearities appearing with complex patterns and large rotations. Moreover, if the relation between the eigenstrain and the peening parameters is established in advance, then the algorithm does not need to be recalibrated for new geometries or thicknesses. The numerical optimization is performed using the interior point algorithm [71], which provides more flexibility in the problem formulation than the constrained least squares algorithm applied in Ref. [3]. The proposed method was applied to compute optimal patterns for shaping the 2024-T3 aluminum plates having sizes of $305 \times 305 \times 4.9$ mm and $762 \times 762 \times 4.9$ mm into saddle and cylindrical shapes. One of the peening patterns prescribed in the saddle case is shown in Figure 2.9. The maximal local error in deflection between the target shapes and the simulated shapes provided by the computed patterns was less than 15.9% of the target shape deflection. The corresponding maximal error between the target shapes and the experimentally developed shapes was 13.8%.

Despite these promising results, the proposed method has two disadvantages. The first one is the formulation of the cost function for the optimization problem in terms of nodal positions of the target and current meshes in absolute coordinates. Such formulation raises a question of optimal positioning of the initial mesh with respect to the target mesh before beginning the optimization process. Indeed, different positions would make the optimization algorithm prescribe different nodal displacements and thus different local eigenstrains to the elements. At the same time, the eigenstrain is an internal load, so the final configuration of the body induced by such a load does not depend on its position in space. Consequently, the inverse problem solution must not depend on the mesh positions neither. This reasoning also applies to the methods described in Ref. [3], [8] and [4]. The second disadvantage is a long computation time. Indeed, the gradients needed for the numerical optimization

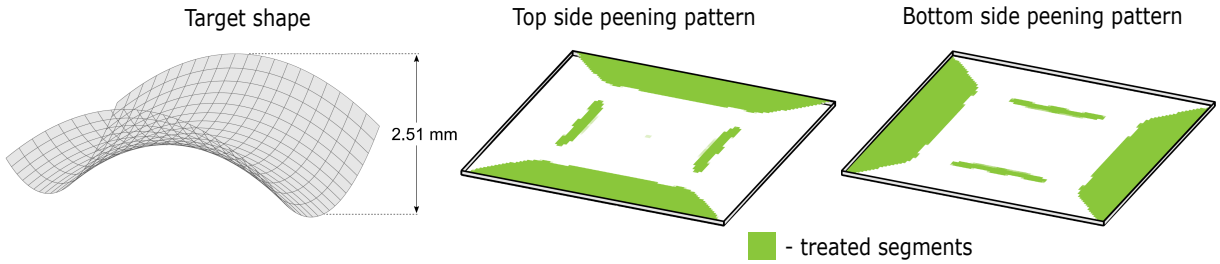

Figure 2.9 Example of a peening pattern prescribed to a $305 \times 305 \times 4.9$ mm 2024-T3 aluminum plate by the inverse problem solver presented in Ref. [5]. The target shape in this case is a saddle, and its deflection is amplified for the illustration purposes. Adapted from Ref. [5] with permission from Elsevier Science & Technology Journals.

are computed using the adjoint state method [72], as it is summarized in Ref. [73] in the context of a similar eigenstrain-based optimization problem. On each iteration, the adjoint method implies computing the inverse tangent stiffness matrix and its multiplication with the pseudo-load vector [73]. The number of variables in this problem equals twice the number of elements in the mesh because each element is assigned with a couple $(\varepsilon_*^t, \varepsilon_*^b)$ (Eqn. 2.10). Consequently, computation of the gradients takes a considerable amount of time with respect to the forward problem resolution, which is also performed on each iteration. In addition, the thicknesses of the active layers were fixed throughout the optimization process at a constant level: $h_*^b = h_*^t = const$. This formulation reduced the number of variables but did not allow to correctly compute the stretching induced by the treatment because the total eigenstrain Γ induced by the idealized eigenstrain profile in this case was not always realistic (Eqn. 2.7).

Another eigensrain-based method was proposed by Siguerdidjane et al. [6], who solved the inverse problem using an artificial neural network. The training phase for the neural network implies fixing the initial mesh, generating a large number of random peening patterns and solving the forward problem for each pattern. The random patterns are generated using the maze algorithm [74]. This algorithm traces random nozzle paths along the plate, so that the eigenstrain is then assigned to the elements that lie along the paths. The forward problem is solved using the shell finite element method in a similar fashion as in Ref. [2]. When the final shape is computed, the software measures the local Gaussian curvatures of each element of the final mesh. Multiple generated examples train the neural network to predict the peening pattern as a function of the local curvatures of the target shape, i.e., to solve the inverse problem. All in all, 50 000 examples were generated to train the network for a plate meshed with 1024 rectangular elements. The researchers considered treatment from one side only with one available peening regime, so they formulated the idealized eigenstrain profile as a step profile (Eqn. 2.9).

The neural network approach was validated numerically against 10 000 random target shapes, that were, in turn, induced by the maze patterns. The network correctly assigned the eigenstrain to 98.8% of elements in average. Along with the accuracy, the advantage of the proposed approach is the swiftness of the inverse problem resolution after the end of the training phase (45 microseconds). In fact, the same trained neural network may also be applied to rapidly solve the forward problem. Another improvement with respect to previously mentioned works is the description of the target shape in terms of Gaussian curvature. It is an intrinsic property of the surface that does not depend on its position in space.

On the other hand, the main limitation of the proposed method is a long phase of the training data generation, which took 20 hours in the described case. Furthermore, the neural network

is only trained for fixed initial geometry, mesh and peening regime. Hence, the training process must be reinitialized if one of these parameters changes. In addition, in practice the component is often treated from both sides with different peening regimes. Accounting for this fact would increase the number of training cases by several times, that would make the training phase computationally heavy. The same effect happens with the increase of the number of elements in the model.

2.4.3 Solution provided by the theory of non-Euclidean plates

An analytical relation that simplifies the inverse problem resolution in the case of bilayers subjected to eigenstrain was proposed by van Rees et al. [7]. It can be summarized as follows. Let us consider a target shape described by the two fundamental forms of its mid-surface $(\mathbf{a}_{tar}, \mathbf{b}_{tar})$. These fundamental forms are *a priori* compatible because they describe a real surface in Euclidean space without any fractures. Thus, the prescription of $\mathbf{a}_r = \mathbf{a}_{tar}$ and $\mathbf{b}_r = \mathbf{b}_{tar}$ as the rest configuration makes a flat plate deform into the target configuration because in this case $E_{ML} = 0$. The authors have shown that the rest configuration $(\mathbf{a}_r, \mathbf{b}_r)$ imposed on the mid-surface of the bilayer and the rest configuration $(\mathbf{a}_{r,t}\,, \mathbf{a}_{r,b})$ imposed on its two layers lead to the same final shapes when these rest configurations are related with Eqn. 2.19. Consequently, the bilayer adopts the target shape $(\mathbf{a}_{tar}, \mathbf{b}_{tar})$ if it is prescribed with the target configuration $(\mathbf{a}_{r,t}\,, \mathbf{a}_{r,b})$ defined as:

$$\begin{cases} \mathbf{a}_{r,t} = \mathbf{a}_{tar} - \dfrac{2h}{3}\mathbf{b}_{tar}, \\ \mathbf{a}_{r,b} = \mathbf{a}_{tar} + \dfrac{2h}{3}\mathbf{b}_{tar}. \end{cases} \tag{2.20}$$

This means that to numerically solve the inverse problem one needs to estimate the local fundamental forms $(\mathbf{a}_{tar}, \mathbf{b}_{tar})$ for each element of the mesh, to compute $(\mathbf{a}_{r,t}\,, \mathbf{a}_{r,b})$ using Eqn. 2.20 and to deduce the eigenstrain prescribed to the two layers by $(\mathbf{a}_{r,t}\,, \mathbf{a}_{r,b})$. This method is executed in one iteration and involves simple arithmetic operations with 2×2 sized matrices, so it takes a negligible amount of time with respect to the numerical optimization algorithms.

The simulations of reconfiguration of a flat disc into a human face and of a flat square into a mountainous landscape (see Section 2.3.3) were executed in order to test the proposed method. The authors quantified the error between the target shapes and the simulated shapes provided by the computed patterns in terms of the Hausdorff distance divided by the square root of the total area of the target shape. This value equaled 0.91% for the human face and 0.05% for the landscape. This error was mostly induced by the imperfection of the forward problem solver, that was also discussed in Refs. [59] and [63]. Namely, given that

the target shapes implied an high eigenstrain of the order of 100%, the numerical minimizer for the elastic energy may not have found the exact global minimum but converged to a close value.

Although the proposed method is precise and computationally efficient, it is not completely suitable for the shot peen forming case. Thus, in the general case, the rest configuration $(\mathbf{a}_{r,t}\,, \mathbf{a}_{r,b})$ computed using Eqn. 2.20 prescribes anisotropic and arbitrary oriented eigenstrain to each element of the mesh. The principal eigenstrain direction can also be different for the two layers. On the contrary, the eigenstrain anisotropy in the peen forming case is caused by the material properties or the prestress, so its direction cannot be *locally* controlled. Nevertheless, the rest configuration formulated in this way may provide an initial guess for an iterative algorithm for the inverse problem resolution in the peen forming case.

2.5 Segmentation of the peening pattern

The shot peen forming equipment imposes constraints on the peening pattern. For example, the maximal applied intensity is limited by the equipment capacity, so the algorithms for the inverse problem resolution presented in Section 2.4 constrain the optimized variables in an admissible range [3–5]. Furthermore, the peening equipment operates with fixed peening regimes and does not admit their gradual variation over the treated component. Therefore, the pattern must be divided into segments treated uniformly with fixed peening regimes. Figure 2.10 illustrates the concept of the pattern segmentation. The necessity for the pattern segmentation is only accounted for in the eigenstrain-based inverse problem solvers [5, 6, 25]. Thus, in Refs. [5, 25] they make use of the Solid Isotropic Material with Penalization (SIMP) method, that had been originally developed in the field of topology optimization [73, 75]. It consists in limiting the overall area assigned with the eigenstrain. This forces the optimization engine to "activate" only the segments that must necessarily be treated and to assign the maximal available eigenstrain to these segments. This penalizes all intermediate eigenstrain values in the peening pattern. Consequently, this method strictly divides the pattern into treated and untreated segments, but it is only suitable for one available peening regime, which corresponds to the maximal eigenstrain value. The same effect is achieved with the help of the artificial neural network presented in Ref. [6]. The neural network computes the peening pattern as a set of local *probabilities*. The probability is assigned to every element and indicates if it must be treated or not. Thus, only the elements with high probabilities are actually assigned with the eigenstrain. In turn, the eigenstrain magnitude is fixed and corresponds to the magnitude used in the training examples. This method is suitable for multiple available regimes, but such condition considerably extends the training phase.

A particularity of the inverse problem solvers presented in Refs. [5, 6, 25] is that they operate with a single predefined eigenstrain magnitude, which is chosen manually. Hence, the solvers only optimize the eigenstrain distribution but keep the magnitude constant. However, manual adjustment of the eigenstrain magnitude may be challenging because it is the main factor determining the induced curvature. Thus, a different eigenstrain magnitude leads to a different eigenstrain distribution, and the effect of an increase or a decrease of the eigenstrain magnitude is not always predictable. Moreover, the curvature of the target shape may be variable, and a fixed eigenstrain magnitude does not provide enough flexibility to precisely shape all segments. Consequently, the use of one predefined eigenstrain magnitude reduces both efficiency and precision of the proposed solvers.

At the same time, the problem of segmentation of a peening pattern is similar to the problem of clustering of points in space [9, 76]. The purpose of clustering algorithms is to optimally split a set of points into groups, i.e., clusters, based on the coordinates of points. In terms of the peening pattern, this means splitting the mesh into segments based on the eigenstrain values assigned to each finite element. Moreover, the clustering algorithms are able to find an optimal centroid for each cluster, i.e., to find the mean point representing the given cluster. With regard to the peening pattern, this allows to find the mean eigenstrain in the given segment and to homogenize the eigenstrain in this segment. Each eigenstrain value corresponds to a peening regime, so a clustering algorithm is in fact able to prescribe an optimal peening regime for each segment. However, the segmentation of a peening pattern

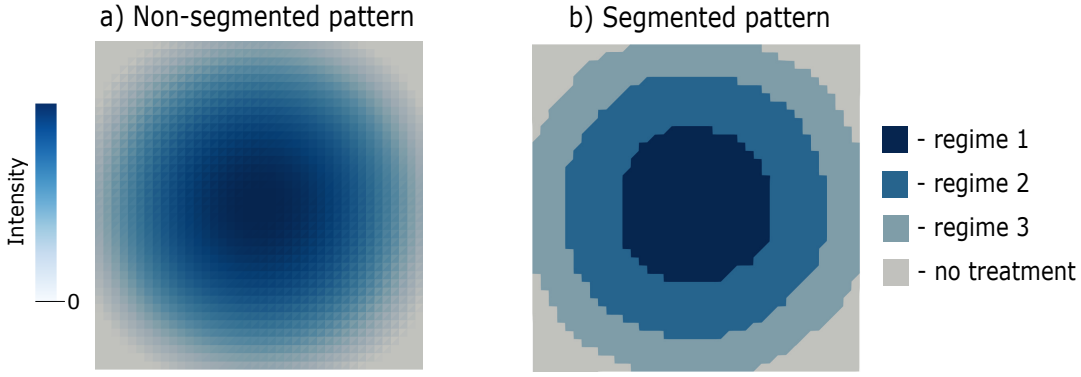

Figure 2.10 Examples of a non-segmented and of a segmented pattern. The non-segmented pattern (a) prescribes a gradual variation of the peening intensity over the plate, which is not reproducible with the peening equipment. The segmented pattern (b) leads to a similar target shape but consists of uniformly treated segments. This pattern is reproducible, and the effect of its application can be precisely simulated because each peening regime implied in the pattern can be characterized in advance. Both patterns are traced on a square plate meshed with triangular finite elements.

was never considered in terms of clustering so far.

The homogenization of eigenstrain inside the segments changes the peening pattern, so the final shape induced by the clustered pattern may be different from the one induced by the original pattern. Consequently, an optimal clustering algorithm for the peen forming case would only group the elements that are prescribed with close eigenstrains, so that the difference between the original and the segmented patterns is minimal. This can be accomplished using the `k`-means clustering algorithm [77], which groups points based on their proximity. More precisely, it iteratively finds optimal positions of the centroids and attributes the elements to the closest centroid in terms of the squared Euclidean distance. The particularity of the `k`-means algorithm is that it is not only simple to implement but also one of the most computationally efficient [76]. However, the result of clustering is dependent on the initial guess on the centroid positions [9], but this drawback can be mitigated by running the algorithm several times with different initial guesses and by picking the best result.

2.6 Automation of the peening process

The shot peening equipment propels the particles through a nozzle either using compressed air or using a rotating turbine that "throws" the particles. The nozzle is attached to the propelling equipment with an elastic pipe, so it can be displaced and redirected without the need to move the whole machine. The nozzle position is either controlled manually or automatically, meaning that the nozzle can be attached to a CNC-controlled robotic arm. In the latter case, the peening parameters can also be preprogrammed. Figure 2.11 presents an example of an automated shot peening machine. The automated peening is usually applied for repeatable operations that are intended to enhance the fatigue life of industrial parts [78]. This process implies uniform treatment of the component with a constant intensity. The forming operations, on the other hand, are often executed manually due to a higher complexity of the peening patterns [79]. This strategy is preferred by smaller companies that do not have the resources to develop a software for automated programming of the peening robot for each target shape. In the absence of such software, the resolution of the inverse problem is a trial and error process, and it is cheaper to execute it manually than to reprogram the robot for each trial. Even when an optimal pattern is developed, its translation into a program for the peening robot is a challenging problem that requires highly qualified staff. However, several commercial companies state to have automated their peen forming process, but the provided information is limited by commercial secrecy [10, 11, 80].

An extensive work towards automation of shot peen forming was done by the Aachen Shot Peening Centre (KSA). Thus, one of the first approaches for the shot peen forming automation

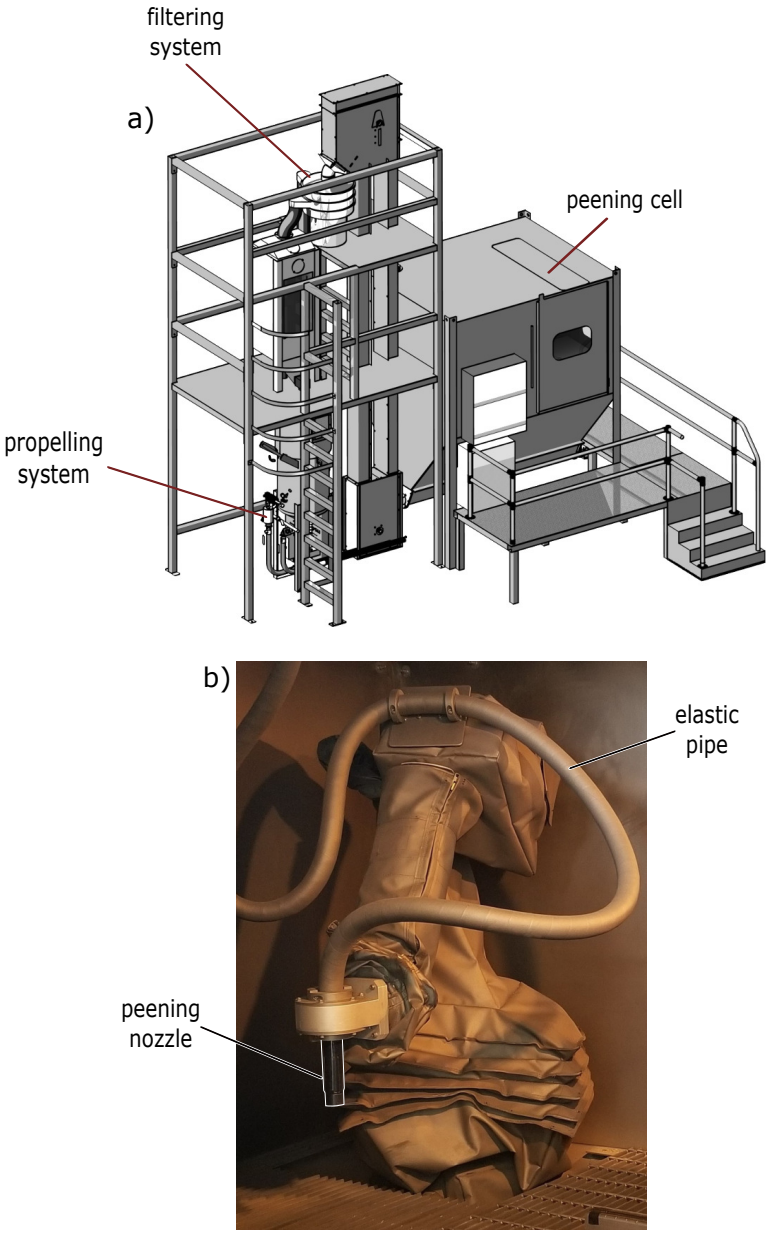

Figure 2.11 The automated shot peening equipment installed in the Aerospace Technology Center (CTA), Quebec, Canada. a) Schematic side of view of the peening machine. The treatment is executed inside a closed peening cell. The shot are propelled using compressed air, which is supplied externally. Used shot are continuously collected and passed to the filtering system that sorts out highly deformed shot using separator screens. The shot that pass through the screens are fed back to the propelling system. Adapted from Ref. [53] with permission from John Wiley & Sons - Books. b) The peening nozzle is attached to a robotic arm installed inside the peening cell. The nozzle is fed with shot by the propelling system through an elastic pipe. The robotic arm is covered with protective cloth preventing damage from reflected shot.

is described in Ref. [80]. It was designed to shape a square plate into a cylindrical patch. The approach involves consecutive shaping of the treated segments, so that the treatment of each segment begins only when the previous segment is shaped accurately. The plates are clamped in the middle of two edges, and the segments that are closer to the clamped part are shaped first. The quality of shaping of each segment is evaluated in real time using a coordinate measuring machine (CMM), so the proposed approach is essentially a feedback loop. A disadvantage of the CMM method in this case is the dependence of coordinates in each segment on the quality of shaping of the previously treated segments. Indeed, the segments treated at last are further from the clamping, so they are "hanging" attached to the previously shaped segments. Consequently, even if the current segment is shaped correctly but the previous segments were not, the CMM indicates an error in the current segment. With this, determination of the *local* error becomes a challenging task because the local error is the sum of errors developed in the previously shaped segments. Hence, the approach requires a precise shaping of each segment and a dense grid for the CMM, which is not always realizable in practice. The bending of the hanging part caused by the plate's own weight is not taken into account either, but it might be significant for large and thin industrial parts. In addition, the researchers compute the local peening intensity using direct relations between the developed curvature and the peening parameters without any global simulations. This strategy requires an extensive experimental work to determine how the local curvature is affected by the curvature in the adjacent segments. An important public contribution of this research consists in the strategy for choosing an offset between the parallel nozzle paths. Thus, it was shown that the particles in the shot stream are distributed according to the Gaussian function. This means that determination of the offset that provides uniform coverage boils down to determining an offset between adjacent Gaussian functions in the way that their superposition is a constant function.

According to a later publication made by the KSA [10], the drawbacks of the approach described in Ref. [80] were eliminated. The researchers state having a reliable software for the inverse problem resolution and a closed loop peen forming system, which is based on a CNC-controlled shot peening robot and an integrated 3D shape measuring gage. The system adjusts the peening pattern in real time based on the measurements provided by the gage. In turn, the pattern is formulated in terms of the peening parameters thus bypassing any transitory values, such as eigenstrain. Nevertheless, the report only describes the achievements without providing any technical information.

A simpler approach for peen forming automation is presented in the brochure issued by the Metal Improvement Company [11]. The company forms wing skins using a "feed through, gantry type machine". This type of machines projects particles from multiple nozzles that

are installed inside a peening cell. The treated plates are passed through the cell at a low speed using a conveyor system. In that way, the plates are uniformly treated with a constant intensity. This means that in order to apply a peening pattern, one needs to protect the segments that are not supposed to be treated with a *mask*, which absorbs the energy of shot. The mask fabrication is normally a manual process, although it is not described in the brochure. If the pattern involves multiple peening regimes, then the component must be passed through the peening cell multiple times, one time for each prescribed regime. The necessity for masking and the constant peening intensity inside the cell reduce the fabrication speed, but the authors do not provide details on the efficiency of their approach.

A similar semi-automated approach was also applied in Refs. [5, 25] to validate the inverse problem solver. In this work, the authors used a CNC-controlled peening nozzle, which was programmed for the uniform treatment of a component with a fixed peening regime. The untreated segments were masked manually using a butyl tape. The plates were free to deform during peening, and the authors state that this factor can be one of the sources of the developed error (see Section 2.2.3). Consequently, the authors recommend to hold the component flat during treatment.

All in all, the strategy proposed in Refs. [5, 25] allows to apply the pattern but does not use the full potential of the peening equipment. Indeed, a CNC-controlled robotic arm can be programmed to treat only the segments prescribed by the pattern, so that the need for masking is eliminated. However, manual programming of the robotic arm for each pattern is more time-consuming than fabrication of a mask. Consequently, a strategy for automated programming of the peening equipment is crucial for the full process automation. Moreover, the automated programming strategy must access the whole range of peening parameters because the peening pattern can include various peening regimes.

CHAPTER 3 OBJECTIVES AND RATIONALE

The deformation of a shot peened component is generated by the plastic strain induced by the treatment and the resulting geometrical incompatibility. Therefore, a natural way to simulate the peening-induced loads is to introduce the plastic strain in the model in the form of eigenstrain [24]. The eigenstrain in a uniformly treated segment is uniform along the surface and varies in the through-thickness direction. It is possible to idealize the through-thickness eigenstrain profile with no loss in the simulation accuracy [2]. This idealization reduces the number of variables in the model and accelerates the forward problem resolution. When the eigenstrain is defined, the forward problem can be solved either using the standard finite element method [29, 48], or using its geometric reformulation based on the theory of non-Euclidean plates [59, 63].

An important step for the peen forming simulation is the software calibration. If the loads are represented as eigenstrain, the calibration implies relating the eigenstrain to the actual peening regimes applied on the given material. Due to a vast variety of peening regimes and materials, this procedure must be simple and fast to be implemented in industry. The experimental identification of the eigenstrain using uniform treatment of small coupons fits these characteristics [29]. This procedure does not require the development of complex material models or the use of additional equipment. However, it must be followed by characterization of the eigenstrain anisotropy, and an efficient solution for this problem has to be determined.

The literature on peen forming lacks efficient methods for the inverse problem resolution. Two eigenstrain-based numerical tools were recently developed for this purpose. One of them uses gradient-based optimization [5] and another one is based on machine learning [6]. They both have an important disadvantage: a long computation time needed for densely meshed models. This problem stems from the fact that these tools are based on generic algorithms and do not take sufficient advantage of the thinness of the simulated components. At the same time, an efficient method for the inverse problem resolution was developed in the field of the theory of non-Euclidean plates [7]. Although the published method cannot be applied to peen forming directly, the theory of non-Euclidean plates provides powerful instruments for the development of an efficient inverse problem resolution algorithm for the peen forming case. The main advantage of this theory is that it is specifically developed to compute the deformation of plates induced by eigenstrain.

The inverse problem solution, i.e. the peening pattern, must be divided into segments with uniformly prescribed eigenstrain. The existing eigenstrain-based inverse problem solvers as-

sume having one fixed peening regime available [5,6], although one of the described methods can be extended to multiple predefined regimes [6]. However, none of the two methods proposes optimal peening regimes if they are not predefined. If the inverse problem solver prescribes a gradually varying eigenstrain along the surface, then the determination of the optimal peening regimes and the division of the pattern into segments is essentially a clustering problem.

There exist industrial shot peening robots with programmable nozzle trajectories and peening parameters [53]. At least one commercial company also states to have fully automated the peen forming process [10]. However, the existing literature provides a limited number of details on the peen forming automation. In other words, the literature lacks a workflow for developing various target shapes without any manual adjustment for each shape. For this reason, the only way for a company to automate the peen forming process nowadays is to develop its own in-house procedure.

Based on this synthesis of literature review and our identification of gaps, the following objectives were defined for this doctoral project:

1. **Develop a peen forming simulation software that solves the inverse problem using the theory of non-Euclidean plates.**

   Our simulation software includes the forward and the inverse problem solvers. Chapter 4 provides details on both solvers and on their numerical implementation. It also provides results of numerical validation of the inverse problem solver that involved randomly generated target shapes. Chapter 6 describes the experimental validation of both the forward and the inverse problem solvers. A particular attention in Chapter 6 is paid to the software calibration.

2. **Develop a pattern segmentation strategy that computes optimal peening regimes.**

   Our segmentation strategy presented in Chapter 5 divides any peening pattern generated by the inverse problem solver of the Objective 1 into segments that must be treated uniformly with fixed peening regimes. The peening regimes are numerically optimized during segmentation. The segmentation strategy relies on the k-means clustering algorithm. A strategy for grouping of the peening pattern in the case when the peening regimes are predefined is described in Chapter 4.

3. **Develop a complete workflow for automated shot peen forming.**

   Chapter 6 describes a complete workflow for the automated shot peen forming process, starting from a numerical model of the target shape and finishing with the ready-to-use component. We present both theoretical and practical details, so this Chapter is a complete guide for the process automation in industry. We also closely examine the strategy for evaluating the error between the target and the final shapes.

CHAPTER 4 ARTICLE 1: DETERMINATION OF OPTIMAL SHOT PEEN FORMING PATTERNS USING THE THEORY OF NON-EUCLIDEAN PLATES

Vladislav Sushitskii, Wim M. van Rees, Martin Lévesque, Frédérick P. Gosselin

Submitted to the Journal of Manufacturing Science and Engineering on the 18th of October 2021.

This article is dedicated to the core element of the automated peen forming workflow — the simulation software. Our forward problem solver is principally based on the solver developed by prof. Wim M. van Rees[1] *for the general case of bilayer shells subjected to eigenstrains. Its adaptation to shot peen forming was done during the internship at the Massachusetts Institute of Technology in March-April 2018. The inverse problem resolution algorithm, which is the main contribution of this paper, was conceived by the author of this thesis. Its numerical implementation was fused with the forward problem solver into a single simulation software. The software also includes the grouping algorithm that serves for the pattern segmentation. This version of the software was used during the experimental campaign presented in Chapter 6. Later, we developed the clustering and the filtering algorithms, which are presented in Chapter 5, and we included them in the software to enhance the pattern segmentation.*

Abstract

We show how a theoretical framework developed for modeling nonuniform growth can model the shot peen forming process. Shot peen forming consists in bombarding a metal panel with multiple millimeter-sized shot, that induce local bending of the panel. When applied to different areas of the panel, peen forming generates compound curvature profiles starting from a flat state. We present a theoretical approach and its practical realization for simulating peen forming numerically. To achieve this, we represent the panel undergoing peen forming as a bilayer plate, and we apply a geometry-based theory of non-Euclidean plates to describe its reconfiguration. Our programming code based on this approach solves two types of problems: it simulates the effect of a predefined treatment (the forward problem) and it finds the optimal treatment to achieve a predefined target shape (the inverse problem). Both problems admit using multiple peening regimes simultaneously. The algorithm was tested numerically on 200 randomly generated test cases.

[1]Assistant Professor, Van Rees Lab, Department of Mechanical Engineering, Massachusetts Institute of Technology

4.1 Introduction

Shot peen forming is a cost-effective technology for shaping large metal plates, such as airplane wing skins, without dies. It consists in bombarding the surface of a component with a large number of millimeter sized shot made of steel, glass or ceramic. The velocity of a shot is sufficiently high to plastically deform the upper layer of the plate upon impact and to stretch the plate locally. This effect causes local bending of thin components and leads to a convex curvature on the peened side [1]. Repeated impacts also induce a field of compressive residual stress, that can improve fatigue life [20].

When developping a shot peen forming process, one is faced with two types of problems as schematized in Fig. 4.1: *the forward problem* and *the inverse problem* [25]. The forward problem is formulated with the following question: which shape will the component adopt if it is peened according to a given pattern? The inverse problem denotes the following: given an initial shape of the component and the target shape, how should one peen the component to make it deform into the target shape? A numerical solver for both problems is necessary to optimize the forming process. Thus, without numerical resolution of the inverse problem, the design of a peen forming procedure for each new component is a craft trial-and-error process plagued with risk and uncertainty. It lasts up to several months and implies many scrapped parts. On the other hand, numerical resolution of the forward problem is necessary to check the quality of the inverse problem resolution and to simulate the effect of additional treatments.

A straightforward simulation of individual peening impact, such as the one conducted in [22], is precise but computationally expensive. For this reason, simplified multiscale simulation approaches, such as the *eigenstrain* approach, were developed. It implies formulating the applied peening loads in terms non-elastic strains imposed on the component [28] [23]. The eigenstrain approach represents the treated plate as a thin bilayer where each layer undergoes a nonuniform plastic in-plane swelling or shrinking. The forward problem in this formulation can be numerically solved using shell finite element models [2] [6]. In addition, such models allow to implement an inverse problem resolution algorithm based on the topology optimization methods [73] [5]. A similar inverse problem resolution approach was also applied in [81] in the context of the *laser* peen forming process. Given that this method is based on numerical optimization, its speed decreases with the number of elements in the model. On the other hand, the inverse problem can be solved using an artificial neural network [6]. The neural network provides near-perfect accuracy and fast calculation on-line. However, it requires the generation of a large finite-element forward problem solution database and a long training phase for each new plate geometry.

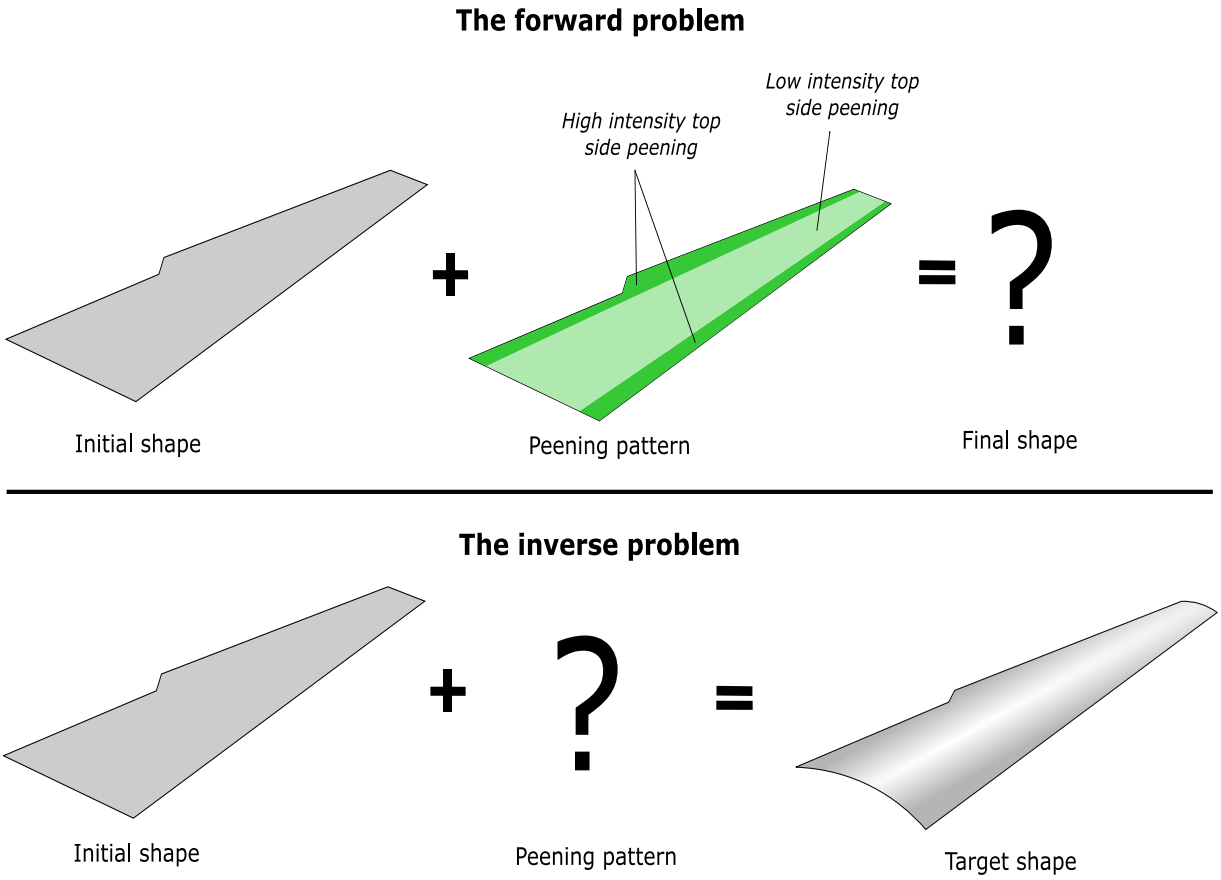

Figure 4.1 The two problems of the shot peen forming simulation. a) The forward problem consists in the determination of the final shape of the component given its initial shape and the peening pattern. b) The inverse problem consists in determination of the optimal peening pattern given the initial shape of the component and its target shape.

To overcome these issues during the inverse problem resolution, we turned to the theory of non-Euclidean plates [45] [47]. This theoretical framework lies at the intersection of mechanics and differential geometry. It precisely describes distortion of multilayer plates induced by the prescription of a nonuniform non-elastic strain. Prescription of such strain makes the surface metric non-Euclidean so that it does not satisfy the compatibility conditions of the Euclidean space, which gives name to the theory [47]. Experiments conducted in the field of 4D printing with elastic polymer sheets have proven the accuracy of the theory of non-Euclidean plates in numerical forward problem resolution. Thus, this theory precisely predicted curvature of spherical, cylindrical and saddle shapes grown out of a flat state by induction of a nonuniform plastic strain [57] [46]. Numerical simulation of growth for these three cases also showed good accordance with analytical shapes [63]. Moreover, this theory predicted the form of more complex shapes, such as helicoid, catenoid or an orchid flower grown out of a flat state

using polymer bilayers with oriented filaments [59]. The theory of non-Euclidean plates also provides instruments for an efficient inverse problem resolution. Such algorithms for the case of polymers with oriented local growth are reported in [59] and [82].

In this paper we propose to use the eigenstrain approach to represent the treated plate as a bilayer, and we resort to the theory of non-Euclidean plates to calculate the resulting distortion of the bilayer. To numerically solve the inverse problem, we created an iterative algorithm that implies resolving the forward problem on each iteration. The algorithm adjusts the peening pattern based on the discrepancy between the current shape, i.e., the shape obtained with the current pattern, and the target shape. The adjustment is done on a local scale using simple arithmetic operations, so the computation speed on this stage stays constant with an increasing number of variables. This stage does not require any preliminary training phase either. The algorithm constitutes a general approach for the inverse problem resolution in case of bilayers subjected to isotropic, i.e. non-oriented, local growth, such as those examined in [46]. When the iterative adjustment is finished, we group the peening pattern to make it practically applicable. In other words, we divide the pattern into zones treated with constant peening regimes. The number of available peening regimes and their intensities are pre-determined based on the practical constraints.

We start this paper with the theoretical background section. First, we examine the eigenstrain approach that relates peen formed plates and swelling non-Euclidean bilayers. Next, we move to the theory of non-Euclidean plates, namely to the geometrical shape description and the forward problem resolution method that it implies. In this section we also formulate the inverse problem in terms of the theory of non-Euclidean plates. We then pass to the methodology section by presenting our inverse problem resolution algorithm and an approach for its numerical implementation. The grouping and validation strategies are presented in the same section. The results of the validation campaign are presented subsequently, and finally the advantages and limitations of our approach are discussed.

4.2 Theoretical background

4.2.1 The eigenstrain approach and strain decomposition

The term eigenstrains denotes all non-elastic strains arising in the material, such as plastic, thermal or piezoelectric strains [28]. The only type of eigenstrain generated by peen forming is the plastic strain. Indeed, numerous overlapping impact indentations plastically stretch the outer layer of material, and the rest of the material responds to this newly-introduced eigenstrain with the emergence of stress. In order to conserve its integrity and to balance the

stress, the plate deforms elastically. In case of small strains the residual strain tensor $\boldsymbol{\varepsilon}^{res}$ is additively decomposed into the eigenstrain part $\boldsymbol{\varepsilon}$ and the elastic part $\boldsymbol{\varepsilon}^{el}$ ([7], Appendix):

$$\boldsymbol{\varepsilon}^{res} = \boldsymbol{\varepsilon} + \boldsymbol{\varepsilon}^{el}. \tag{4.1}$$

This relation holds for peen forming because the process deals with thin plates and the peening-induced strains are small [22]. The elastic strain may affect the whole plate, while the eigenstrain is present only in the stretched outer layer. The thickness of the plastically deformed outer layer varies depending on the peening parameters and the treated material. With the eigenstrain approach, the resolution of the forward problem for shot peen forming consists in introducing the eigenstrain over the whole shot peened area and determining the elastic springback.

The through-thickness eigenstrain profile

We endow the mid-surface of the plate with two Lagrangian coordinates (x, y), and we assign a Lagrangian coordinate z in the through-thickness direction. The Lagrangian coordinates follow the plate as it deforms. We assume that the material is plastically isotropic, so the eigenstrains are the same in all in-plane directions: $\varepsilon_{xx}(x, y, z) = \varepsilon_{yy}(x, y, z)$. Also, $\varepsilon_{zz} = -2\varepsilon_{xx}$ due to plastic incompressibility. For a small area around a point (x_0, y_0) on the mid-surface, the through-thickness eigenstrain profile $\varepsilon_{xx}(x_0, y_0, z) = \varepsilon_{yy}(x_0, y_0, z)$ can be measured directly using the X-ray diffraction method [21]. Otherwise, it can be deduced from the residual stress profile, which is determined with such methods as hole drilling [43], layer-removal [43] [27] or the two cut compliance method [83]. In this case, the through-thickness eigenstrain profile is reconstructed numerically from the measured residual stress profile [2] [44].

Mechanically, the introduction of the eigenstrain can be modeled as slicing the plate into thin layers, stretching the outer layers separately following the eigenstrain profile, and then gluing everything back. To numerically simplify the problem, we virtually idealize the eigenstrain profile by assuming that the plate consists of two layers of equal thickness that can separately undergo nonuniform in-plane swelling or shrinking. The idealized eigenstrain profile leads to the same deformed shape as the real one. Figure 4.2 illustrates both profiles induced on a uniformly treated plate. Mathematically, the idealization procedure consists in finding the local eigenstrain $\varepsilon^t(x, y)$ to be introduced in the top layer and the local eigenstrain $\varepsilon^b(x, y)$ to be introduced in the bottom layer. It is done by equating the total eigenstrain $\Gamma(x, y)$ and the first eigenstrain moment $\Gamma_1(x, y)$ induced locally by the real eigenstrain profile to those

induced locally by $\varepsilon^t(x,y)$ and $\varepsilon^b(x,y)$ [2]. In the general case, Γ and Γ_1 are defined as:

$$\Gamma(x,y) = \int_{-h/2}^{h/2} \varepsilon_{xx}(x,y,z)dz, \tag{4.2}$$

$$\Gamma_1(x,y) = \int_{-h/2}^{h/2} \varepsilon_{xx}(x,y,z)zdz, \tag{4.3}$$

where h stands for the plate thickness. For the idealized bilayer profile Γ and Γ_1 are expressed as:

$$\Gamma(x,y) = \frac{h}{2}\left(\varepsilon^t(x,y) + \varepsilon^b(x,y)\right), \tag{4.4}$$

$$\Gamma_1(x,y) = \frac{h^2}{8}\left(\varepsilon^t(x,y) - \varepsilon^b(x,y)\right). \tag{4.5}$$

The idealized eigenstrain is positive on the side that undergoes the peening treatment and is negative on the other side. It should be noted that although ε^t and ε^b give rise to the same in-plane extension and curvature as the real eigenstrain profile, the idealized one generates a different residual stress profile.

4.2.2 The theory of non-Euclidean plates applied to the modeling of shot peen forming

The theory of non-Euclidean plates allows to numerically determine the elastic response of thin bodies to an applied nonuniform non-elastic strain, e.g., biological growth or eigenstrain. The introduction of such strain leads to the emergence of stresses, which are *entirely* eliminated only if the plate adopts a so-called *rest configuration.* The rest configuration implies $\boldsymbol{\varepsilon}^{res} = \boldsymbol{\varepsilon}$, so it is generally not realizable without the loss of integrity. Instead of the rest configuration the plate adopts an integral *final configuration (final shape)*, which is still residually stressed. The theory of non-Euclidean plates relates the rest and the final configurations through elastic energy and uses tools from differential geometry to describe the shape of plates. We adopted the approach described in [7] for its numerical implementation.

Geometrical shape description

In the framework of the theory of non-Euclidean plates, the plate shape is associated with the shape of its mid-surface [7]. We denote by U the domain of the plane containing the

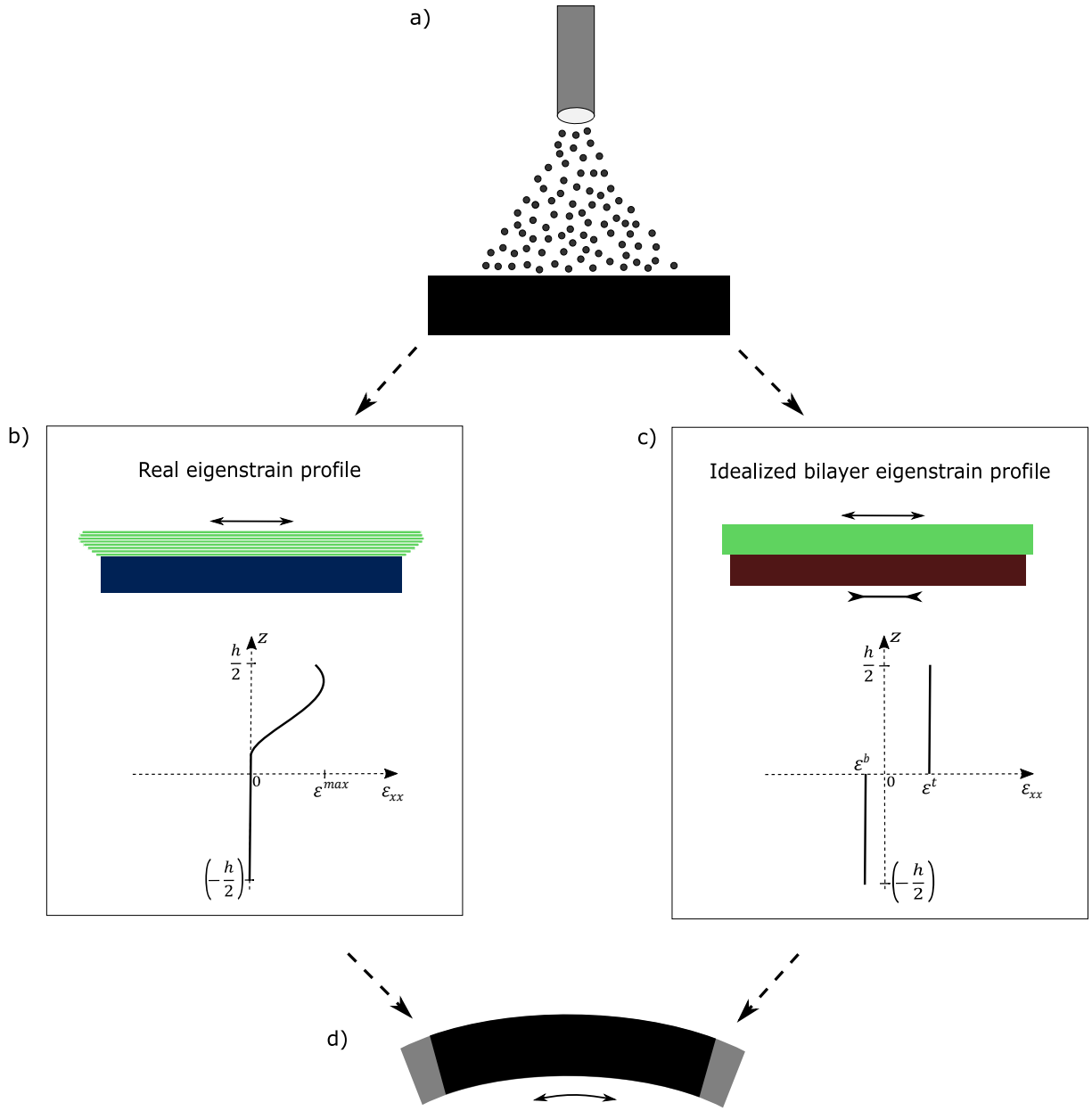

Figure 4.2 The real and idealized eigenstrain profiles induced by uniform shot peening of a plate. We denote the plate thickness as h. The coordinate z goes along the thickness and measures form the midplane. a) The plate undergoes uniform shot peening (side view). b) The through-thickness eigenstrain profile is nonuniform along z, and its peak value ε^{max} is close to the surface [2]. Its effect is accurately simulated by virtually dividing the plate into thin layers and imposing different eigenstrain to each of the layers thus reproducing the profile shape. c) We idealize the eigenstrain profile and represent the plate as a bilayer consisting of two layers of the thickness $h/2$. The eigenstrain ε^t and ε^b assigned to each of the layers is derived from the real eigenstrain profile by equating the total eigenstrain and the first eigenstrain moment. d) Both real and idealized eigenstrain profiles lead to the same deformed shape, which is bent and stretched with respect to the initial state.

coordinates (x, y) that parametrize the mid-surface: $(x, y) \in U \subset \mathbb{R}^2$. The position of each point of the mid-surface in a 3D space is defined by the mapping $\vec{m} : U \to \mathbb{R}^3$. We adopt the Kirchhoff-Love assumptions, so the position $\vec{r}$ of a point belonging to the plate is expressed as:

$$\vec{r}\,(x, y, z) = \vec{m}\,(x, y) + z\vec{n}\,(x, y)\,, \tag{4.6}$$

where $\vec{n}$ is the unit normal vector. The mid-surface shape is described by *the first and the second fundamental forms*, that are binary quadratic forms associated with a symmetric 2×2 matrix. Both fundamental forms are local quantities varying smoothly along the surface. The first fundamental form describes changes in the length of curves and areas of regions on the surface. In other words, it describes the local stretching of the surface. The 2×2 matrix containing coefficients of the first fundamental form $\mathbf{a}(x, y)$ is computed as:

$$\mathbf{a}\,(x, y) = \begin{bmatrix} \partial_x\vec{m} \cdot \partial_x\vec{m} & \partial_x\vec{m} \cdot \partial_y\vec{m} \\ \partial_y\vec{m} \cdot \partial_x\vec{m} & \partial_y\vec{m} \cdot \partial_y\vec{m} \end{bmatrix}, \tag{4.7}$$

where $\partial_x\vec{m} = \partial\vec{m}/\partial x$ and $\partial_y\vec{m} = \partial\vec{m}/\partial y$ are two vectors tangent to the mid-surface at the point $\vec{m}(x, y)$. If a certain area of the surface does not undergo any stretching, the first fundamental form in this area is represented by the identity matrix $\mathbf{I}$.

Together with the first fundamental form, the second fundamental form determines local curvatures on a surface. The matrix containing its coefficients $\mathbf{b}(x, y)$ is computed as:

$$\mathbf{b}(x, y) = \begin{bmatrix} \partial_{xx}\vec{m} \cdot \vec{n} & \partial_{xy}\vec{m} \cdot \vec{n} \\ \partial_{yx}\vec{m} \cdot \vec{n} & \partial_{yy}\vec{m} \cdot \vec{n} \end{bmatrix} = - \begin{bmatrix} \partial_x\vec{m} \cdot \partial_x\vec{n} & \partial_x\vec{m} \cdot \partial_y\vec{n} \\ \partial_y\vec{m} \cdot \partial_x\vec{n} & \partial_y\vec{m} \cdot \partial_y\vec{n} \end{bmatrix}, \tag{4.8}$$

where $\partial_{xx}\vec{m}$, $\partial_{xy}\vec{m}$ and $\partial_{yy}\vec{m}$ denote the second derivatives of $\vec{m}(x, y)$. If a surface is locally flat, its second fundamental form at this area is described by the zero matrix, because all second derivatives of the mapping $\vec{m}$ are orthogonal to the normal vector $\vec{n}$.

The two fundamental forms define a unique surface up to solid body motions. The surface is integral if its fundamental forms are *compatible*, i.e., if they satisfy three partial differential equations called the Gauss–Peterson–Mainardi–Codazzi (GPMC) equations [55]. Hence, the two *final* fundamental forms of a non-Euclidean plate are compatible, while its *rest* fundamental forms are not. This phenomenon illustrated in Fig. 4.3 is also called *geometric incompatibility.*

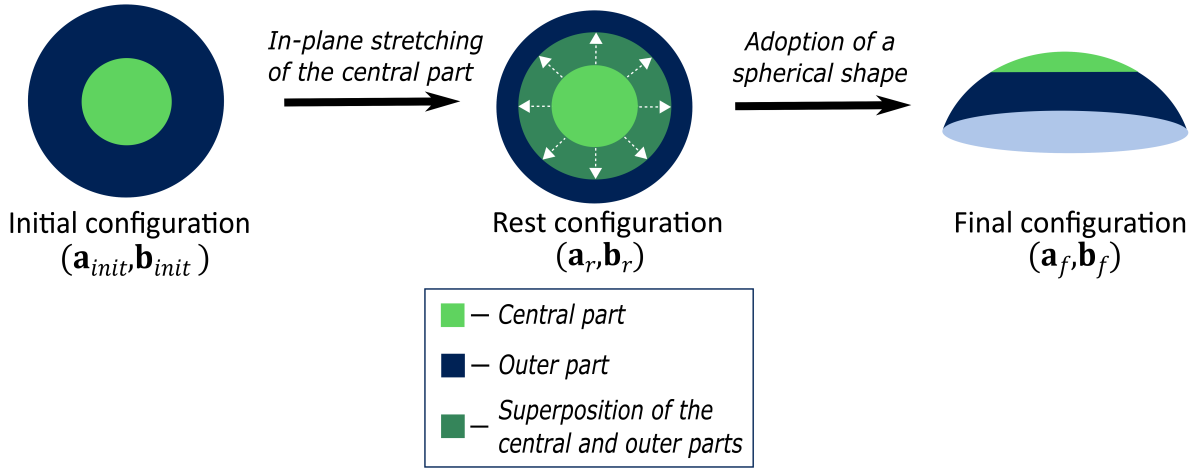

Figure 4.3 An example of geometric incompatibility inspired by [57]. The initial configuration described by the fundamental forms $(\mathbf{a}_{init}, \mathbf{b}_{init})$ is a flat unstretched disc, so that $\mathbf{a}_{init}$ is the identity matrix and $\mathbf{b}_{init}$ is the zero matrix along the whole surface. The rest configuration $(\mathbf{a}_r, \mathbf{b}_r)$ prescribes isotropic in-plane stretching of the central part (green) while conserving the flat shape, so that $\mathbf{a}_r \neq \mathbf{a}_{init}$ and $\mathbf{b}_r = \mathbf{b}_{init}$. In the general case, $\mathbf{b}_r$ can be different from $\mathbf{b}_{init}$ as well. The rest fundamental forms are incompatible, because the adoption of the rest configuration means superposition of the inner (green) part and the outer (blue) part of the disc and thus loss of integrity. In order to conserve its integrity, the disc adopts a curved final configuration $(\mathbf{a}_f, \mathbf{b}_f)$ described by compatible fundamental forms. However, the disc stays residually stressed in its final configuration.

Elastic energy and the forward problem resolution

In the terms of the theory of non-Euclidean plates, the forward problem consists in determining the final configuration as a function of the rest configuration. First, we consider the simple case of an initially flat *monolayer* plate, which is subjected to a rest configuration described by the incompatible fundamental forms $\mathbf{a}_r$ and $\mathbf{b}_r$. The rest and final configurations of such a plate are related through the elastic energy functional. We denote the final fundamental forms as $\mathbf{a}_f$ and $\mathbf{b}_f$ and we express the elastic energy E_{ML} of an integral monolayer plate as [7]:

$$E_{ML} = \frac{1}{2}\int_U \left[\frac{h}{4}\left\|\mathbf{a}_r^{-1}\mathbf{a}_f - \mathbf{I}\right\|_e^2 + \frac{h^3}{12}\left\|\mathbf{a}_r^{-1}(\mathbf{b}_f - \mathbf{b}_r)\right\|_e^2\right]\sqrt{\det \mathbf{a}_r}\; dxdy. \tag{4.9}$$

In this expression we have introduced the elastic norm $\|\mathbf{A}\|_e^2 = \alpha Tr^2(\mathbf{A}) + 2\beta Tr(\mathbf{A}^2)$ with coefficients $\alpha = Y\nu/(1-\nu^2)$ and $\beta = Y/(2+2\nu)$. Here, Y is the Young modulus and ν is the Poisson's ratio.

A zero elastic energy means that the final configuration perfectly coincides with the rest one. If the rest fundamental forms are incompatible, an integral plate adopts an equilibrated final configuration that minimizes the elastic energy under constraints that $\mathbf{a}_f$ and $\mathbf{b}_f$ be compatible. The first term of the elastic energy functional represents the stretching energy, and the second term defines the bending energy. If the plate is thin, the bending term is small in comparison to the stretching one, so the plate stretches as prescribed by the rest configuration but adopts a different curvature. In other words, $\mathbf{a}_f$ in this case is close to $\mathbf{a}_r$, but $\mathbf{b}_f$ can be largely different from $\mathbf{b}_r$. On the contrary, a thick plate adopts the curvature prescribed by the rest configuration but stretches in a different way, so that $\mathbf{b}_f$ is close to $\mathbf{b}_r$ [45].

Now let us consider a *bilayer* plate, where each layer is of thickness $h/2$. We suppose that the plate is initially flat, and that its layers exhibit nonuniform in-plane swelling or shrinking. The eigenstrain introduced in each layer may vary along the surface, but it is constant along the layer thickness. Locally, the eigenstrain is different for each layer, so that the rest first fundamental forms of each layer ($\mathbf{a}_{r,t}$ and $\mathbf{a}_{r,b}$) are different. Matrices $\mathbf{a}_{r,t}$ and $\mathbf{a}_{r,b}$ contain information on the principal eigenstrain direction and magnitude on the top and bottom layers respectively. Essentially, the $\mathbf{a}_{r,t}$ and $\mathbf{a}_{r,b}$ describe stretching that the layers would adopt if they were not attached together. By assuming that each layer expands uniformly across its thickness, all terms of the rest second fundamental form of each layer are zero along the whole surface:

$$\mathbf{b}_{r,t} = \mathbf{b}_{r,b} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}. \tag{4.10}$$

Accordingly, $\mathbf{a}_{r,t}$ and $\mathbf{a}_{r,b}$ fully describe the rest configuration. The forward problem consists in finding $\mathbf{a}_f$ and $\mathbf{b}_f$ that describe the shape of the integral plate mid-surface after reconfiguration. The reconfiguration process for the bilayer case is presented in Fig. 4.4. Following [7], we express the elastic energy of the bilayer plate as the sum of the elastic energies of two monolayers of thickness $h/2$. After integration over the total plate thickness we obtain:

$$\begin{aligned} E_{BL} = &\frac{1}{2}\int_U \left[\frac{h}{8}\left\|\mathbf{a}_{r,b}^{-1}\mathbf{a}_f - \mathbf{I}\right\|_e^2 + \frac{h^3}{24}\left\|\mathbf{a}_{r,b}^{-1}\mathbf{b}_f\right\|_e^2 + \frac{h^2}{8}\left\langle\left(\mathbf{a}_{r,b}^{-1}\mathbf{a}_f - \mathbf{I}\right), \mathbf{a}_{r,b}^{-1}\mathbf{b}_f\right\rangle_e\right]\sqrt{\det \mathbf{a}_{r,b}}\; dxdy \\ &+\frac{1}{2}\int_U \left[\frac{h}{8}\left\|\mathbf{a}_{r,t}^{-1}\mathbf{a}_f - \mathbf{I}\right\|_e^2 + \frac{h^3}{24}\left\|\mathbf{a}_{r,t}^{-1}\mathbf{b}_f\right\|_e^2 - \frac{h^2}{8}\left\langle\left(\mathbf{a}_{r,t}^{-1}\mathbf{a}_f - \mathbf{I}\right), \mathbf{a}_{r,t}^{-1}\mathbf{b}_f\right\rangle_e\right]\sqrt{\det \mathbf{a}_{r,t}}\; dxdy. \end{aligned} \tag{4.11}$$

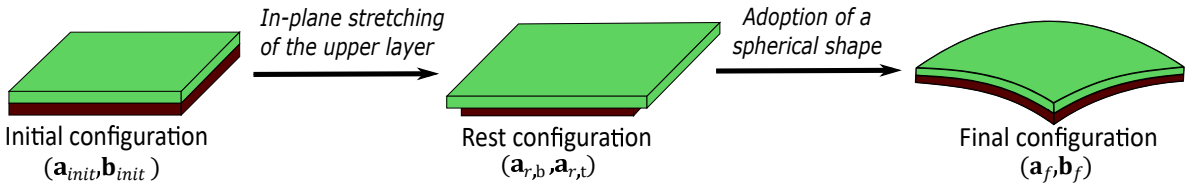

Figure 4.4 An example of geometric incompatibility in a bilayer plate inspired by [7]. The initial configuration described by the fundamental forms $(\mathbf{a}_{init}, \mathbf{b}_{init})$ is a flat unstretched rectangular plate, so that $\mathbf{a}_{init}$ is the identity matrix and $\mathbf{b}_{init}$ is the zero matrix along the whole surface. The rest configuration $(\mathbf{a}_{r,t}, \mathbf{a}_{r,b})$ prescribes isotropic in-plane stretching of the top layer and shrinking of the bottom layer while conserving the flat shape, so that $\mathbf{a}_{r,t} \neq \mathbf{a}_{init}$, $\mathbf{a}_{r,b} \neq \mathbf{a}_{init}$, $\mathbf{b}_{r,t} = \mathbf{b}_{init}$ and $\mathbf{b}_{r,b} = \mathbf{b}_{init}$. Since peen forming causes in-plane eigenstrain we do not consider cases when $\mathbf{b}_{r,t} \neq \mathbf{0}$ and $\mathbf{b}_{r,b} \neq \mathbf{0}$ for our simulations, so that the rest configuration is entirely described by $(\mathbf{a}_{r,t}, \mathbf{a}_{r,b})$. Adoption of the rest configuration means dissection of the plate in two layers. Instead, the plate adopts an integral but residually stressed final configuration $(\mathbf{a}_f, \mathbf{b}_f)$ described by compatible fundamental forms.

The elastic energy inner product $\langle \cdot, \cdot \rangle_e$ introduced in this context defines the following operation: $\langle \mathbf{A}, \mathbf{B} \rangle_e = \alpha Tr(\mathbf{A}) Tr(\mathbf{B}) + 2\beta Tr(\mathbf{AB})$. Similarly to the monolayer case, the plate adopts a curved final configuration $(\mathbf{a}_f, \mathbf{b}_f)$ that minimizes the elastic energy. Generally, the final configuration is not unique, and moreover, the plate can get stuck in a configuration corresponding to a local energetic minimum on its way to the global minimum.

For a bilayer subjected to any rest configuration $(\mathbf{a}_{r,b}, \mathbf{a}_{r,t})$, there exists an equivalent monolayer that morphs into the same final configuration $(\mathbf{a}_f, \mathbf{b}_f)$ after being subjected to a rest configuration $(\mathbf{a}_r, \mathbf{b}_r)$. The relation between $(\mathbf{a}_{r,b}, \mathbf{a}_{r,t})$ and $(\mathbf{a}_r, \mathbf{b}_r)$ is derived by equating the monolayer energy (Eqn. 4.9) and the bilayer energy (Eqn. 4.11) and is expressed as [7]:

$$\begin{cases} \mathbf{a}_r = \dfrac{1}{2} (\mathbf{a}_{r,b} + \mathbf{a}_{r,t}) ; \\ \mathbf{b}_r = \dfrac{3}{4h} (\mathbf{a}_{r,b} - \mathbf{a}_{r,t}) . \end{cases} \tag{4.12}$$

Inversely,

$$\begin{cases} \mathbf{a}_{r,t} = \mathbf{a}_r - \dfrac{2h}{3} \mathbf{b}_r ; \\ \mathbf{a}_{r,b} = \mathbf{a}_r + \dfrac{2h}{3} \mathbf{b}_r . \end{cases} \tag{4.13}$$

Here, the monolayer and bilayer plates are supposed to have the same initial geometry and the same total thickness h.

The inverse problem resolution

The inverse problem consists in determining the rest configuration that leads to a target configuration due to the elastic material response. In the bilayer case, it means finding the rest first fundamental forms of the bottom and top layers $\mathbf{a}_{r,b}$ and $\mathbf{a}_{r,t}$, respectively as a function of the target shape described by $\mathbf{a}_{tar}$ and $\mathbf{b}_{tar}$. As each fundamental form is represented by a symmetric 2×2 matrix, the target configuration is locally defined by six scalar variables. At the same time, the rest configuration also has six degrees of freedom, so the problem admits an analytical solution provided by Eqn. (4.13). Moreover, the final configuration obtained with this solution is stress-free [7]. A practical application of the analytically calculated solution means inducing local *orthotropic* eigenstrain, so that $\varepsilon_{11}(x, y, z) \neq \varepsilon_{22}(x, y, z)$, where ε_{11} and ε_{22} are the local principal eigenstrains. In other words, one has to locally control the principal strain directions and the strain magnitude along both directions, and this – at each of the two layers separately. Such local control over the direction of expansions is generally not possible with shot peen forming, because here we assume that shot peening induces local *isotropic* in-plane strain. Thus, at each point we only control two degrees of freedom: $\varepsilon^t(x, y)$ and $\varepsilon^b(x, y)$. Hence, we have control over fewer degrees of freedom than input variables. In this case, a solution leading exactly to the target shape may not exist, so we can only numerically optimize $\mathbf{a}_{r,t}$ and $\mathbf{a}_{r,b}$. The uniqueness of solution is not guaranteed either.

4.3 Methodology

Our method for the inverse problem resolution consists in the iterative correction of the peening pattern on a local scale until convergence is reached within a tolerance and subsequent grouping of the pattern. The grouping algorithm divides the pattern in zones treated with predefined peening regimes.

4.3.1 Iterative adjustment of the peening pattern

The iterative method idea is to adjust the rest configuration by comparing the local stretching and curvature of the current shape with the stretching and curvature of the target shape. The current shape is computed at each iteration through a numerical resolution of the forward problem. For the initial guess, we use analytical expressions to approximately define an appropriate rest configuration.

The initial guess

We characterize the target configuration in terms of fundamental forms $(\mathbf{a}_{tar}, \mathbf{b}_{tar})$. Let us consider the monolayer rest configuration $\left(\mathbf{a}_r^{ortho}, \mathbf{b}_r^{ortho}\right)$ described by $\mathbf{a}_r^{ortho} = \mathbf{a}_{tar}$ and $\mathbf{b}_r^{ortho} = \mathbf{b}_{tar}$, that in a general case prescribes local orthotropic in-plane strain. According to the expression for the elastic energy (Eqn. 4.9), imposition of this rest configuration makes the plate adopt exactly the target configuration, because the elastic energy in this case equals zero. Following [7], we express the equivalent bilayer rest configuration $\left(\mathbf{a}_{r,t}^{ortho}, \mathbf{a}_{r,b}^{ortho}\right)$ using Eqn. (4.13):

$$\begin{cases} \mathbf{a}_{r,t}^{ortho} = \mathbf{a}_r^{ortho} - \dfrac{2h}{3}\mathbf{b}_r^{ortho}; \\ \mathbf{a}_{r,b}^{ortho} = \mathbf{a}_r^{ortho} + \dfrac{2h}{3}\mathbf{b}_r^{ortho}. \end{cases} \tag{4.14}$$

Equivalently, using the definition of $\left(\mathbf{a}_r^{ortho}, \mathbf{b}_r^{ortho}\right)$ we rewrite:

$$\begin{cases} \mathbf{a}_{r,t}^{ortho} = \mathbf{a}_{tar} - \dfrac{2h}{3}\mathbf{b}_{tar}; \\ \mathbf{a}_{r,b}^{ortho} = \mathbf{a}_{tar} + \dfrac{2h}{3}\mathbf{b}_{tar}. \end{cases} \tag{4.15}$$

Thus, application of the rest configuration $\left(\mathbf{a}_{r,t}^{ortho}, \mathbf{a}_{r,b}^{ortho}\right)$ leads to the target shape $(\mathbf{a}_{tar}, \mathbf{b}_{tar})$. However, this configuration implies local orthotropic eigenstrain, which is not feasible with shot peen forming. We comply with this constraint and find a suitable local isotropic eigenstrain based on this prediction. To that end, we first compute the local eigenstrains in the principal directions in the top (ε_{11}^t, ε_{22}^t) and bottom (ε_{11}^b, ε_{22}^b) layers of the bilayer prescribed by $\mathbf{a}_{r,t}^{ortho}$ and $\mathbf{a}_{r,b}^{ortho}$, respectively. Next, we take their averages ε_{avg}^t and ε_{avg}^b and impose them locally in all in-plane directions thus making the initial guess.

To find (ε_{11}^t, ε_{22}^t) and (ε_{11}^b, ε_{22}^b) we perform a spectral decomposition of $\mathbf{a}_{r,t}^{ortho}$ and $\mathbf{a}_{r,b}^{ortho}$, respectively [7]. At the top layer, the distortions prescribed by $\mathbf{a}_{r,t}^{ortho}$ imply stretching by a factor of $(\varepsilon_{11}^t + 1)$ in the first principal direction and by a factor of $(\varepsilon_{22}^t + 1)$ in the orthogonal second principal direction. The first principal direction is rotated by an angle of θ^t with respect to the x-axis. At the bottom layer, the stretch factors are $(\varepsilon_{11}^b + 1)$ and $(\varepsilon_{22}^b + 1)$, and the first principal direction is rotated by an angle of θ^b with respect to the x-axis. The initial configuration is unstretched, so its first fundamental form $\mathbf{a}_{init}$ is represented by the identity matrix:

$$\mathbf{a}_{init} = \mathbf{I}. \tag{4.16}$$

Consequently, the eigenvalues of $\mathbf{a}_t^{ortho}$ equal $(\varepsilon_{11}^t + 1)^2$ and $(\varepsilon_{22}^t + 1)^2$, and the eigenvalues of $\mathbf{a}_b^{ortho}$ equal $(\varepsilon_{11}^b + 1)^2$ and $(\varepsilon_{22}^b + 1)^2$, so that:

$$\mathbf{a}_{r,j}^{ortho} = \begin{bmatrix} \cos(\theta^j) & -sin(\theta^j) \\ \sin(\theta^j) & \cos(\theta^j) \end{bmatrix}^T \begin{bmatrix} (\varepsilon_{11}^j + 1)^2 & 0 \\ 0 & (\varepsilon_{22}^j + 1)^2 \end{bmatrix} \begin{bmatrix} \cos(\theta^j) & -sin(\theta^j) \\ \sin(\theta^j) & \cos(\theta^j) \end{bmatrix}, \quad \text{for} \quad j = t, b. \tag{4.17}$$

We deduce the local eigenstrain in the principal directions (ε_{11}^t, ε_{22}^t) and (ε_{11}^b, ε_{22}^b) from the eigenvalues and calculate the average local eigenstrain ε_{avg}^t and ε_{avg}^b for both layers:

$$\varepsilon_{avg}^j = \frac{\varepsilon_{11}^j + \varepsilon_{22}^j}{2}, \quad \text{for} \quad j = t, b. \tag{4.18}$$

We impose the local isotropic eigenstrain ε_{avg}^t and ε_{avg}^b on the top and bottom layers, respectively, thus making the initial guess. The bilayer rest fundamental forms $\mathbf{a}_{r,t}$ and $\mathbf{a}_{r,b}$ corresponding to this strain are expressed as:

$$\mathbf{a}_{r,j} = \begin{bmatrix} (\varepsilon_{avg}^j + 1)^2 & 0 \\ 0 & (\varepsilon_{avg}^j + 1)^2 \end{bmatrix}, \quad \text{for} \quad j = t, b. \tag{4.19}$$

We substitute $(\mathbf{a}_{r,t}, \mathbf{a}_{r,b})$ to the bilayer elastic energy functional (Eqn. 4.11) and find the *current* shape $(\mathbf{a}_c, \mathbf{b}_c)$ that minimizes the elastic energy.

Following Eqn. (4.12), it is possible to find equivalent monolayer rest fundamental forms $(\mathbf{a}_r, \mathbf{b}_r)$. Prescription of the rest fundamental forms $(\mathbf{a}_r, \mathbf{b}_r)$ to a monolayer plate leads to the same current shape $(\mathbf{a}_c, \mathbf{b}_c)$ as the prescription of $(\mathbf{a}_{r,t}, \mathbf{a}_{r,b})$ to a bilayer plate. Essentially, $(\mathbf{a}_r, \mathbf{b}_r)$ can be viewed as the rest configuration imposed on the bilayer mid-surface. The forms $\mathbf{a}_r$ and $\mathbf{b}_r$ may be incompatible as they were defined analytically, but $\mathbf{a}_c$ and $\mathbf{b}_c$ are always compatible because they describe a surface in Euclidean space. Consequently, $(\mathbf{a}_c, \mathbf{b}_c)$ are typically different from $(\mathbf{a}_r, \mathbf{b}_r)$.

Adjustment of the local stretching

In the previous subsection, we defined a procedure to obtain a quick estimate of the inverse problem by solving for orthotropic expansions and averaging them to estimate isotropic expansions. In this subsection and the next, we seek to iteratively correct these expansions

by using only local information.

As the rest configuration $(\mathbf{a}_{r,t}, \mathbf{a}_{r,b})$ prescribes local *isotropic* in-plane strain, we measure stretching in terms of local areas. The area A of each region of the surface constrained by $(x, y) \in U_1 \subset \mathbb{R}^2$ is expressed in terms of the first fundamental form $\mathbf{a}$ as $A = \iint_{U_1} \sqrt{det(\mathbf{a})}\, dxdy$. The first fundamental form is considered constant inside small regions, so we conclude that the current area of each small region A_c and its target area A_{tar} are related as:

$$\frac{A_{tar}}{A_c} \approx \frac{\sqrt{\det(\mathbf{a}_{tar})}}{\sqrt{\det(\mathbf{a}_c)}} = k_A. \tag{4.20}$$

This means that if we locally multiply $\mathbf{a}_c$ by the coefficient k_A, then the current area will equal that of the target. However, we are only able to influence $\mathbf{a}_c$ indirectly through the adjustment of $\mathbf{a}_r$. Consequently, as a part of the iterative procedure, we multiply $\mathbf{a}_r$ by k_A and thus obtain the equivalent monolayer rest fundamental form $\mathbf{a}_r^{new}$ to be imposed during the subsequent iteration:

$$\mathbf{a}_r^{new} = k_A \mathbf{a}_r. \tag{4.21}$$

The $\mathbf{a}_r^{new}$ may be different from the current first fundamental form on the subsequent iteration $\mathbf{a}_c^{new}$, and thus this correction of the rest fundamental form does not lead to an exact solution but allows to approach it. In other words, multiplication of $\mathbf{a}_r$ by k_A does not correct the local area exactly by the coefficient k_A but reduces the difference between the current local stretching and the target one.

Adjustment of the local curvature

We characterize the surface curvature in terms of the local mean curvature H. By definition, H is the average of two local principal curvatures κ_1 and κ_2, that are computed as eigenvalues of the shape operator $\mathbf{S} = \mathbf{a}^{-1}\mathbf{b}$ [84]. We compute the local ratios k_H between the current mean curvatures H_c and the target mean curvatures H_{tar} and assign an upper threshold δ for $|k_H|$:

$$\begin{cases} k_H = \dfrac{H_{tar}}{H_c} & \text{for} \quad \left|\dfrac{H_{tar}}{H_c}\right| < \delta; \\ k_H = \delta \cdot sgn\left(\dfrac{H_{tar}}{H_c}\right) & \text{for} \quad \left|\dfrac{H_{tar}}{H_c}\right| \geq \delta. \end{cases} \tag{4.22}$$

Here, H_{tar} is the average of the two eigenvalues of $\mathbf{S}_{tar} = \mathbf{a}_{tar}^{-1}\mathbf{b}_{tar}$, and H_c is the average of

the two eigenvalues of $\mathbf{S}_c = \mathbf{a}_c^{-1}\mathbf{b}_c$. The threshold δ is assigned in order to deal with special cases when $|H_c|$ is small. Provided that multiplication of a matrix by a constant multiplies its eigenvalues by the same constant, the multiplication of $\mathbf{S}_c$ by k_H would make the current local mean curvatures equal to those of the target. As we are unable to adjust any of the current fundamental forms $(\mathbf{a}_c, \mathbf{b}_c)$ directly, we influence them through adjustment of the rest fundamental forms $(\mathbf{a}_r, \mathbf{b}_r)$ in order to get:

$$\mathbf{S}_r^{new} = k_H \mathbf{S}_r = k_H \mathbf{a}_r^{-1}\mathbf{b}_r = k_H k_A \frac{\mathbf{a}_r^{-1}}{k_A}\mathbf{b}_r = k_H k_A \left(\mathbf{a}_r^{new}\right)^{-1}\mathbf{b}_r. \tag{4.23}$$

Thus, we define:

$$\mathbf{b}_r^{new} = k_H k_A \mathbf{b}_r. \tag{4.24}$$

Once the $\mathbf{a}_r^{new}$ and $\mathbf{b}_r^{new}$ are found, we compute the bilayer rest fundamental forms $(\mathbf{a}_{r,t}^{new}, \mathbf{a}_{r,b}^{new})$ as:

$$\begin{cases} \mathbf{a}_{r,t}^{new} = \mathbf{a}_r^{new} - \dfrac{2h}{3}\mathbf{b}_r^{new}; \\ \mathbf{a}_{r,b}^{new} = \mathbf{a}_r^{new} + \dfrac{2h}{3}\mathbf{b}_r^{new}. \end{cases} \tag{4.25}$$

Next, we substitute $(\mathbf{a}_{r,t}^{new}, \mathbf{a}_{r,b}^{new})$ to the bilayer elastic energy functional (Eqn. 4.11) and find the current shape $(\mathbf{a}_c^{new}, \mathbf{b}_c^{new})$ that minimizes the elastic energy.

Subsequent iterations and stop criterion

We compare the current shape with the target shape and recalculate the bilayer rest fundamental forms until a convergence criterion is satisfied. The convergence criterion is based on the calculation of the Hausdorff distance d_H between the new current configuration $\mathcal{C}_c^{new}$ defined by $(\mathbf{a}_c^{new}, \mathbf{b}_c^{new})$ and the current configuration from the previous iteration $\mathcal{C}_c$ defined by $(\mathbf{a}_c, \mathbf{b}_c)$. We nondimensionalize d_H by the square root of the total area of the plate in its initial configuration A_{total}. We stop iterating either when a predefined maximal number of iterations M is reached, or when the nondimensionalized Hausdorff distance becomes inferior to a chosen threshold τ:

$$\frac{d_H(\mathcal{C}_c, \mathcal{C}_c^{new})}{\sqrt{A_{total}}} < \tau. \tag{4.26}$$

Calculation of the adjusted eigenstrain

We denote the bilayer *final rest* fundamental forms, i.e., the ones obtained on the last iteration, as $(\mathbf{a}_{rf,t}\ , \mathbf{a}_{rf,b})$. To relate them with peen forming parameters, we determine the recalculated eigenstrain $(\varepsilon^{rf,t}, \varepsilon^{rf,b})$.

Provided that the iterative adjustment implies only addition, subtraction and multiplication by a constant of diagonal matrices, the $\mathbf{a}_{rf,t}$ and $\mathbf{a}_{rf,b}$ are diagonal. Moreover, the imposed local eigenstrain $(\varepsilon^{rf,t}, \varepsilon^{rf,b})$ is isotropic, so $\mathbf{a}_{rf,t}$ and $\mathbf{a}_{rf,b}$ have the following form:

$$\mathbf{a}_{rf,j} = \begin{bmatrix} \left(\varepsilon^{rf,j}+1\right)^2 & 0 \\ 0 & \left(\varepsilon^{rf,j}+1\right)^2 \end{bmatrix}, \quad \text{for} \quad j = t, b. \tag{4.27}$$

Consequently,

$$\varepsilon^{rf,j} = \sqrt{a^{11}_{rf,j}} - 1, \quad \text{for} \quad j = t, b. \tag{4.28}$$

4.3.2 Numerical implementation

We mesh the plate mid-surface with triangular elements and follow the energy calculation strategy presented in [59] [61]. The first and the second fundamental forms are estimated separately for each element and are constant inside the element. The global elastic energy is calculated as a sum of local energetical contributions from all the elements.

The first fundamental form on a triangular element such as that schematized in Fig. 4.5 depends entirely on the coordinates of the vertices. The three vertices are defined by position vectors $\vec{v}_0$, $\vec{v}_1$ and $\vec{v}_2$, and the edge vectors constituting the triangle are expressed as: $\vec{e}_0 = \vec{v}_1 - \vec{v}_0$, $\vec{e}_1 = \vec{v}_2 - \vec{v}_1$, $\vec{e}_2 = \vec{v}_0 - \vec{v}_2$. These vectors are tangent to the plane containing the triangle, so following Eqn. (4.7), the first fundamental form on a triangular element is computed as:

$$\mathbf{a} = \begin{bmatrix} \vec{e}_1 \cdot \vec{e}_1 & \vec{e}_1 \cdot \vec{e}_2 \\ \vec{e}_2 \cdot \vec{e}_1 & \vec{e}_2 \cdot \vec{e}_2 \end{bmatrix}. \tag{4.29}$$

In a general case of isotropic eigenstrain ε^r imposed on a triangular element, its rest first fundamental form $\mathbf{a}_r$ is expressed as:

$$\mathbf{a}_r = \begin{bmatrix} \vec{e}_1^{\,init} \cdot \vec{e}_1^{\,init} & \vec{e}_1^{\,init} \cdot \vec{e}_2^{\,init} \\ \vec{e}_2^{\,init} \cdot \vec{e}_1^{\,init} & \vec{e}_2^{\,init} \cdot \vec{e}_2^{\,init} \end{bmatrix} \begin{bmatrix} (\varepsilon^r + 1)^2 & 0 \\ 0 & (\varepsilon^r + 1)^2 \end{bmatrix} = \mathbf{a}_{init} \begin{bmatrix} (\varepsilon^r + 1)^2 & 0 \\ 0 & (\varepsilon^r + 1)^2 \end{bmatrix}, \tag{4.30}$$

where $\vec{e}_1^{\,init}$, $\vec{e}_2^{\,init}$, $\vec{e}_3^{\,init}$ are the edge vectors in the initial configuration and $\mathbf{a}_{init}$ is the element first fundamental form in its initial configuration. Otherwise, if an orthotropic in-plane strain is imposed, then the rest first fundamental form $\mathbf{a}_r^{ortho}$ becomes:

$$\mathbf{a}_r^{ortho} = \mathbf{a}_{init} \begin{bmatrix} \cos(\theta) & -sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix}^T \begin{bmatrix} (\varepsilon_{11}^r + 1)^2 & 0 \\ 0 & (\varepsilon_{22}^r + 1)^2 \end{bmatrix} \begin{bmatrix} \cos(\theta) & -sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix}, \tag{4.31}$$

where ε_{11}^r and ε_{22}^r define the eigenstrain imposed in the principal directions and θ stands for the angle between the first principal direction and the x axis.

The second fundamental form defines the surface curvature, so the information about surface normals is required. In this connection we introduce a unit normal vector $\vec{n}_i,\ i = 1, 2, 3$ at the center of each edge of the mesh (the *edge-director*). This vector is normal to the edge, and its angle of inclination $\varphi_i,\ i = 1, 2, 3$ in the plane perpendicular to the edge provides a supplementary degree of freedom. This angle is measured with respect to the average of the

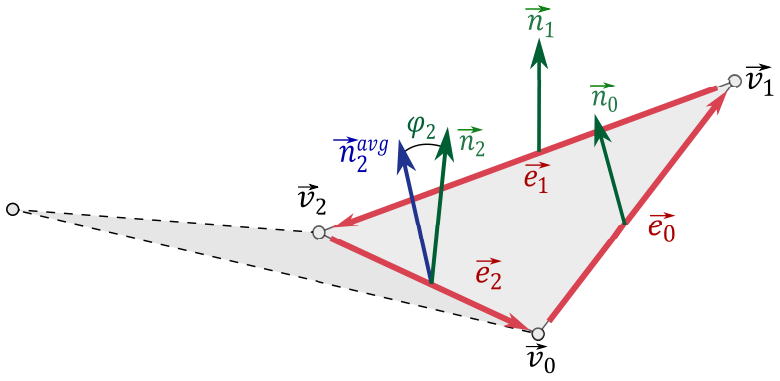

Figure 4.5 A triangular mesh element and vectors that determine local fundamental forms: the vertex position vectors ($\vec{v}_0$, $\vec{v}_1$ and $\vec{v}_2$), the edge vectors ($\vec{e}_0$, $\vec{e}_1$ and $\vec{e}_2$) and the mid-edge normals ($\vec{n}_0$, $\vec{n}_1$ and $\vec{n}_2$). The mid-edge normals are perpendicular to the edge, and their direction is determined by the angle of inclination with respect to the average normal of the two adjacent faces. Thus, the direction of $\vec{n}_2$ is determined by the angle φ_2. This angle is measured with respect to $\vec{n}_{avg}$, which is the average normal of the two faces that share the edge $\vec{e}_2$.

adjacent face normals [61]. A finite-difference approximation of the derivatives appearing in Eqn. (4.8) yields the following expression for the second fundamental form of a triangular element:

$$\mathbf{b} = \begin{bmatrix} \vec{e}_1 \cdot 2(\vec{n}_0 - \vec{n}_2) & \vec{e}_1 \cdot 2(\vec{n}_1 - \vec{n}_0) \\ \vec{e}_1 \cdot 2(\vec{n}_1 - \vec{n}_0) & \vec{e}_2 \cdot 2(\vec{n}_1 - \vec{n}_0) \end{bmatrix} = \begin{bmatrix} \vec{e}_1 \cdot 2(\vec{n}_0 - \vec{n}_2) & -\vec{e}_1 \cdot \vec{n}_0 \\ -\vec{e}_1 \cdot \vec{n}_0 & \vec{e}_2 \cdot 2(\vec{n}_1 - \vec{n}_0) \end{bmatrix}. \tag{4.32}$$

According to Eqn. (4.11), the global elastic energy for a plate composed of K triangular elements is expressed in terms of local fundamental forms as:

$$\begin{aligned} E_{BL} = \frac{1}{2}\sum_{k=1}^{K} \Bigg[& \frac{h_k}{8} \left\| (\mathbf{a}_{r,b}^{-1})_k (\mathbf{a}_f)_k - \mathbf{I} \right\|_e^2 + \frac{(h_k)^3}{24} \left\| (\mathbf{a}_{r,b}^{-1})_k (\mathbf{b}_f)_k \right\|_e^2 \\ & + \frac{(h_k)^2}{8} \left\langle \left((\mathbf{a}_{r,b}^{-1})_k (\mathbf{a}_f)_k - \mathbf{I} \right), (\mathbf{a}_{r,b}^{-1})_k (\mathbf{b}_f)_k \right\rangle_e \Bigg] \sqrt{\det (\mathbf{a}_{r,b})_k} \\ & + \frac{1}{2}\sum_{k=1}^{K} \Bigg[\frac{h_k}{8} \left\| (\mathbf{a}_{r,t}^{-1})_k (\mathbf{a}_f)_k - \mathbf{I} \right\|_e^2 + \frac{(h_k)^3}{24} \left\| (\mathbf{a}_{r,t}^{-1})_k (\mathbf{b}_f)_k \right\|_e^2 \\ & - \frac{(h_k)^2}{8} \left\langle \left((\mathbf{a}_{r,t}^{-1})_k (\mathbf{a}_f)_k - \mathbf{I} \right), (\mathbf{a}_{r,t}^{-1})_k (\mathbf{b}_f)_k \right\rangle_e \Bigg] \sqrt{\det (\mathbf{a}_{r,t})_k}. \end{aligned} \tag{4.33}$$

The local plate thickness h_k can be different for each element. Similarly, the local Young's modulus Y_k and the Poisson's ratio ν_k can vary along the plate.

As follows from Eqn. (4.30), the element's rest first fundamental forms $\mathbf{a}_{r,t}$ and $\mathbf{a}_{r,b}$ for the isotropic growth case are defined by the imposed local eigenstrain ε^t and ε^b and the initial vertex positions $\vec{v}_i{}^{init}$, $i = 1, 2, 3$. At the same time, according to Eqn. (4.29) and Eqn. (4.32), the two final fundamental forms $\mathbf{a}_f$ and $\mathbf{b}_f$ are defined by the final vertex positions $\vec{v}_i^f$, $i = 1, 2, 3$, and final angles of inclination of the edge directors φ_i^f, $i = 1, 2, 3$. Numerically, the forward problem consists in minimizing the global elastic energy functional (Eqn. 4.33) with respect to $\vec{v}_i^f$ and φ_i^f provided with $\vec{v}_i{}^{init}$, ε^t and ε^b. We perform the minimization using a quasi-Newton L-BFGS algorithm [85]. The gradients of the elastic energy functional required by the minimization algorithm are computed analytically following [86]. The corresponding programming code for the forward problem resolution was developed by van Rees et al. and is publicly accessible [87].

In this implementation, the inverse problem resolution consists in finding the local eigenstrain $\varepsilon^{rf,t}$ and $\varepsilon^{rf,b}$ to be imposed on each triangular element. This means that the iterative correction of the rest fundamental forms $\mathbf{a}_{r,t}$ and $\mathbf{a}_{r,b}$ is executed on a local scale for each element separately, whilst the forward problem is resolved on each iteration on a global scale, thus reflecting the mechanics of the plate.

Algorithm 1 The inverse problem resolution

The iterative loop

while condition (4.26) is not satisfied and the number of iterations is below maximum **do**
  for each triangular element **do**
    Find principal curvatures (κ_1^c, κ_2^c) as eigenvalues of the shape operator $\mathbf{S}_c = \mathbf{a}_c^{-1}\mathbf{b}_c$
    Compute the current mean curvature as $H_c = 0.5\,(\kappa_1^c + \kappa_2^c)$
    Compute the current area as $A_c = \sqrt{\det(\mathbf{a}_c)}$
    Compute the ratios k_A and k_H following Eqn. (4.20) and Eqn. (4.22) respectively
    Compute the monolayer rest fundamental forms $\mathbf{a}_r$ and $\mathbf{b}_r$ following Eqn. (4.12) using only local information
    Compute the adjusted monolayer rest fundamental forms $\mathbf{a}_r^{new}$ and $\mathbf{b}_r^{new}$ following Eqn. (4.21) and Eqn. (4.24) respectively
    Compute the adjusted bilayer rest fundamental forms $\mathbf{a}_{r,t}^{new}$ and $\mathbf{a}_{r,b}^{new}$ following Eqn. (4.25)
  end for
  Substitute $\mathbf{a}_{r,t}^{new}$ and $\mathbf{a}_{r,b}^{new}$ to the bilayer elastic energy functional (Eqn. 4.33) and minimize it to solve the forward problem and find the current configuration ($\mathbf{a}_c^{new}$,$\mathbf{b}_c^{new}$)
end while

The initial guess

for each triangular element **do**
  Compute the first fundamental form of the initial shape $\mathbf{a}_{init}$ following Eqn. (4.29)
  Compute the monolayer fundamental forms of the target shape $\mathbf{a}_{tar}$ and $\mathbf{b}_{tar}$ following Eqn. (4.29) and Eqn. (4.32) respectively
  Find principal curvatures κ_1^{tar} and κ_2^{tar} as eigenvalues of the target shape operator $\mathbf{S}_{tar} = \mathbf{a}_{tar}^{-1}\mathbf{b}_{tar}$
  Compute the target mean curvature as $H_{tar} = 0.5(\kappa_1^{tar} + \kappa_2^{tar})$
  Compute the target area as $A_{tar} = \sqrt{\det(\mathbf{a}_{tar})}$
  Compute the orthotropic bilayer rest fundamental forms $\mathbf{a}_{r,t}^{ortho}$ and $\mathbf{a}_{r,b}^{ortho}$ following Eqn. (4.15)
  Find eigenvalues λ_1^j and λ_2^j of $(\mathbf{a}_{init}^{-1}\mathbf{a}_{r,j}^{ortho})$ for $j = t, b$
  Compute the orthotropic eigenstrain as $\varepsilon_{ii}^j = \sqrt{\lambda_i^j} - 1$ for $i = 1, 2$ and $j = t, b$

Algorithm 1 The inverse problem resolution

10: Compute the average eigenstrain ε^t_{avg} and ε^b_{avg} following Eqn. (4.18)
11: Compute bilayer rest fundamental forms $\mathbf{a}_{r,t}$ and $\mathbf{a}_{r,b}$ by substituting ε^t_{avg} and ε^b_{avg} respectively for ε^r in Eqn. (4.30)
12: **end for**
13: Substitute $\mathbf{a}_{r,t}$ and $\mathbf{a}_{r,b}$ to the bilayer elastic energy functional (Eqn. 4.33) and minimize it to solve the forward problem and find the current configuration $(\mathbf{a}_c, \mathbf{b}_c)$

Final step after exiting the iterative loop

1: **for** each triangular element **do**
2: Compute the eigenstrain $\varepsilon^{rf,t}$ and $\varepsilon^{rf,b}$ prescribed by $\mathbf{a}_{rf,t} = \mathbf{a}^{new}_{r,t}$ and by $\mathbf{a}_{rf,b} = \mathbf{a}^{new}_{r,b}$ respectively following Eqn. (4.28)
3: **end for**

4.3.3 Grouping of the peening pattern

In the general case, the eigenstrains $(\varepsilon^{rf,t}, \varepsilon^{rf,b})$ provided by Algorithm 1 are different for each element and can take any real values. Peen forming often deals with smoothly curved target shapes, so $(\varepsilon^{rf,t}, \varepsilon^{rf,b})$ may also vary smoothly along the surface given that these two values depend on the target shape curvature. We call the eigenstrain pattern provided by the Algorithm 1 the *free* pattern. From a practical point of view, each pair $(\varepsilon^{rf,t}, \varepsilon^{rf,b})$ represents a peening regime. However, a limited number of regimes is available when peening a real part. Thus, we divide the pattern into zones with uniform prescribed eigenstrain and obtain a *grouped* pattern.

We associate all triangular elements with points on a plane with cartesian coordinates $\left(\varepsilon^t, \varepsilon^b\right)$, and the coordinates of each point k are determined by the eigenstrain $\left(\varepsilon^{rf,t}_k, \varepsilon^{rf,b}_k\right)$ assigned to the corresponding element k, as illustrated in Fig. 4.6. We divide the points in groups, and the group centroids are determined by the predefined peening regimes. We denote the centroid of a group n by $\left(\varepsilon^{cen,t}_n, \varepsilon^{cen,b}_n\right)$. Each point $\left(\varepsilon^{rf,t}_k, \varepsilon^{rf,b}_k\right)$ is attributed to the group with the closest centroid in terms of Euclidean distance. When all points are grouped, we homogenize the eigenstrain inside each group, i.e., we assign the eigenstrain $\left(\varepsilon^{cen,t}_n, \varepsilon^{cen,b}_n\right)$ to all triangular elements that fall into the group n.

Consider N predefined peening regimes such as the ones illustrated in Fig. 4.7. Regime $i = 1, 2, ..., N$ induces expansions ε^t_i and ε^b_i on the treated and opposite layers, respectively. In addition, we consider the lack of treatment $\varepsilon^t_0 = \varepsilon^b_0 = 0$. Since the top and bottom

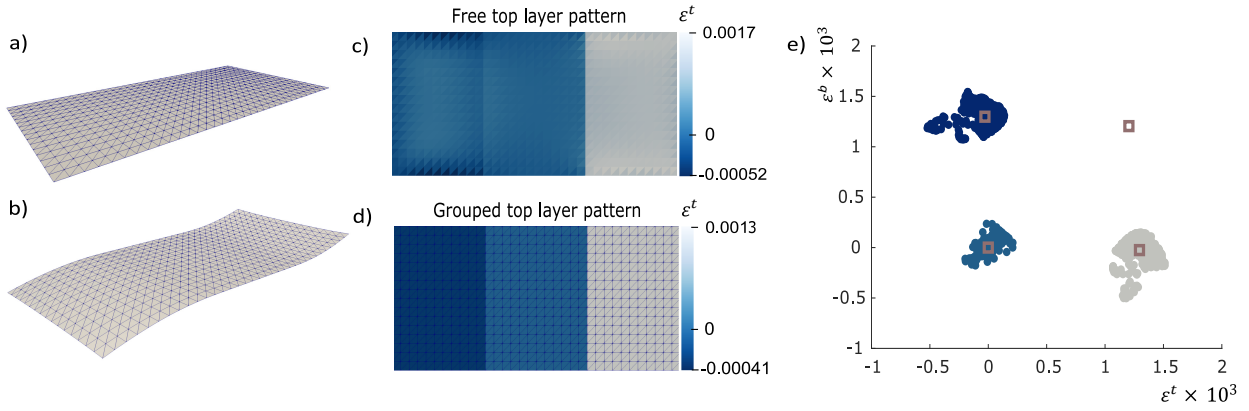

Figure 4.6 Graphical representation of grouping of the eigenstrain pattern. The triangulated flat initial configuration (a) and the target configuration - wavy shape (b) - are the input data for the inverse problem resolution. We divide the eigenstrain pattern (c) into three zones treated uniformly (d). A plane with cartesian coordinates $(\varepsilon^t, \varepsilon^b)$ (e) illustrates the grouping from the numerical point of view. The colored points correspond to the eigenstrain assigned to each element of the triangular mesh. The four centroids are denoted by colored squares. They are generated by one peening regime and the lack of treatment as an additional regime. The grouping is based on calculation of the least Euclidean distance from the points to the group centroids. In the presented case there are no points close to the centroid denoting treatment from both sides, so the corresponding group is empty. Once the points are divided in groups, we homogenize the eigenstrain for all the elements attributed to the same group.

surfaces can be peened independently, there are $(N+1)^2$ possible treatment combinations. Each treatment combination gives rise to a group centroid. Combining regime $i = 1, 2, ..., N$ on the top surface with regime $j = 1, 2, ..., N$ on the bottom surface leads to the following expansions of the top and bottom surface: $(\varepsilon_i^t, \varepsilon_i^b) + (\varepsilon_j^t, \varepsilon_j^b) = (\varepsilon_{ij}^{cen,t}, \varepsilon_{ij}^{cen,b})$. Figure 4.7 illustrates this principle. Figure 4.6 e) also provides a cartesian representation of $(N+1)^2$ centroids for the case $N = 1$.

4.3.4 Numerical validation of the inverse problem solver

We generated target shapes numerically to test our algorithms for the iterative inverse problem resolution and grouping. To ensure that the target shapes were achievable with peen forming, we generated them by assigning a random peening pattern to the initial configuration and then solved the forward problem. The random peening patterns were generated following Algorithm 2.

When the target shapes were generated, we solved the inverse problem for each of them following Algorithm 1 and then grouped the peening pattern. The predefined regimes were fixed as those that were originally used to generate the target shapes. To quantify the error,

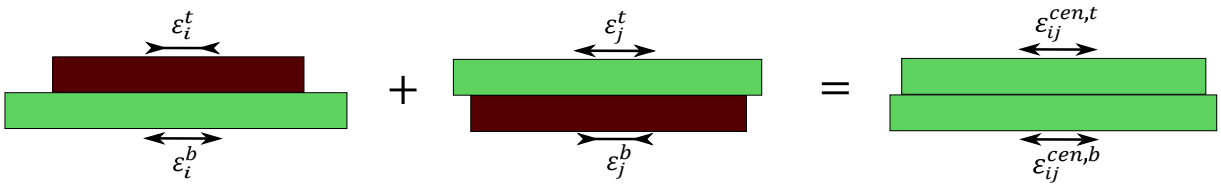

Figure 4.7 Calculation of the group centroids. The group ij implies treatment from the bottom side with regime i and treatment from the top side with regime j. Its centroid is determined by a pair of parameters $\varepsilon_{ij}^{cen,t}$ and $\varepsilon_{ij}^{cen,b}$, and each of them is a superposition of eigenstrain generated by the two regimes that form this group: $\varepsilon_{ij}^{cen,t} = \varepsilon_i^t + \varepsilon_j^t$, $\varepsilon_{ij}^{cen,b} = \varepsilon_i^b + \varepsilon_j^b$. In this example, the regime i is more intense than the regime j, so $\varepsilon_{ij}^{cen,t} < \varepsilon_{ij}^{cen,b}$.

we solved the forward problem for the free and the grouped patterns. We thus obtained two final shapes for each test case and compared them with the target shape by calculating the nondimensionalized Hausdorff distance Ω:

$$\Omega = \frac{d_H(\mathcal{C}_f, \mathcal{C}_{tar})}{\sqrt{A_{total}}}, \tag{4.34}$$

where $\mathcal{C}_f$ stands for the final configuration and $\mathcal{C}_{tar}$ stands for the target configuration. The overall process for the inverse problem validation is schematized in Fig. 4.8.

Algorithm 2 Generation of random peening patterns

1: Mark 1 to 6 random points on the top and bottom surfaces of the plate
2: **for** each point **do**
3: Draw a square of random size (but not bigger than the plate size) centred on the point
4: Assign randomly one of the available peening regimes to the square
5: **if** the square protrudes beyond the plate area **then**
6: Translate the part that protrudes symmetrically on the other side of the plate
7: **end if**
8: **if** the square superimposes with a previously drawn square on the same side **then**
9: Erase the previously assigned regime in the superimposing area and leave only the latest one
10: **end if**
11: **end for**

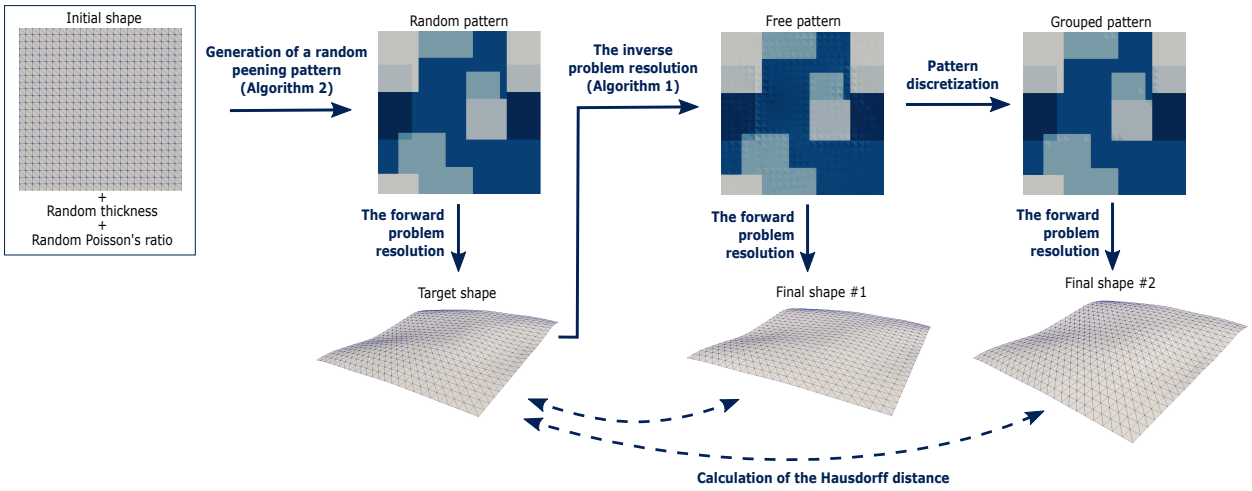

Figure 4.8 A validation process to test the quality of the inverse problem resolution. The initial shape is fixed as a flat 1×1 m plate, its thickness and Poisson's ratio are chosen randomly. These parameters are kept constant throughout the whole validation process. A random peening pattern is generated following Algorithm 2 and assigned to the initial shape. Different colors on the peening pattern correspond to different peening regimes applied from the top, bottom or both sides. We solve the forward problem taking the random pattern as input, and the result is used as the target shape for the inverse problem validation. The eigenstrain pattern is determined following Algorithm 1 and grouped using pre-determined regimes. Finally, the forward problem is resolved for the free and the grouped patterns. The difference between the target shape and the two final shapes is quantified with the nondimensionalized Hausdorff distance Ω.

4.4 Results

We generated 200 random patterns with Algorithm 2 and applied them on a flat square plate (1×1 m). The plate thickness was arbitrarily assigned in each case and ranged from 2 mm to 15 mm. The Poisson's ratio was also arbitrarily picked between 0.32 and 0.36. The plate was meshed with 1152 triangular elements. The forward problem resolution took 10-30 seconds for one shape, depending on the pattern and the plate thickness. Thus, the forward problem resolution took longer time for thinner plates with bigger treated areas due to larger deflection of these plates.

For the first 100 test cases (*series 1*), we made only one peening regime available, so the entire treated area was peened with the same parameters. For the second 100 test cases (*series 2*), we assigned randomly one of four available peening regimes to each square on both sides. We considered real peening regimes presented in [2]. The authors of this paper deduced idealized eigenstrain profiles from the residual stress measurements performed on

the treated specimens. The idealized eigenstrain profiles were formulated as one uniformly expanding layer of a constant thickness. The layer thickness and the eigenstrain magnitude were different for each regime. We reformulated the idealized eigenstrain profiles in terms of $\left(\varepsilon^t, \varepsilon^b\right)$ by equating the total eigenstrain Γ and the first eigenstrain moment Γ_1 induced by $\left(\varepsilon^t, \varepsilon^b\right)$ and by the one expanding layer. Table 4.1 summarizes the eigenstrain $\left(\varepsilon^t, \varepsilon^b\right)$ induced by each of the four regimes applied on a 5 mm thick plate from the top side.

Figures 4.9 and 4.10 present the free and the grouped patterns along with the convergence curves for two particular test cases from series 2: a low-error case and a high-error case. Figures 4.9 b) and 4.10 b) show that the free pattern on the final iteration is locally close to the originally generated random pattern. Due to that, most of the elements are grouped correctly, so that the eigenstrains prescribed by the grouped and the random patterns to these elements become equal (Fig. 4.9 c and Fig. 4.10 c). However, in each case there are elements that are attributed to a wrong group. In the low-error case this happens only for several elements. Consequently, the dimensionless error Ω is lower for the grouped pattern than for the free pattern. In the high-error case the grouped pattern undergoes the *checkerboard* problem, meaning that the pattern locally alternates two peening regimes over a certain area (Fig. 4.10 c). A large area affected by the checkerboard problem increases the Ω in comparison with the free pattern (Fig. 4.10 e). Nevertheless, the regimes that are mixed up in the checkerboard-affected zone have only a slight difference in terms of the induced eigenstrain, so the Ω increases up to 0.23% at most.

In terms of convergence, the most important correction is done on the first iteration after the initial guess, as illustrated by Fig. 4.9 e) and Fig. 4.10 e). Thus, the first iteration decreases the Ω by 65% on average, and the Ω becomes inferior to 0.1% for all the cases. All the subsequent iterations together decrease the Ω obtained on the first iteration by 60% on average. Because of the local nature of the eigenstrain adjustment, the solution does not

Table 4.1 The in-plane eigenstrain induced by the regimes used to generate the random peening patterns. The in-plane eigenstrain is presented for the case of a 5 mm thick plate. The peening regimes represent four real treatments inducing different eigenstrain profiles, which are examined in [2].

Regime	$\varepsilon^t \times 10^3$	$\varepsilon^b \times 10^3$
1	2.5	-0.4
2	3.2	-0.6
3	1.7	-0.4
4	2.0	-0.4

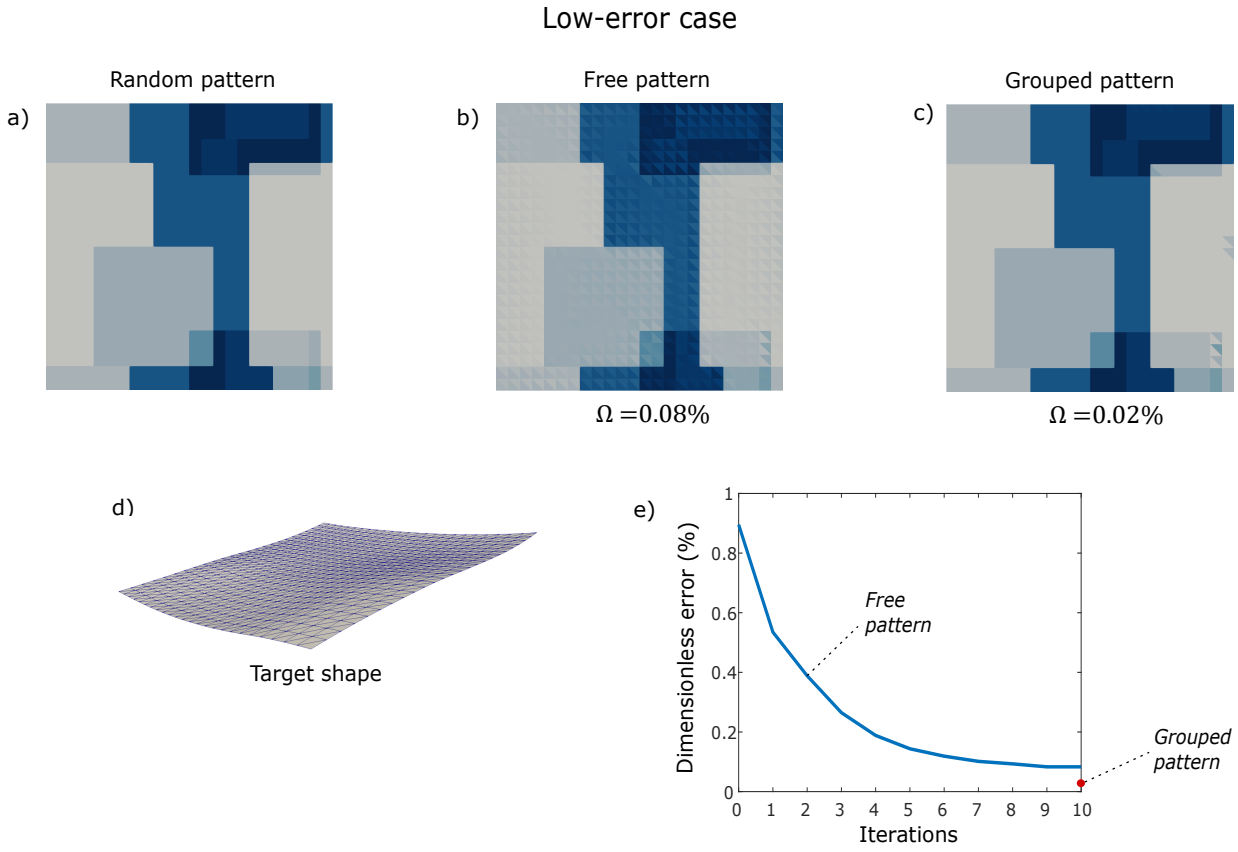

Figure 4.9 One of the test cases from series 2 with low dimensionless error. The plate is 7 mm thick, and the Poisson's ratio equals 0.34. a) The random peening pattern generated for this test case (top layer). Different colors on the peening pattern denote different peening regimes. b) The free peening pattern on the final iteration. c) The grouped peening pattern. A visual comparison of (c) and (a) shows that almost all elements were attributed to a correct group. Consequently, grouping decreased the dimensionless error Ω. d) The target shape induced by the pattern (a). The deformations are at their original scale. e) The convergence curve showing the dimensionless error Ω on each iteration. The optimization required 10 iterations, that was the maximum fixed for this test.

converge to the exact target shape but to a shape which is close to the target. In other words, after several iterations the Ω plateaus at a low but finite level (Fig. 4.9 e and Fig. 4.10 e). The Ω may slightly grow during the subsequent iterations, but the stop criterion (4.26) terminates the iterative process as soon as this happens.

The results of the numerical validation are presented with histograms in Fig. 4.11. The inverse problem solver provided free peening patterns that led to the target shape with the Ω inferior to 0.35% for both series of tests (Fig. 4.11, top). The pattern optimization needed between 2 to 10 iterations, depending on the target shape.

The histograms in Fig. 4.11 show that the pattern grouping has decreased the Ω in most

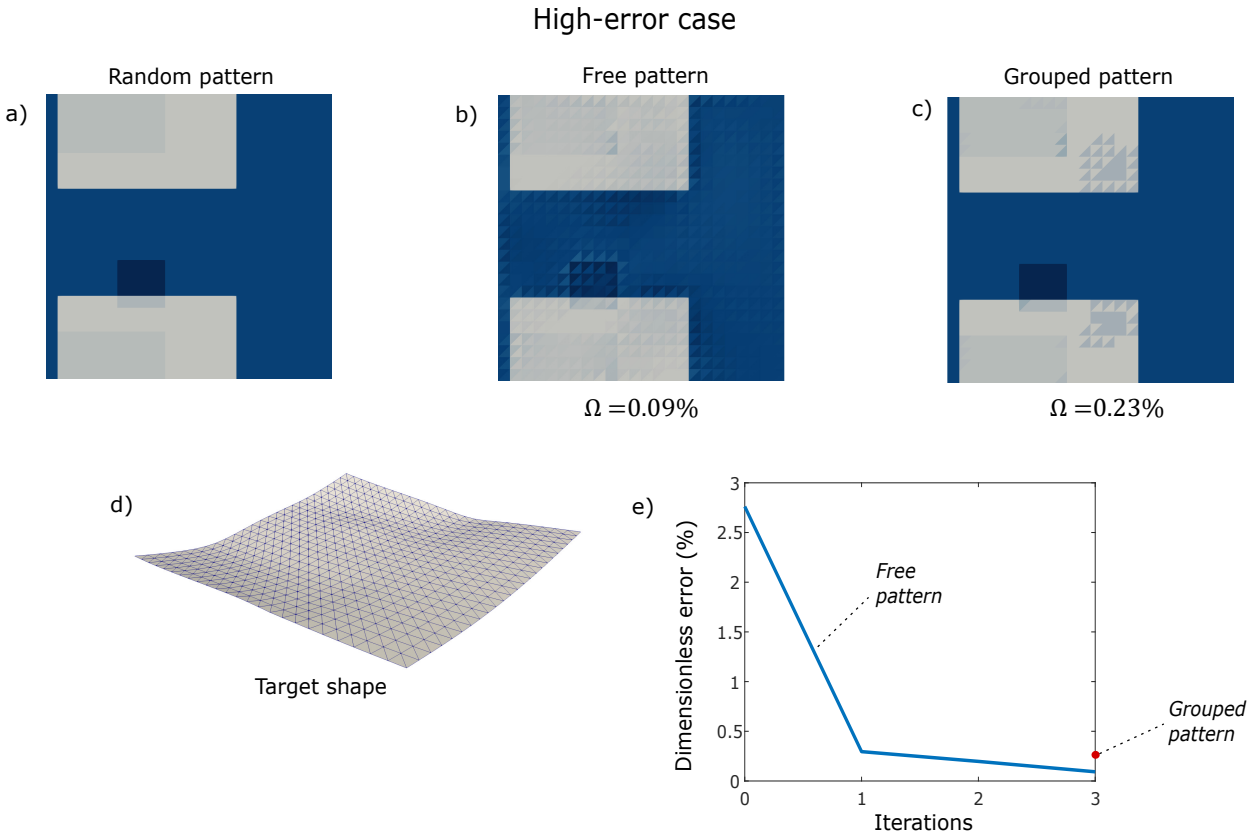

Figure 4.10 One of the test cases from series 2 with high dimensionless error. The plate is 4 mm thick, and the Poisson's ratio equals 0.34. a) The random peening pattern generated for this test case (top layer). Different colors on the peening pattern denote different peening regimes. b) The free peening pattern on the final iteration. c) The grouped peening pattern. A visual comparison of (c) and (a) indicates elements that were not attributed to a correct group. The deformations are at their original scale. However, grouping decreased the dimensionless error Ω in this case. d) The target shape induced by the pattern (a). e) The convergence curve showing the dimensionless error Ω on each iteration. The optimization required 3 iterations to converge.

of the cases. Thus, the original pattern was perfectly reproduced for all test cases in series 1, so the Ω after grouping became less than $10^{-3}\%$. The corresponding Ω for series 2 was bigger because of the higher complexity of the grouping problem: four available regimes induced twenty-five group centroids for series 2, while there were only four centroids induced by one available regime for series 1. The group centroids for series 2 were situated close to each other, so several test cases were significantly affected by the checkerboard problem, as illustrates Fig. 4.10 c). This explains the increased error after grouping for 18 cases out of 100 from series 2.

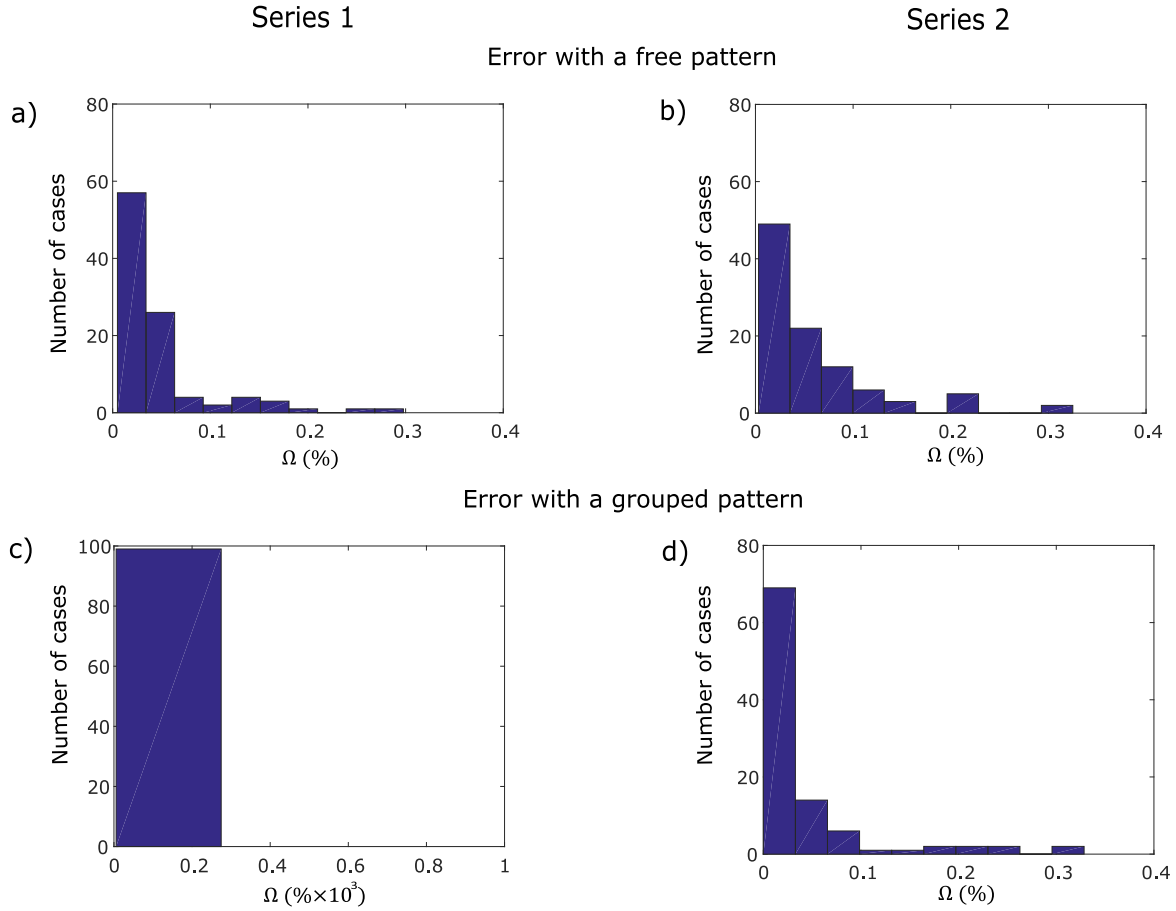

Figure 4.11 Histograms evaluating the dimensionless error Ω between the target shapes and the final shapes obtained during the numerical validation of the inverse problem solver. All in all, 200 test cases were considered. They were divided in two series of 100 cases each. The target shapes in series 1 were obtained with one peening regime, and the same peening regime was fixed as the only available for the grouping stage. The target shapes in series 2 were obtained with four different peening regimes, and the four predefined regimes were available for grouping. Figures (a) and (b) show the dimensionless error induced by application of the free pattern for series 1 and 2 correspondingly. Similarly, figures (c) and (d) represent the dimensionless error induced by the grouped pattern.

4.5 Discussion

The described inverse problem resolution algorithm relies on the assumption that the peening treatment induces isotropic expansions. In practice, however, peen forming sometimes induces different eigenstrain along the x and y axes: $\varepsilon_{xx}(x, y, z) \neq \varepsilon_{yy}(x, y, z)$. Such anisotropic expansions are due to plastic anisotropy of the treated material, which is especially explicit for rolled aluminum sheets, and to prestressing the component in one direction before treatment, i.e., stress peen forming. This effect is examined in detail in [35]. For uniform plastic

anisotropy which does not vary over the area of the plate, the inverse problem resolution algorithm can be easily adapted by introducing a fixed eigenstrain anisotropy coefficient χ in the model:

$$\varepsilon^j_{xx} = \frac{(1+\chi)}{(1-\chi)}\varepsilon^j_{yy}, \quad \text{for} \quad j = t, b. \tag{4.35}$$

This relation may be imposed after adjustment of the bilayer rest fundamental forms $\mathbf{a}_{r,t}$ and $\mathbf{a}_{r,b}$ on each iteration. Thus, the forward problem will be solved taking into account the plastic anisotropy.

Given that the rest fundamental forms are numerically adjusted for each triangular element separately, the efficiency of the inverse problem resolution depends on the consistency between the target mesh and the initial mesh. More precisely, it depends on the mapping $\vec{m}$ between the initial 2D shape and the target 3D shape. The general requirement for the mapping is to preserve the shape of each triangular element as well as possible. This minimizes the local eigenstrain assigned by the algorithm and makes the computed free pattern smoother. This problem was not faced during the numerical validation because the target shapes were derived from the initial shapes through the forward problem resolution, so they were optimally meshed by default.

The mesh consistency can be ensured by fixing the target shape mesh and by its mapping onto the initial 2D geometry. The fixed initial geometry is an important constraint for the mapping because it involves a fixed 2D boundary. The mapping can be done using the methods oriented on maximal preservation of local angles, such as the Least Squares Conformal Mapping (LSCM) algorithm [88]. Next, local mesh distortions with respect to the target mesh can be minimized using a numerical optimization algorithm. For example, the L-BFGS algorithm that we use for the global elastic energy minimization can cope with this task.

4.6 Conclusion

The theory of non-Euclidean plates in combination with the eigenstrain approach provides an extensive theoretical framework for the modeling of shot peen forming. The eigenstrain approach represents the treated plate as a bilayer undergoing nonuniform eigenstrain, and the theory of non-Euclidean plates accurately solves the forward problem for this case. The deformed shape is calculated through minimization of the global elastic energy following analytical gradients.

The iterative inverse problem resolution is based on the comparison of geometrical properties of the plate in its current and target configurations. The adjustment of the prescribed eigenstrain on each iteration is done on a local scale involving simple arithmetic operations and takes negligible amount of time. A low number of iterations (not more than 10) ensures fast resolution of the inverse problem. The eigenstrain formulation of the inverse problem makes the algorithm applicable for any type of processes that induce small isotropic eigenstrain. These include, among others, laser peen forming of metal plates or 4D printing of shape-shifting polymer structures. The precision of the inverse problem resolution is independent of plate thickness and its mechanical properties. It is, however, dependent on consistency between the initial and the target meshes.

The pattern grouping makes the inverse problem solution practically applicable. It adds uncertainty to the solution, but in many cases it decreases the induced error. The grouping algorithm can locally mix up the peening regimes having slightly different intensities. A method for correction of the local grouping errors would enhance the solution quality.

The future work implies experimental validation of the proposed inverse problem simulation technique. It will reveal practical constraints that may cause simulation error. Among others, we will examine influence of the peening parameters and of the material plastic anisotropy on the induced eigenstrain.

4.7 Acknowledgements

The authors gratefully acknowledge financial support from The Fonds de recherche du Québec – Nature et technologies (FRQNT) and The Aluminium Research Center (REGAL).

CHAPTER 5 SEGMENTATION OF THE PEENING PATTERN BASED ON THE k-MEANS CLUSTERING

This chapter deals with the clustering and the filtering algorithms that serve for the pattern segmentation and are not published. Both algorithms are included in the simulation software (see Chapter 4) and can be applied after the inverse problem resolution. The necessity for these two algorithms was revealed during the experimental campaign presented in Chapter 6. The experiments showed that the manual search for optimal peening regimes and the manual correction of the segmentation errors are time-consuming. Therefore, we developed the clustering algorithm, which automatically computes the optimal regimes, and the filtering algorithm, which automatically corrects local clustering errors.

5.1 Introduction

The numerical inverse problem resolution implies the discretization of the initial geometry with shell finite elements, so the computed peening pattern is a discrete map. In the general case, the eigenstrain prescribed to each element may vary from one element to another. On the other hand, the practical conditions limit this variation. Thus, the maximal eigenstrain that can be prescribed to a segment is limited by the equipment capacity. In addition, the peening parameters can not be varied gradually during treatment, so a gradual variation of the eigenstrain in the pattern can not be reproduced. Moreover, each modification of peening parameters during treatment slows down the shaping process, which is especially the case for changing the media [25]. These reasons suggest that the elements must form segments united by the same eigenstrain prescribed, and these segments must be larger than the shot stream width. Furthermore, the segmented peening pattern must involve as few as possible peening regimes while preserving the precision of the final shape. In practice, such segmented patterns are usually applied using masks [12, 89]. The masks are glued to the surface of the component and protect the zones that should not be treated. If the peening equipment admits only uniform treatment of the component during one cycle, then a separate mask is produced for each peening regime involved.

In particular, the above-mentioned limitations on the prescribed eigenstrain are addressed in Ref. [5, 6]. The presented inverse problem solvers operate with one available peening regime, so the peening pattern is essentially a set of treated and untreated segments. This effect is achieved in Ref. [5] by limiting the overall treated area, so that the optimization algorithm is forced to assign only the maximal available eigenstrain to the elements. On the other

hand, the neural network presented in Ref. [6] naturally operates with only one eigenstrain magnitude, which is set during the training phase. Nevertheless, the usage of one available peening regime considerably limits the range of achievable target shapes because the peening regime is the main factor determining the locally induced curvatures.

The inverse problem solver presented in Chapter 4 does not restrict the variation of the eigenstrain assigned to each element. The division of the pattern into segments is done during post-processing with the help of grouping. This approach allows any number of regimes in the pattern, which means that it is flexible in terms of target shapes. However, the original and the grouped patterns induce different final shapes, so the peening regimes have to be thoroughly adjusted to minimize this difference.

The described strategies for segmentation of the peening pattern share a common drawback: they require having pre-defined peening regimes. This suggests that if the current peening regimes do not allow to achieve the target shape according to the simulations, then the optimal regimes must be found by trial and error. Thus, if the segmentation is embedded in the inverse problem solver [5, 6], then each new trial means restarting the inverse problem solver. For instance, the solver presented in Ref. [5] must be restarted with a new maximal eigenstrain. Similarly, the neural network presented in Ref. [6] has to be retrained with another fixed eigenstrain value. Otherwise, if the segmentation is executed after the inverse problem resolution by means of grouping (see Chapter 4), then the grouping algorithm must be relaunched with a new combination of peening regimes, which are defined manually.

Here, we present a numerical strategy for segmenting the peening pattern without pre-defined peening regimes. It is designed as a post-processing tool that complements the inverse problem solver described in Chapter 4. The segmentation strategy is based on the `k`-means clustering [77], which means that the original variation of the eigenstrain in the pattern is preserved as much as possible. This feature reduces the difference between the final shapes provided by the original and the segmented patterns.

5.2 Methodology

We start the Methodology section by examining two common eigenstrain formulations of the peening pattern. Next, we consider the relation between the `k`-means clustering and segmentation of the peening pattern. Subsequently, we examine in detail the two clustering algorithms developed for the two eigenstrain formulations. At the end, we present the filtering algorithm that corrects local clustering errors and thus makes the pattern fully applicable.

5.2.1 The two eigenstrain formulations

The eigenstrain prescribed to each finite element by an eigenstrain-based inverse problem solver is considered constant in the in-plane direction inside the element, but it is assumed to vary along the through-thickness coordinate z. The exact number of variables prescribed to each finite element depends on the adopted formulation of the through-thickness eigenstrain profile $\boldsymbol{\varepsilon}(z)$. To make our formulation clearer, we assume that the treated material is isotropic and not pre-stressed, so that $\varepsilon_{xx}(z) = \varepsilon_{yy}(z) = \gamma(z)$ and $\varepsilon_{zz}(z) = -2\gamma(z)$. The peening-induced isotropic eigenstrain profile $\gamma(z)$ is a continuous function, but it can be idealized, i.e., discretized, in the way that the final shape induced by the idealized profile is the same as that induced by the original profile (see Section 2.1.2). In the general case of treatment from both sides, the simplest idealized profiles involving the fewest variables are the *trilayer* and the *bilayer*.

The trilayer formulation implies the assignment of eigenstrain to the two outer layers of a variable thickness. Under the assumption that the origin of the through-thickness coordinate z is at the level of the mid-surface, the trilayer profile $\gamma_{tri}(z)$ takes the form:

$$\gamma_{tri}(z) = \begin{cases} \varepsilon_*^t & \text{for} \quad \left(\frac{h}{2} - h_*^t\right) < z < \frac{h}{2}, \\ 0 & \text{for} \quad \left(h_*^b - \frac{h}{2}\right) < z < \left(\frac{h}{2} - h_*^t\right), \\ \varepsilon_*^b & \text{for} \quad -\frac{h}{2} < z < \left(h_*^b - \frac{h}{2}\right), \end{cases} \tag{5.1}$$

where the parameters ε_*^t and ε_*^b stand for the eigenstrains assigned to the top and bottom layers respectively, the parameters h_*^t and h_*^b denote the thicknesses of these layers, and h denotes the total plate thickness. Consequently, in the trilayer formulation, the eigenstrain assigned to each element is determined by four variables: $(\varepsilon_*^t, h_*^t, \varepsilon_*^b, h_*^b)$. All four variables are non-negative because the outer layers expand due to peen-forming. In terms of the peening regimes, the regime applied from the top side determines (ε_*^t, h_*^t) and has no influence on (ε_*^b, h_*^b). The same principle applies to the bottom side peening regime, which is related only to (ε_*^b, h_*^b) [25]. We suppose that the plate is fixed flat during treatment, so it does not undergo progressive bending that can influence the induced eigenstrains.

The bilayer eigenstrain profile $\gamma_{bi}(z)$ implies that the plate consists of two equally thick layers assigned with the eigenstrains ε^t and ε^b:

$$\gamma_{bi}(z) = \begin{cases} \varepsilon^t & \text{for} \quad 0 < z < \frac{h}{2}; \\ \varepsilon^b & \text{for} \quad -\frac{h}{2} < z < 0. \end{cases} \tag{5.2}$$

Hence, the equal thickness of both layers allows to describe the bilayer profile with two variables: $(\varepsilon^t, \varepsilon^b)$. In this formulation, *each* of the two peening regimes applied from the top and bottom sides influences both ε^t and ε^b. Consequently, the values of ε^t and ε^b are superpositions of contributions made by the top and the bottom side peening regimes.

5.2.2 `k`-means clustering applied to the peening pattern

In terms of clustering, the variables $(\varepsilon^t_*, h^t_*, \varepsilon^b_*, h^b_*)$ and $(\varepsilon^t, \varepsilon^b)$ can be viewed as coordinates of points in space, where each point corresponds to the peening state of a finite element in the peening pattern. The `k`-means algorithm, in turn, divides the points in space into clusters and computes a centroid for each cluster. In terms of peen forming, a cluster regroups all elements assigned with the same top and bottom peening regimes, and its centroid determines these peening regimes. We developed two separate clustering algorithms for the bilayer and the trilayer formulations, and they both are based on the `k`-means algorithm. Typical of a `k`-means algorithm, our implementation uses a pre-defined number of centroids, or in other words, a pre-defined number of peening regimes.

When the points are clustered, the algorithms homogenize the eigenstrain for all finite elements that fall into the same cluster, so that the eigenstrain is uniform inside each segment. Namely, each element inside a cluster is assigned with the eigenstrain corresponding to the centroid of this cluster. This strategy allows, in particular, to constrain the maximal eigenstrain in the pattern by constraining the coordinates of the centroids. Given that the cluster centroids are related to the peening regimes, the clustering algorithms essentially output top- and bottom-side peening patterns formulated in terms of peening regimes.

The `k`-means clustering is an iterative process involving the relocation of the centroids on each iteration. The result of clustering is dependent on the initial guess of the centroid positions, which are assigned randomly. For this reason, the algorithm must be relaunched several times, each time with different initial positions of the centroids. Each trial, in turn, comprises several iterations. At the end of each trial, the algorithm evaluates the quality of clustering by computing the sum of squared Euclidean distances between the points and the centroids of clusters to which the points are assigned. The trial providing the lowest sum of distances is chosen as the optimal clustering.

5.2.3 Clustering in the trilayer case

In the trilayer formulation, each finite element e is assigned with the variables $(\varepsilon^t_{*e}, h^t_{*e}, \varepsilon^b_{*e}, h^b_{*e})$ by the inverse problem solver. The range of peening regimes that can be

applied from the top and bottom sides is the same, so the eigenstrains assigned to the top and bottom layers are clustered simultaneously. This means that each element generates two points on a coordinate plane with axes ε_* and h_*. The coordinates of these points are $(\varepsilon^t_{*e}, h^t_{*e})$ and $(\varepsilon^b_{*e}, h^b_{*e})$. The points are clustered using the standard `k`-means algorithm [77], except for the fact that the zero centroid $(0, 0)$ corresponding to the absence of treatment is fixed at the origin (Algorithm 3). The presence of this centroid suggests that N peening regimes actually induce $\mathtt{k} = (N + 1)$ cluster centroids. The other N centroids are initialized at random positions in the rectangle limited by the minimal and the maximal coordinates of the generated points:

$$\begin{cases} \min\left(\min\limits_e \varepsilon^t_{*e}, \min\limits_e \varepsilon^b_{*e}\right) < \varepsilon_* < \max\left(\max\limits_e \varepsilon^t_{*e}, \max\limits_e \varepsilon^b_{*e}\right); \\ \min\left(\min\limits_e h^t_{*e}, \min\limits_e h^b_{*e}\right) < h_* < \max\left(\max\limits_e h^t_{*e}, \max\limits_e h^b_{*e}\right). \end{cases} \tag{5.3}$$

Algorithm 3 The clustering algorithm in the trilayer case

for each finite element e **do**
    Initialize two points with coordinates $(\varepsilon^t_{*e}, h^t_{*e})$ and $(\varepsilon^b_{*e}, h^b_{*e})$
end for
Initialize N centroids at random positions in coordinates (ε_*, h_*) under constraints 5.3.
Initialize the zero centroid at the origin
while at least one centroid changes its position **do**
    for each point **do**
        Compute the squared Euclidean distance to each centroid
        Assign the point to the cluster formed by the closest centroid
    end for
    for each cluster except for the zero cluster **do**
        Compute the mean of all points in the cluster
        Assign the mean as the new cluster centroid
    end for
end while
for each finite element e **do**
    Assign the top layer with a couple (ε^t_*, h^t_*) corresponding to the centroid of the cluster containing point $(\varepsilon^t_{*e}, h^t_{*e})$
    Assign the bottom layer with a couple (ε^b_*, h^b_*) corresponding to the centroid of the cluster containing point $(\varepsilon^b_{*e}, h^b_{*e})$.
end for

5.2.4 Clustering in the bilayer case

In the bilayer formulation, the treatment from the top side induces positive ε^t and negative ε^b, while treatment from the bottom side induces negative ε^t and positive ε^b (see Chapter 4). For this reason, we characterize a peening regime with a couple $\left(\varepsilon^{(1)}, \varepsilon^{(2)}\right)$, where, by convention, $\varepsilon^{(1)}$ is positive and $\varepsilon^{(2)}$ is negative. Hence, if a plate is treated from the top side with regime i inducing $\left(\varepsilon_i^{(1)}, \varepsilon_i^{(2)}\right)$ and from the bottom side with regime j inducing $\left(\varepsilon_j^{(1)}, \varepsilon_j^{(2)}\right)$, then their combination results in $\left(\varepsilon_i^{(1)} + \varepsilon_j^{(2)}\right)$ in the top layer and in $\left(\varepsilon_i^{(2)} + \varepsilon_j^{(1)}\right)$ in the bottom layer. An unclustered pattern consists of such eigenstrain *combinations* assigned to each element, but the aim of the clustering algorithm is to find the optimal *regimes*. Consequently, the clustering algorithm must split the combinations into contributions made by the top and bottom peening regimes.

The clustering algorithm starts with initialization of the points that are to be clustered on the coordinate plane $(\varepsilon^t, \varepsilon^b)$. Each finite element e generates one point with coordinates $(\varepsilon_e^t, \varepsilon_e^b)$, where $(\varepsilon_e^t, \varepsilon_e^b)$ is the eigenstrain prescribed to this element by the inverse problem solver. Therefore, in the bilayer formulation, the elements and the points are directly associated. Subsequently, the clustering algorithm initializes the "void" regime $(0, 0)$ corresponding to the absence of treatment, and it also randomly initializes N peening regimes in terms of $\left(\varepsilon^{(1)}, \varepsilon^{(2)}\right)$ under the following constraints:

$$\begin{cases} 0 < \varepsilon^{(1)} < \max\left(\max\limits_e \varepsilon_e^t, \max\limits_e \varepsilon_e^b\right); \\ \min\left(\min\limits_e \varepsilon_e^t, \min\limits_e \varepsilon_e^b\right) < \varepsilon^{(2)} < 0. \end{cases} \tag{5.4}$$

These constraints impose signs to $\varepsilon^{(1)}$ and $\varepsilon^{(2)}$ and set limits based on the eigenstrain range of the given pattern.

Afterwards, the clustering algorithm initializes the cluster centroids, that are computed as all possible combinations of the peening regimes. Hence, there are $\mathtt{k} = (N+1)^2$ centroids initialized, and each of them represents a cluster. A centroid formed by regime i applied from the top side and by regime j applied from the bottom side is marked as ij (Table 5.1). The void regime is numbered as 0, so the centroids marked as $i0$ and $0i$ correspond to treatment with regime i only from the top side and only from the bottom side respectively. Figure 5.1 a) shows the points and the centroids initialized on the plane.

The next stage is deletion of all centroids that lie above the line $\varepsilon^t = \varepsilon^b$. Indeed, the range of available peening regimes is the same for both sides of the plate, but the centroids ij and ji are induced by the same peening regimes applied conversely. This principle also applies to

Table 5.1 Formation of the cluster centroids in the bilayer case. Each centroid represents a combination of peening regimes applied from the top and bottom sides. N available peening regimes form $(N+1)^2$ centroids because the absence of treatment is considered as an additional "void" regime numbered as 0.

Centroid	Peening regime (top)	Peening regime (bottom)
00	0	0
01	0	1
02	0	2
⋮	⋮	⋮
ij	i	j
⋮	⋮	⋮
NN	N	N

the points corresponding to the elements. Thus, the eigenstrains $(\varepsilon_e^t, \varepsilon_e^b)$ and $(\varepsilon_e^b, \varepsilon_e^t)$ represent the same treatment combination applied from different sides. However, the points cannot be deleted like the centroids because all points contain information on the given pattern, so the clustering algorithm reflects all points that lie above the line $\varepsilon^t = \varepsilon^b$ across this line. With this, the points lying on either side of the line $\varepsilon^t = \varepsilon^b$ are fused and clustered simultaneously. Figure 5.1 b) traces the reflected points and the preserved centroids.

Next, the iterative process begins. Same as during the standard `k`-means clustering, each point, i.e., element, is assigned to the cluster represented by the closest centroid, which is shown in Figure 5.1 c). However, the centroids cannot be relocated freely for every iteration because they are interconnected. Indeed, the relocation of centroid ij means altering the eigenstrains induced by regimes i and j. This induces a simultaneous relocation of all centroids formed by regimes i and j in combination with another regime. Hence, the bilayer clustering algorithm does not relocate centroids but adjusts the peening regimes directly. The new centroid positions are then computed as a function of the new peening regimes.

The peening regimes are adjusted using the notion of *contributions*. Let us consider a point $(\varepsilon_e^t, \varepsilon_e^b)$ generated by element e, which is assigned to cluster ij on the current iteration. The centroid of this cluster represents the combination of treatments with regimes i and j. In turn, point e also represents a combination of treatments, but the contributions made by the top and bottom peening are not defined. However, the clustering algorithm derives these contributions, which have coordinates $\left(\varepsilon_{e,i}^{(1)}, \varepsilon_{e,i}^{(2)}\right)$ and $\left(\varepsilon_{e,j}^{(1)}, \varepsilon_{e,j}^{(2)}\right)$, and uses them to adjust regimes i and j, respectively. The contributions are derived in the way that they are as close as possible to the eigenstrains induced by regimes i and j.

We formulate the expression for the coordinates of the contributions using the method of Lagrange multipliers [90]. These coordinates must minimize the sum of squared Euclidean distances f to the regimes i and j:

$$f = \left(\varepsilon_{e,i}^{(1)} - \varepsilon_i^{(1)}\right)^2 + \left(\varepsilon_{e,i}^{(2)} - \varepsilon_i^{(2)}\right)^2 + \left(\varepsilon_{e,j}^{(1)} - \varepsilon_j^{(1)}\right)^2 + \left(\varepsilon_{e,j}^{(2)} - \varepsilon_j^{(2)}\right)^2. \tag{5.5}$$

In addition, the combination of contributions $\left(\varepsilon_{e,i}^{(1)}, \varepsilon_{e,i}^{(2)}\right)$ and $\left(\varepsilon_{e,j}^{(1)}, \varepsilon_{e,j}^{(2)}\right)$ must equal $(\varepsilon_e^t, \varepsilon_e^b)$:

$$\begin{cases} \varepsilon_{e,i}^{(1)} + \varepsilon_{e,j}^{(2)} = \varepsilon_e^t, \\ \varepsilon_{e,i}^{(2)} + \varepsilon_{e,j}^{(1)} = \varepsilon_e^b. \end{cases} \tag{5.6}$$

With this, the Lagrangian function $\mathcal{L}$ takes the form:

$$\begin{aligned} \mathcal{L} = &\left(\varepsilon_{e,i}^{(1)} - \varepsilon_i^{(1)}\right)^2 + \left(\varepsilon_{e,i}^{(2)} - \varepsilon_i^{(2)}\right)^2 + \left(\varepsilon_{e,j}^{(1)} - \varepsilon_j^{(1)}\right)^2 + \left(\varepsilon_{e,j}^{(2)} - \varepsilon_j^{(2)}\right)^2 \\ &- \lambda_1 \left(\varepsilon_{e,i}^{(1)} + \varepsilon_{e,j}^{(2)} - \varepsilon_e^t\right) - \lambda_2 \left(\varepsilon_{e,i}^{(2)} + \varepsilon_{e,j}^{(1)} - \varepsilon_e^b\right). \end{aligned} \tag{5.7}$$

The stationary point of the Lagrangian function is:

$$\begin{cases} \varepsilon_{e,i}^{(1)} = 0.5 \left(\varepsilon_i^{(1)} - \varepsilon_j^{(2)} + \varepsilon_e^t\right), \\ \varepsilon_{e,i}^{(2)} = 0.5 \left(\varepsilon_i^{(2)} - \varepsilon_j^{(1)} + \varepsilon_e^b\right), \\ \varepsilon_{e,j}^{(1)} = 0.5 \left(\varepsilon_j^{(1)} - \varepsilon_i^{(2)} + \varepsilon_e^b\right), \\ \varepsilon_{e,j}^{(2)} = 0.5 \left(\varepsilon_j^{(2)} - \varepsilon_i^{(1)} + \varepsilon_e^t\right), \end{cases} \tag{5.8}$$

$$\begin{cases} \lambda_1 = \varepsilon_e^t - \varepsilon_i^{(1)} - \varepsilon_j^{(2)}, \\ \lambda_2 = \varepsilon_e^b - \varepsilon_j^{(1)} - \varepsilon_i^{(2)}. \end{cases} \tag{5.9}$$

The two contributions computed according to Eqn. (5.8) minimize the sum of squared Euclidean distances (5.5) and satisfy conditions (5.6), simultaneously.

Let $\mathcal{T}_i$ be the set of points assigned to all the centroids formed using regime i on the current iteration. The algorithm computes the new eigenstrain induced by regime i as the mean of all contributions $\left(\varepsilon_{e,i}^{(1)}, \varepsilon_{e,i}^{(2)}\right)$ for $e \in \mathcal{T}_i$. The "void" regime is, however, fixed at point $(0, 0)$. When all regimes are adjusted, the algorithm recalculates the centroids and reassigns the points to the clusters. Same as with the standard `k`-means algorithm, this iterative process is repeated until stabilization of the peening regimes and, consequently, of the centroids. The clustering algorithm in the bilayer case is summarized as Algorithm 4. Figure 5.1 d) illustrates the result of clustering.

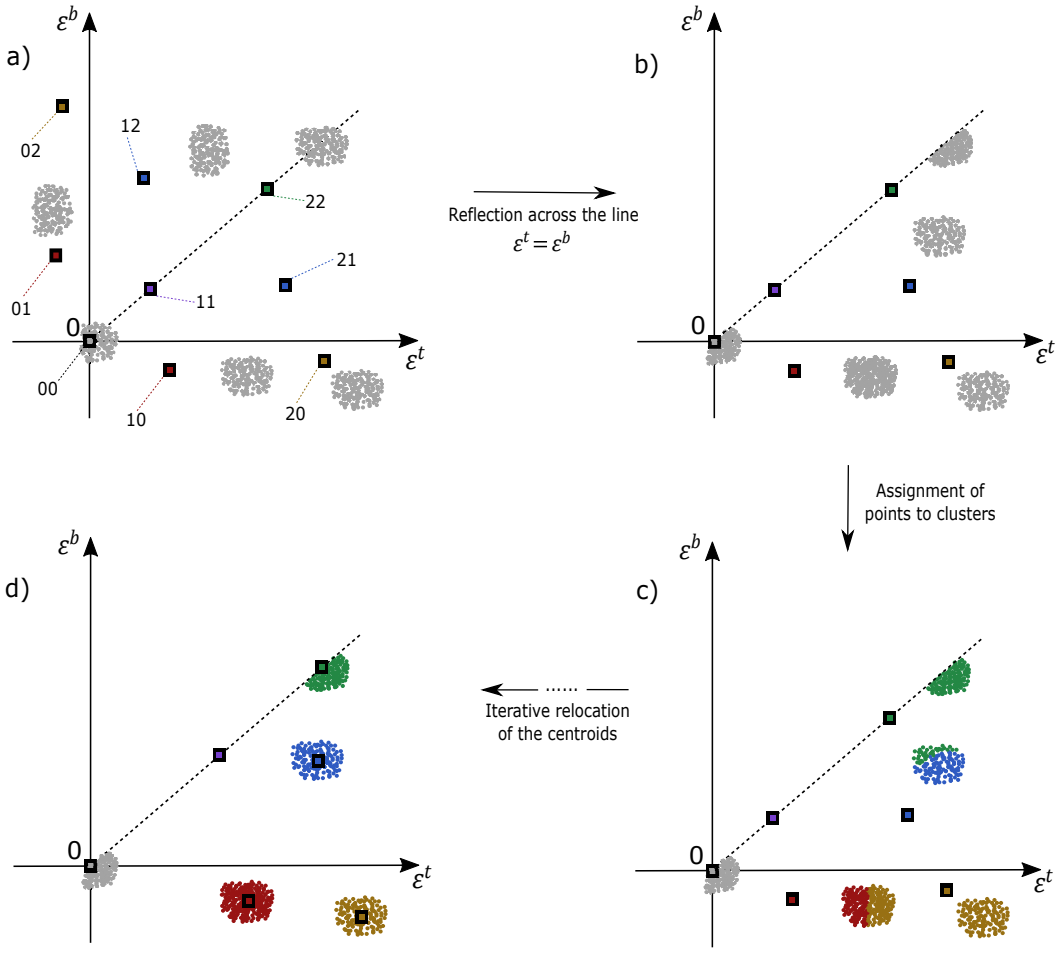

Figure 5.1 Example of clustering in the bilayer case with $N = 2$ available peening regimes.
a) At the beginning, the clustering algorithm traces the points (grey dots) that are to be clustered at coordinates $(\varepsilon^t, \varepsilon^b)$. Each point represents a finite element, and its coordinates reflect the eigenstrain assigned to the element by the inverse problem solver. The clustering algorithm randomly initializes the peening regimes and traces the cluster centroids (colored squares), each of which is a combination of two peening regimes for all possible permutations. The coordinates of centroid ij represent the eigenstrain induced by the application of regime i from the top side and of regime j from the bottom side.
b) Next, the algorithm eliminates the symmetry by reflecting the points that lie above the line $\varepsilon^t = \varepsilon^b$ across this line. The centroids that lie above this line are deleted.
c) Subsequently, each point is assigned to the closest centroid (cluster) in terms of squared Euclidean distance.
d) The algorithm iteratively adjusts the peening regimes, derives new centroid positions and reassigns the points until the stabilization of the centroids. The centroid $(0, 0)$ is fixed at the origin during all iterations. In this example, no points were attributed to cluster 11 (purple), which corresponds to the treatment with regime 1 from both sides.

Algorithm 4 The clustering algorithm in the bilayer case

for each finite element e **do**
    Initialize one point with coordinates $(\varepsilon_e^t, \varepsilon_e^b)$
end for
Randomly initialize N peening regimes in terms of $\left(\varepsilon^{(1)}, \varepsilon^{(2)}\right)$ under constraints 5.4
Initialize the "void" regime $(0, 0)$
Compute the centroids in coordinates $(\varepsilon^t, \varepsilon^b)$ as all possible combinations of two peening regimes
for each centroid that lies above the line $\varepsilon^t = \varepsilon^b$ **do**
    Delete the centroid
end for
for each point that lies above the line $\varepsilon^t = \varepsilon^b$ **do**
    Swap its coordinates ε^t and ε^b
end for
while at least one centroid changes its position **do**
    for each point **do**
        Compute the squared Euclidean distance to each centroid
        Assign the point to the cluster formed by the closest centroid
        Compute the two contributions using Eqn. (5.8)
    end for
    for each regime except for the "void" regime **do**
        Compute the mean of all contributions belonging to this regime
        Assign the mean as the new peening regime
    end for
    Compute the centroids as the combinations of the adjusted peening regimes
end while
for each finite element e **do**
    Assign the element with the eigenstrain corresponding to the centroid of the cluster containing point $(\varepsilon_e^t, \varepsilon_e^b)$
end for

5.2.5 Filtering of the peening pattern using cellular automata

The two clustering algorithms group the finite elements based on the eigenstrain prescribed during the inverse problem solution. After clustering, each finite element in the peening pattern is assigned with a peening regime. The elements assigned with the same regime can be physically situated in different parts of the plate and thus form separate segments. In practice, however, the minimal size of the segments is limited by the shot stream width or the masking resolution. For example, a segment consisting only on one element is not typically reproducible in practice. The presence of such segments in the clustered pattern is called the *checkerboard* problem, which is illustrated in Figure 5.2. This problem arises when the eigenstrain prescribed by the inverse problem solver is on the border between two clusters, so the elements assigned with different peening regimes are mixed together after clustering.

We cope with the checkerboard problem with a filtering algorithm, which is executed after clustering for the top and bottom patterns separately. More precisely, we use an image noise filter based on cellular automata (CA) described in Ref. [91]. The filter is initially conceived for images consisting of square pixels that are assigned with a gray level, but we reformulate it for the case of triangular meshes where each element is assigned with a peening regime. The filtering algorithm scans the elements and notes the regimes assigned to each element and to its three neighbors, i.e., to its triangular Von Neumann neighborhood [92]. If the algorithm detects that the element is mostly surrounded by elements prescribed with another regime, then it assigns the element with the regime that constitutes the majority. In case when none of the peening regimes constitutes a majority in the local neighborhood, then the element is

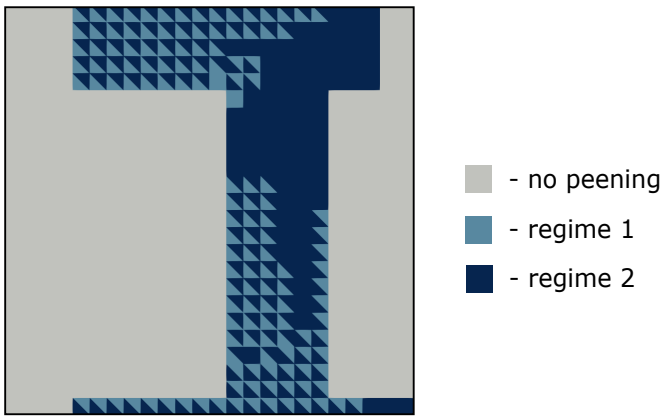

Figure 5.2 Example of a peening pattern heavily affected by the checkerboard problem after clustering. The pattern is traced on a square plate, which is meshed with triangular finite elements. The elements assigned with different peening regimes of a close intensity are locally mixed, so the proposed pattern is not reproducible in practice.

assigned with the regime corresponding to its first neighbor. The same principle applies to the edge elements that have only two neighbors. The corner elements are strictly assigned with the regime corresponding to their single neighbor. The filtering algorithm is iterative and lasts until the stabilization of the assigned regimes. All triangular elements are scanned and corrected simultaneously at each iteration. This filter can also be applied after the grouping of the peening pattern (Chapter 4), because the grouped patterns can also face the checkerboard problem.

The filtering algorithm is summarized as Algorithm 5. We use a general notation and formulate the algorithm in terms of the peening regime ρ_e assigned to element e. The regimes assigned to the neigboring elements are denoted as ρ_{e1}, ρ_{e2}, ρ_{e3}.

5.3 Results

The clustering algorithm for the bilayer case and the filtering algorithm were validated numerically using 200 random target shapes. For each test case, we firstly solved the inverse problem using the algorithm presented in Chapter 4 and obtained the *free* peening pattern. Secondly, we applied the clustering algorithm (Algorithm 4) and thus obtained the *clustered* peening pattern. Thirdly, we applied the filtering algorithm (Algorithm 5) and obtained the *filtered clustered* peening pattern. Finally, we simulated the application of each of these three patterns using the method presented in Chapter 4 to evaluate the induced error. In other words, we solved the forward problem for each pattern and thus obtained three final shapes. We computed the dimensionless error Ω in each case as the Hausdorff distance between the final shape and the target shape divided by the total area of the plate.

The target shapes were generated as a result of the application of random peening patterns on a 1×1 m square plate. The plate thickness and the Poisson's ratio were randomly assigned in each test case. The plates were meshed with 1152 triangular elements, and the random peening patterns were generated on this mesh using the algorithm presented in Chapter 4. The first 100 test cases (series 1) involved only one available peening regime, which was applied in randomly situated rectangular segments of the plate. The second 100 cases (series 2) involved four available regimes applied in random segments.

Figure 5.3 illustrates the numerical validation workflow using one test case from series 2. The presented clustered pattern (Figure 5.3 d) leads to the highest Ω in series 2 because the pattern is significantly affected by the checkerboard problem. The filtered clustered pattern (Figure 5.3 e) contains irregularly shaped peening segments, but it is not subjected to the checkerboard problem, so it is applicable in practice.

Algorithm 5 The filtering algorithm

```
1: while the eigenstrain is reassigned at least for one element do
2:     for each finite element e do
3:         if the element has three neighbors then
4:             if ρ_e ≠ ρ_e1 and (ρ_e1 = ρ_e2 or ρ_e1 = ρ_e3) then
5:                 Assign the value of ρ_e1 to ρ_e
6:             else if ρ_e ≠ ρ_e2 and ρ_e2 = ρ_e3 then
7:                 Assign the value of ρ_e2 to ρ_e
8:             else if
9:                 (ρ_e ≠ ρ_e1 and ρ_e ≠ ρ_e2 and ρ_e ≠ ρ_e3) and
10:                (ρ_e1 ≠ ρ_e2 and ρ_e1 ≠ ρ_e3 and ρ_e2 ≠ ρ_e3)  then
11:                Assign the value of ρ_e1 to ρ_e
12:            end if
13:        end if
14:        if the element has two neighbors then
15:            if ρ_e ≠ ρ_e1 and ρ_e ≠ ρ_e2 then
16:                Assign the value of ρ_e1 to ρ_e
17:            end if
18:        end if
19:        if the element has one neighbor then
20:            if ρ_e ≠ ρ_e1 then
21:                Assign the value of ρ_e1 to ρ_e
22:            end if
23:        end if
24:    end for
25: end while
```

Figure 5.4 summarizes the validation results in terms of Ω. The efficiency of the inverse problem resolution with a free pattern was earlier discussed in Chapter 4. The histograms demonstrate that both clustering and filtering algorithms had a slight influence in terms of Ω. Nevertheless, Ω stayed inferior to 0.4 for all test cases. Moreover, Ω stayed inferior to 0.1 for more than 80% of cases in series 1 and for more than 70% of cases in series 2.

During the clustering phase, we fixed the number of available regimes as one for series 1

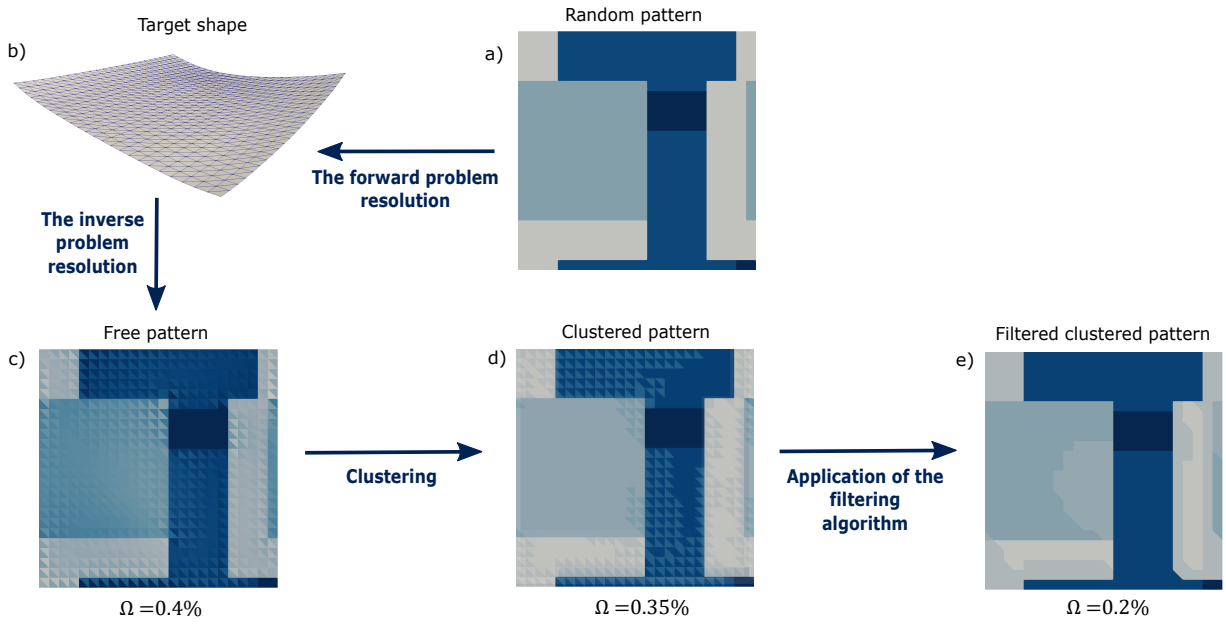

Figure 5.3 One of the test cases from series 2 that demonstrated a marked checkerboard problem. a) The random peening pattern generated on a 1×1 m sized plate using the algorithm presented in Chapter 4. Different colors on the peening pattern denote different peening regimes. b) The target shape generated by application of the random pattern. The plate is 6 mm thick, and the Poisson's ratio equals 0.36. The deflections are at the original scale. c) The free pattern computed using the inverse problem resolution algorithm (Chapter 4). d) The clustered peening pattern computed by application of Algorithm 4 to the free pattern. e) The filtered clustered algorithm obtained by application of Algorithm 5 to the clustered pattern. We solved the forward problem for the patterns c), d) and e) and computed the dimensionless error Ω for each case.

and as four for series 2. No other information on the peening regimes was provided to the clustering algorithm, and the centroid positions were not constrained. With this, the clustered pattern decreased Ω with respect to the free pattern for 78 cases in series 1 and for 32 cases in series 2. This difference is explained by the fact the clustered patterns in series 1 were less subjected to the checkerboard problem. Indeed, the only available peening regime was situated far from the origin in terms of $(\varepsilon^t, \varepsilon^b)$, so the majority of elements was assigned to correct clusters. The main source of error in this case was the peening regime itself, which was slightly different from the peening regime applied in the original random pattern. On the other hand, the regimes in series 2 were situated close to each other, so the quantity of neighboring elements that were assigned to different clusters was more significant. Also, the four computed peening regimes did not exactly correspond to the original regimes.

It should be noted that although we fixed the quantity of peening regimes N for each series of tests, in practice N is not always pre-determined. In a such case, the clustering algorithm

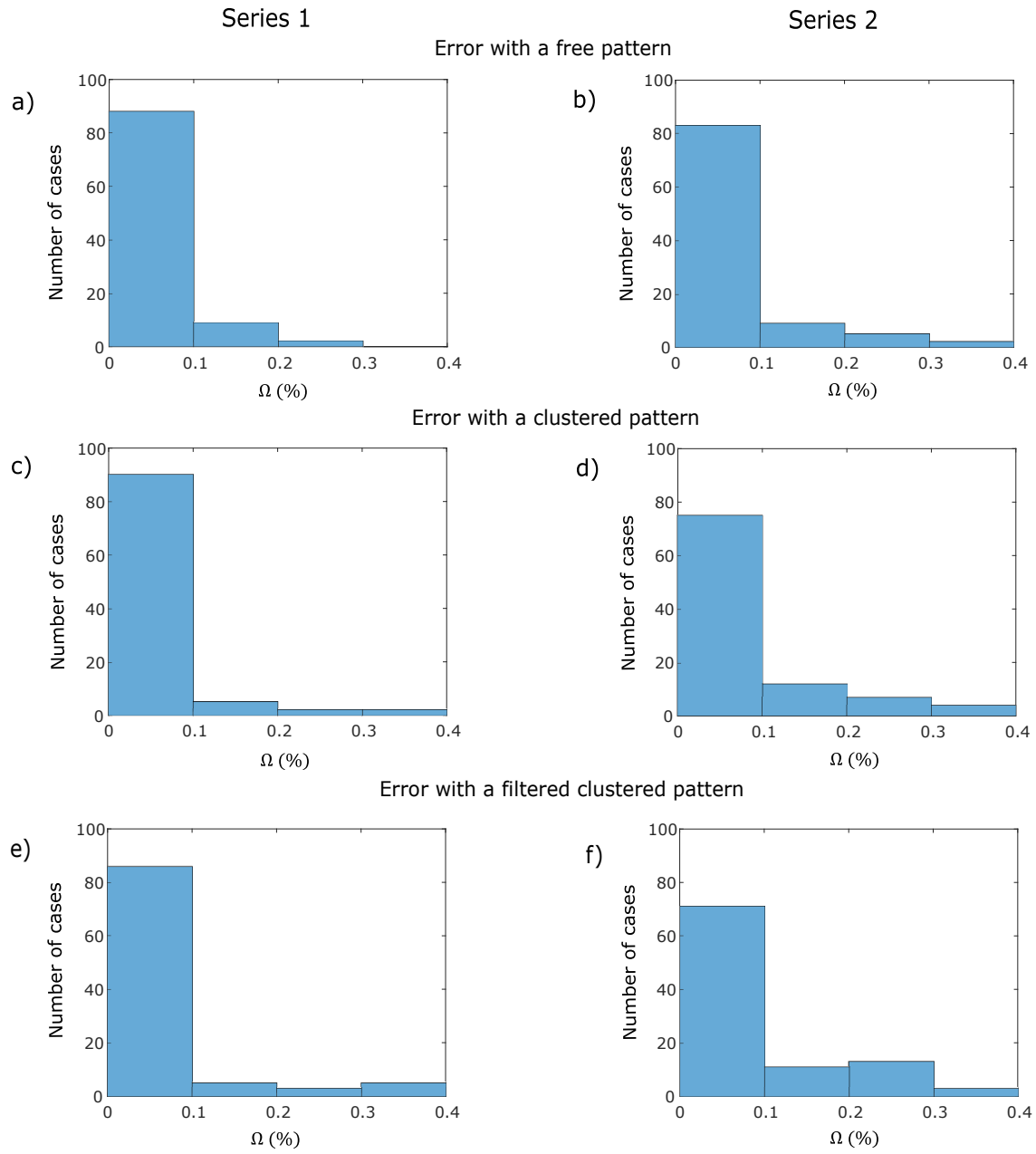

Figure 5.4 Histograms tracing the dimensionless error Ω between the target shapes and the final shapes computed during validation of the clustering and filtering algorithms. We considered 200 test cases that were divided in two series of 100 cases each. In each case, we solved the inverse problem, clustered the peening pattern and applied the filtering algorithm. One peening regime was available for clustering in series 1, and four regimes were available in series 2. We simulated application of the peening patterns on each stage and computed the dimensionless error Ω. Figures (a) and (b) trace the simulated Ω provided by the unclustered free patterns in series 1 and 2 respectively. Figures (c) and (d) trace the simulated Ω provided by the clustered patterns, and figures (e) and (f) show Ω provided by the filtered clustered patterns.

can be launched several times for different N, and the best suitable N can be determined based on the minimal Ω. A larger N provides more flexibility in terms of target shapes but extends the time needed for application of the pattern.

The filtered clustered pattern decreased Ω with respect to the clustered pattern for 23 cases in series 1 and for 33 cases in series 2. The reason for an increase in Ω in most of the cases is that the filtering algorithm often enlarges the segments assigned to wrong clusters instead of eliminating them. This is explained by the fact that the algorithm does not measure Ω, so its output depends only on the free pattern. Consequently, the usage of this algorithm must be limited only to the patterns that are significantly affected by the checkerboard problem because there is a probability of a slight increase in Ω. If a clustered pattern contains only several elements assigned to wrong clusters, then a manual reassignment of the eigenstrain prescribed to these elements is more efficient.

5.4 Conclusion

We developed a general strategy for the segmentation of the peening pattern, which makes every pattern applicable with shot peening equipment. The segmentation strategy computes optimal peening regimes and the segments where they should be applied, so the need for trial and error determination of the optimal regimes is eliminated. The strategy consists in a clustering and a filtering algorithms. These algorithms are conceived for the case of patterns formulated in terms of eigenstrain. Two separate clustering algorithms based on the `k`-means method were developed for the trilayer and the bilayer eigenstrain formulations. When applied consecutively, the clustering and the filtering algorithms remove from the pattern all eigenstrain variations that are not reproducible in practice. Hence, the strategy provides a ready-to use peening pattern, which is suitable for industrial reproduction using masking.

The bilayer clustering algorithm and the filtering algorithm were validated numerically using the existing bilayer shell model. The validation showed that the clustering algorithm had a weak influence on the quality of the peening pattern, so that the original and the clustered patterns induced similar final shapes. The clustered pattern is less affected by the checkerboard problem if the number of peening regimes is low, but a large number of regimes provides more flexibility in terms of target shapes. The filtering also has a slight but majorly negative influence on the final shape. However, it efficiently deals with the checkerboard problem and rapidly eliminates local clustering errors. Numerical validation of the trilayer clustering algorithm is left for future studies.

CHAPTER 6 ARTICLE 2: SIMULATION AND AUTOMATION OF ALUMINUM PANEL SHOT PEEN FORMING

Vladislav Sushitskii, Pierre-Olivier Dubois, Hong Yan Miao, Martin Lévesque, Frédérick P. Gosselin

Submitted to the International Journal of Material Forming on the 27th of January 2022.

In this article, we present the automated shot peen forming workflow. It was initially designed for the purpose of experimental validation of the simulation software presented in Chapter 4. The workflow regroups all methods that are used to apply a peening pattern, which is provided by the simulation software, in practice. An important part of the workflow, namely, the software that translates the peening pattern into a program for the peening robot, was designed and implemented by Pierre-Olivier Dubois[1]*. The practical manipulations related to the forward problem solver validation were done by the author of this thesis and by Raphaël Paradis*[2] *during their internships at the Aerospace Technology Center (CTA). Raphaël Paradis also designed and fabricated the fixture for clamping of the panels during treatment. The practical manipulations related to the inverse problem solver validation were done by Nada Abbassi*[3]*. An assistance in fabrication of the demonstrator was also provided by Olivier Duchesne*[4]*.*

Abstract

We present a methodology for automated forming of metal plates into freeform shapes using shot peening. The methodology is based on a simulation software that computes the peening pattern and simulates the effect of its application. The pattern generation requires preliminary experimental characterization of the treatment. The treatment is applied by a shot peening robot. The program for the robot is generated automatically according to the peening pattern. We validate the methodology with a series of tests. Namely, we form nine aluminum plates into doubly curved shapes and we also shape model airplane wing skins. The article describes the complete workflow and the experimental results.

[1]Leader of the robotics team, Aerospace Technology Center (CTA)

[2]Intern, Aerospace Technology Center (CTA)

[3]Intern, Laboratory for Multiscale Mechanics (LM2), Department of Mechanical Engineering, Polytechnique Montreal

[4]PhD student, Laboratory for Multiscale Mechanics (LM2), Department of Mechanical Engineering, Polytechnique Montreal

6.1 Introduction

Shot peen forming (peen forming) is an industrial process for forming metal plates into smoothly curved shapes. It consists in projecting a stream of rigid particles (shot) towards a metal plate surface at high velocity. Numerous overlapping impacts induce plastic expansion of the outer layer of material, which causes bending of the plate in the treated segment. The set of treated segments over the flat plate surface is called the peening pattern. Altering the pattern shape or the treatment effectiveness alters the deformed shape as well. This versatility makes peen forming an efficient instrument for achieving complex curvature profiles on large plates, such as airplane wing skins [1] or rocket tank bulkheads [93]. In contrast to stretch forming, which is often used for forming of such components, shot peen forming requires no punch matrices, dies or intermediate heat treatments [94, 95]. In addition, shot peen forming allows to shape composite aluminum panels that constitute fuselage skins [96].

Nowadays, the market proposes a wide range of programmable peen forming robots, such as those mentioned in Refs. [10] and [97]. Nevertheless, there exists no publicly accessible numerical tool that would automatically control a peening robot, bridging the gap between the numerical model (CAD) and the robot path. Such a tool must generate an optimal peening pattern for a given target shape, i.e., solve the *inverse problem*, then translate the pattern into a program for the peening robot, and simulate the effect of its application, i.e., solve the *forward problem*. At least one commercial company claims to have developed an appropriate tool, but the adopted approach is confidential [10, 79]. Smaller companies thus use manual peen forming, which is cheaper but less efficient. It is a trial-and-error process that provides lower repeatability than the automated one. The quality of forming in this case relies completely on the operator's experience, which involves costly training. Moreover, it can lead to chronic health problems for the operator [79].

In the matter of peen forming simulation, a straightforward modelling of every impact would be computationally unachievable for the case of large industrial parts [22]. However, the simulation can be accelerated by means of the *eigenstrain* representation of the peening-induced loads [28]. In the peen forming case, it implies formulating the loads as an anelastic strain (eigenstrain) introduced in the treated segments analogous to thermal expansion, so that the individual impacts are not simulated [2, 23]. The deformation of the plate caused by the eigenstrain is described as the elastic material response, i.e., the springback [44]. It can be computed using the shell finite element method [2, 6]. This method can also be reformulated in terms of the theory of non-Euclidean plates [59], which is specifically designed to describe the reconfiguration of plates subjected to eigenstrain. The eigenstrain approach together with the theory of non-Euclidean plates provide a framework for the inverse problem

resolution [98].

In terms of the eigenstrain approach, the peening pattern is described by the eigenstrain magnitude and its distribution over the surface of the plate. However, from a practical point of view, the pattern is described by the shot stream parameters and by the shape of the treated segment. The eigenstrain magnitude can be related with the shot stream parameters through direct impact simulation on a small representative volume of material [22, 40]. This requires building a specific simulation model that reflects plastification of material induced by the impacts. Otherwise, the eigenstrain can be quantified experimentally using small specimens made of the same material as the plates subjected to forming. The specimens must be uniformly treated with the same parameters that are applied for treatment of the plates. The peening-induced plastic strain, i.e., eigenstrain, is either measured directly using X-ray diffraction [41, 42], or computed based on the residual stress measurements [2]. These strategies require costly measuring equipment and can be replaced by an eigenstrain-based simulation software. Indeed, in such case the eigenstrain is deduced from the deflection of a uniformly treated specimen [29, 99]. More precisely, the eigenstrain is adjusted in the way that the simulated deflection corresponds to the measured one. Although this method does not allow reconstructing the full residual stress profile, it allows quantifying the macroscopic bending and stretching induced by the treatment, and since it does not require any specific equipment or numerical models, it can be efficiently applied in industry.

The eigenstrain induced in an aluminum plate by a perfectly symmetrical impact can be anisotropic [34]. The main causes for this effect are the material plastic anisotropy [100, 101] and the initial stresses induced by the fabrication process [25, 35]. Another potential cause is the prestress, i.e., pre-bending of the plate with clamping supports during peening, which is not present if the plate is held flat during treatment. The influence of the three causes can be quantified experimentally using an additional equipment [35]. Otherwise, the eigenstrain anisotropy can be characterized using an eigenstrain-based simulation software, so that the anisotropy is numerically adjusted to fit the simulations to the experiments. In terms of this method, a particular challenge lies in the fact that the influence of the eigenstrain anisotropy on the final shape depends on the geometry of the plate [35]. Consequently, the characterization should ideally be done using exactly the same plates as those that are supposed to be shaped afterwards.

Precise reproduction of the peening pattern shape is crucial for the process efficiency. One possible approach implies masking of the untreated segments with subsequent uniform treatment of the plate. The masks are glued to the surface of the plate to absorb the energy of the shot and to protect the material underneath. A thick adhesive tape or cardstock cope

well with this task. The masks are shaped either manually or using a laser cutter and are manually glued to the component [5, 25]. This approach is cumbersome when the applied pattern changes from one component to another, because a different mask needs to be fabricated for each pattern. For this reason, an automated peen forming system would require a different approach to remove this manual and time consuming component of the workflow.

In the existing literature on shot peen forming, the experimental validation of the numerical forward problem resolution is mostly limited to uniform treatment of small coupons [29, 38]. Such experiments do not require an automated peen forming system due to the simplicity of the applied pattern. Moreover, the existing literature on the numerical inverse problem resolution in this domain is mostly limited to the description of the developed algorithms [3] or to their numerical validation [6, 98]. Only Faucheux and Miao et al. have validated their inverse problem resolution strategy experimentally on complex shapes such as a cylinder, a saddle and a wave [5, 25]. The peening patterns in this research were applied using masking, so the overall peen forming workflow is not completely automated. Moreover, the developed deflection was of the order of millimeters for 1×1 m panels, which is too small to reveal nonlinear geometric effects appearing with large deflections.

Here, we present a fully automated peen forming workflow implying a programmable peening robot. We describe the numerical tools, the steps for their calibration and the specifically designed installations for the robot. The numerical tools include the simulation software and the translation software. The former solves both the forward and the inverse problems and allows to quantify the eigenstrain anisotropy. The latter automatically converts the peening pattern into the nozzle paths. Hence, the shot stream “paints" the peening pattern on the plate, so the necessity for masking is eliminated. Together, the numerical tools form a “CAD-to-path" solution allowing the user to input a desired shape and to generate the robot program with a minimal intervention.

The automated workflow was developed for experimental validation of the numerical forward and inverse problem solvers described in Ref. [98]. The forward problem solver validation involves randomly generated and strongly curved target shapes that have deflection of $10 - 31$ mm. The inverse problem solver validation consists in shaping parts of a model airplane wing skin of a predefined geometry.

6.2 Methodology

The methodology section begins with a description of the underlying principles of our simulation software. Next, we present the translation software and the peen forming robot. We

then examine our strategy for the estimation of the experimental error implying 3D scanning. Subsequently, we describe the experimental steps for the software calibration. They include characterization of the shot stream, determination of the eigenstrain induced by a given treatment and quantification of the eigenstrain anisotropy on a given component. All the elements mentioned above constitute together the automated peen forming workflow, which is schematized in Figure 6.1. The end of the methodology section is dedicated to the experimental validation of the forward and the inverse problem solvers.

6.2.1 The simulation software

Representation of peening-induced loads with eigenstrain

Following Ref. [98], we virtually divide the plate into two layers of equal thickness to simulate the curving effect of peen forming. We assign in-plane swelling to one layer and in-plane shrinking to the other in the spots where the plate is treated. The swelling layer is located on the side that undergoes treatment. The swelling and shrinking are assigned in the form of eigenstrain and are analogous to thermal expansions [2]. The eigenstrain magnitude is constant in the through-thickness direction for each layer but varies along the plate surface according to the peening pattern. We denote the eigenstrain tensors imposed on the top and bottom layers as $\boldsymbol{\varepsilon}^t$ and $\boldsymbol{\varepsilon}^b$ respectively. The incompatibility between $\boldsymbol{\varepsilon}^t$ and $\boldsymbol{\varepsilon}^b$ forces the plate to bend.

We endow the component with Lagrangian coordinates x, y, z, where the $x-$ and $y-$axes follow the plate mid-surface. Both eigenstrain tensors in these coordinates have the following form [25]:

$$\boldsymbol{\varepsilon}^j = \begin{pmatrix} \varepsilon^j_{xx} & 0 & 0 \\ 0 & \varepsilon^j_{yy} & 0 \\ 0 & 0 & -(\varepsilon^j_{xx} + \varepsilon^j_{yy}) \end{pmatrix} \quad \text{for} \quad j = t, b. \tag{6.1}$$

For a plate with uniform properties and initial stresses, the proportion between ε^j_{xx} and ε^j_{yy} is constant along the whole component. It is expressed with the *anisotropy coefficient* χ [35]:

$$\begin{cases} \varepsilon^j_{xx} = (1+\chi)\varepsilon^j, \\ \varepsilon^j_{yy} = (1-\chi)\varepsilon^j, \end{cases} \quad \text{for} \quad j = t, b, \tag{6.2}$$

where $\varepsilon^j = (\varepsilon^j_{xx} + \varepsilon^j_{yy})/2$ [5]. If the material is plastically isotropic and free of stresses then

[5] In the original version of the paper submitted to the International Journal of Material Forming on the 27$^{\text{th}}$ of January 2022, ε^j is denoted as ε^j_*. The correction was made in order to ensure the uniformity of notations throughout the thesis.

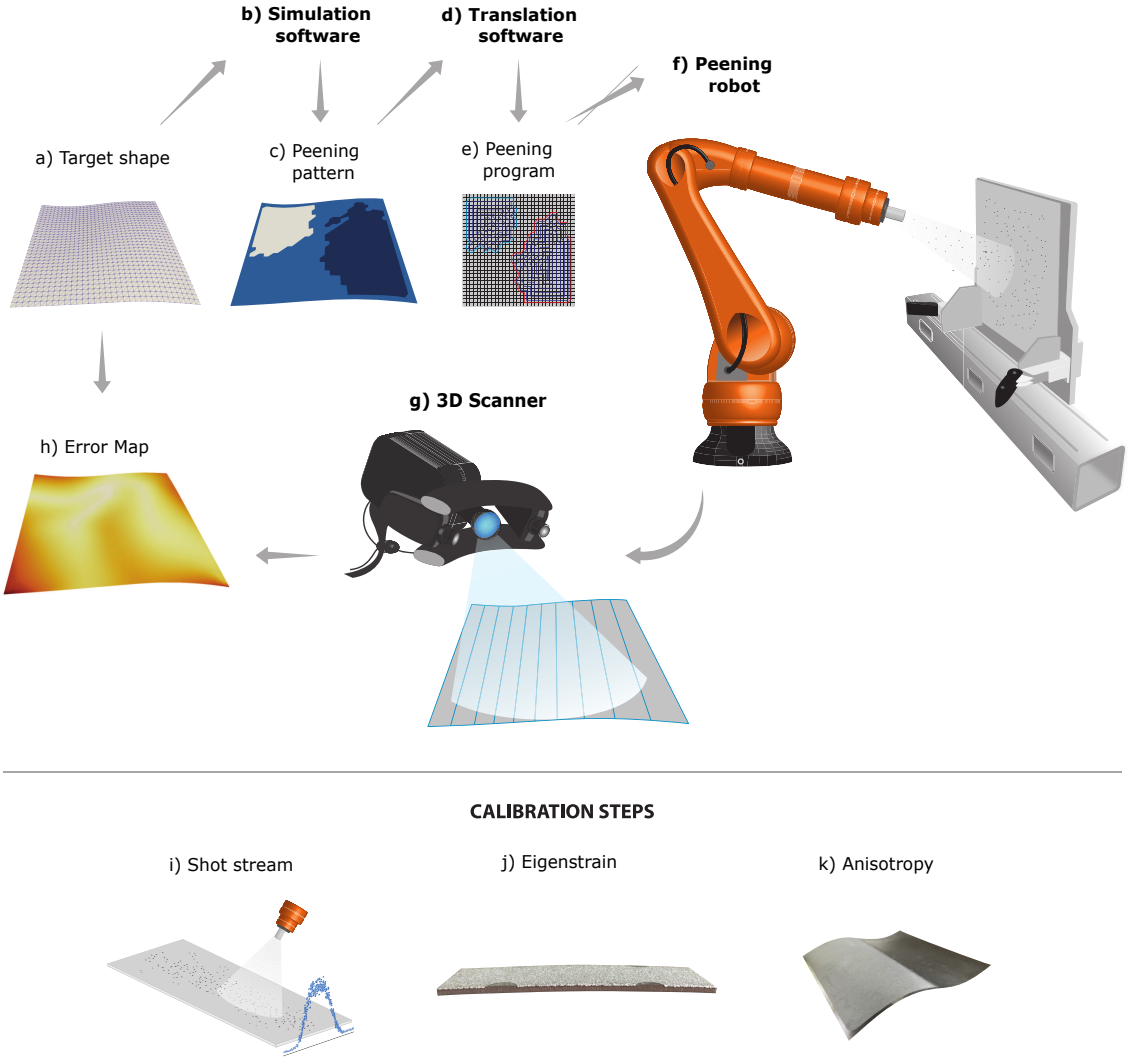

Figure 6.1 Schematic overview of the automated shot peen forming workflow. The numerical model of the target shape (a) is input into the simulation software (b). The latter computes an optimal peening pattern (c), that must be applied to form the plate into the target shape. The peening pattern is input into the translation software (d), which generates the peening program (e). The peening program includes the nozzle paths and a set of peening parameters that allow the peening robot (f) to reproduce the peening pattern. The peening program is loaded into the peening robot in the form of a programming code. When the treatment is finished, we characterize the curved shape of the plate using a 3D scanner (g) and trace an error map (h), which shows discrepancy between the target and the scanned shapes. The simulation and the translation softwares require experimental calibration, which consists in characterizations of the shot stream (i), of the eigenstrain (j) induced by the peening treatment, and of the eigenstrain anisotropy (k). The shot stream is characterized with flat dummy specimens, the eigenstrain is measured using 76×19 mm aluminum coupons, and the anisotropy is quantified with the help of square aluminum plates. Art by IMPAKT Scientifik.

$\varepsilon^j_{xx} = \varepsilon^j_{yy} = \varepsilon^j$.

Geometrical shape description

Provided that peen forming deals with thin plates, we associate the plate shape with the shape of its mid-surface. We numerically discretize the mid-surface with triangular elements. Following Ref. [7], we describe the shape of each element with two matrices: **a** and **b**. The former represents the local first fundamental form and the latter represents the local second fundamental form, that respectively characterize the local stretching and the local curvature [55]. The set of fundamental forms computed for each element uniquely describes the shape of a triangulated surface. For a triangular element defined by vertices $\vec{v}_0$, $\vec{v}_1$, $\vec{v}_2$ and edges $\vec{e}_0 = \vec{v}_1 - \vec{v}_0, \vec{e}_1 = \vec{v}_2 - \vec{v}_1$, $\vec{e}_2 = \vec{v}_0 - \vec{v}_2$ the two matrices are computes as:

$$\mathbf{a} = \begin{bmatrix} \vec{e}_1 \cdot \vec{e}_1 & \vec{e}_1 \cdot \vec{e}_2 \\ \vec{e}_2 \cdot \vec{e}_1 & \vec{e}_2 \cdot \vec{e}_2 \end{bmatrix}; \tag{6.3}$$

$$\mathbf{b} = \begin{bmatrix} \vec{e}_1 \cdot 2(\vec{n}_0 - \vec{n}_2) & -\vec{e}_1 \cdot \vec{n}_0 \\ -\vec{e}_1 \cdot \vec{n}_0 & \vec{e}_2 \cdot 2(\vec{n}_1 - \vec{n}_0) \end{bmatrix}. \tag{6.4}$$

Here, $\vec{n}_i$ stands for a vector normal to the edge $\vec{e}_i, i = 1, 2, 3$. The angle of inclination of $\vec{n}_i$ with respect to the plane enclosing the triangular element is variable and thus allows a more flexible shape description. Figure 6.2 illustrates the components of a triangular element that participate in computation of the fundamental forms.

The local principal curvatures can be computed for each element as the eigenvalues of the shape operator $\boldsymbol{S} = \mathbf{a}^{-1}\mathbf{b}$ [102, 103]. The local mean curvature H is the average of the two principal curvatures, and the curvature along any direction may be deduced from the principal curvatures using Euler's formula [102].

The forward problem resolution

In terms of peen forming, the forward problem consists in the determination of the deformed shape of the component provided with the eigenstrain pattern. For this, we rely on the approach presented in Ref. [7]. We assume that the plate is free to deform and that the treatment is applied instantly from both sides. We also assume that the initial configuration is flat, and we denote the initial lengths of the element edges as $\vec{e}_0$, $\vec{e}_1$, $\vec{e}_2$. We consider the eigenstrain constant inside a triangular element. An anisotropic eigenstrain $(\varepsilon_{xx}, \varepsilon_{yy})$

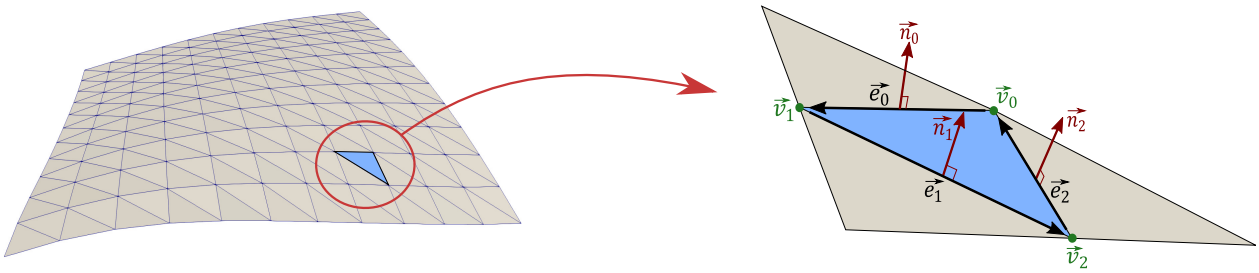

Figure 6.2 A triangular mesh element and the vectors serving for computation of the local fundamental forms. The vectors $\vec{v}_0$, $\vec{v}_1$, $\vec{v}_2$ define positions of the element vertices in the global coordinate system. The edge vectors $\vec{e}_0$, $\vec{e}_1$, $\vec{e}_2$ are computed as the difference between the corresponding vertex vectors. The vectors $\vec{n}_0$, $\vec{n}_1$, $\vec{n}_2$ are traced in the center of the corresponding edges and define the local curvature of the mesh. They are normal to the edges, but their angle of inclination with respect to the element constitutes a degree of freedom.

prescribes new *rest* edges $\vec{e}^{\,r}_0$, $\vec{e}^{\,r}_1$, $\vec{e}^{\,r}_2$:

$$\vec{e}^{\,r}_i = (\varepsilon_{xx} + 1)\vec{e}_{ix} + (\varepsilon_{yy} + 1)\vec{e}_{iy} \quad \text{for} \quad i = 1, 2, 3. \tag{6.5}$$

Here, $\vec{e}_{ix}$ is the projection of $\vec{e}_i$ on the x-axis and $\vec{e}_{iy}$ is the projection of $\vec{e}_i$ on the $y-$axis. New edges induce a new *rest* first fundamental form $\mathbf{a}_r$:

$$\mathbf{a}_r = \begin{bmatrix} \vec{e}^{\,r}_1 \cdot \vec{e}^{\,r}_1 & \vec{e}^{\,r}_1 \cdot \vec{e}^{\,r}_2 \\ \vec{e}^{\,r}_2 \cdot \vec{e}^{\,r}_1 & \vec{e}^{\,r}_2 \cdot \vec{e}^{\,r}_2 \end{bmatrix}. \tag{6.6}$$

A different eigenstrain $(\varepsilon^t_{xx}, \varepsilon^t_{yy})$ and $(\varepsilon^b_{xx}, \varepsilon^b_{yy})$ imposed on the top and bottom layers respectively induce different first fundamental forms $\mathbf{a}_{r,t}$ and $\mathbf{a}_{r,b}$. Concurrently, all components of the rest second fundamental forms $\mathbf{b}_{r,t}$ and $\mathbf{b}_{r,b}$ equal zero because the edge normals stay perpendicular to the plane englobing the flat initial configuration.

Prescription of a different eigenstrain to the top and bottom layers does not make the plate delaminate. Instead, the plate adopts a curved configuration described with the *final* fundamental forms $\mathbf{a}_f$ and $\mathbf{b}_f$. Hence, from the mathematical point of view, the forward problem resolution consists in the determination of $\mathbf{a}_f$ and $\mathbf{b}_f$ taking $\mathbf{a}_{r,t}$ and $\mathbf{a}_{r,b}$ as the input. This is done through numerical minimization of the global elastic energy functional E_{BL} [98]:

$$
\begin{aligned}
E_{BL} = \frac{1}{2}\sum_{k=1}^{K}\Bigg[&\frac{h}{8}\Big\|(\mathbf{a}_{r,b}^{-1})_k(\mathbf{a}_f)_k - \mathbf{I}\Big\|_e^2 + \frac{h^3}{24}\Big\|(\mathbf{a}_{r,b}^{-1})_k(\mathbf{b}_f)_k\Big\|_e^2 \\
&+\frac{h^2}{8}\Big\langle\Big((\mathbf{a}_{r,b}^{-1})_k(\mathbf{a}_f)_k - \mathbf{I}\Big), (\mathbf{a}_{r,b}^{-1})_k(\mathbf{b}_f)_k\Big\rangle_e\Bigg]\sqrt{\det(\mathbf{a}_{r,b})_k} \\
+\frac{1}{2}\sum_{k=1}^{K}\Bigg[&\frac{h}{8}\Big\|(\mathbf{a}_{r,t}^{-1})_k(\mathbf{a}_f)_k - \mathbf{I}\Big\|_e^2 + \frac{h^3}{24}\Big\|(\mathbf{a}_{r,t}^{-1})_k(\mathbf{b}_f)_k\Big\|_e^2 \\
&-\frac{h^2}{8}\Big\langle\Big((\mathbf{a}_{r,t}^{-1})_k(\mathbf{a}_f)_k - \mathbf{I}\Big), (\mathbf{a}_{r,t}^{-1})_k(\mathbf{b}_f)_k\Big\rangle_e\Bigg]\sqrt{\det(\mathbf{a}_{r,t})_k}.
\end{aligned}
\tag{6.7}
$$

Here, the index k is attributed to the fundamental forms belonging to the element k, and K denotes the total number of elements in the model. The matrix $\mathbf{I}$ is the 2×2 identity matrix. The elastic norm $\|\cdot\|$ and the elastic energy inner product $\langle\cdot,\cdot\rangle$ denote the following operations:

$$\Big\|\mathbf{A}\Big\| = \alpha Tr^2(\mathbf{A}) + 2\beta Tr(\mathbf{A}^2), \tag{6.8}$$

$$\Big\langle\mathbf{A},\mathbf{B}\Big\rangle = \alpha Tr(\mathbf{A})\,Tr(\mathbf{B}) + 2\beta Tr(\mathbf{AB}), \tag{6.9}$$

where $\mathbf{A}$ and $\mathbf{B}$ are arbitrary matrices, $\alpha = Y\nu/(1-\nu^2)$ and $\beta = Y/(2+2\nu)$. The values ν and Y stand respectively for the Poisson's ratio and the Young modulus of the treated material.

We minimize the elastic energy functional with the help of the Limited-memory Broyden–Fletcher–Goldfarb–Shanno (L-BFGS) optimization algorithm [85] using the analytically computed gradients [86].

The inverse problem resolution

In terms of peen forming, the inverse problem consists in the determination of the eigenstrain tensors $\boldsymbol{\varepsilon}^t$ and $\boldsymbol{\varepsilon}^b$ imposed on each element taking the target shape as input. Another necessary input is the initial shape geometry. Our algorithm for the inverse problem resolution relies on full consistency between the initial and the target meshes. For this reason, the initial mesh is generated as a function of the target mesh, which is illustrated in Figure 6.3. It is done in two steps. Firstly, the target mesh is projected on the initial geometry using the Least Squares Conformal Mapping (LSCM) algorithm [88]. Secondly, the initial vertex positions are numerically optimized using the L-BFGS algorithm to minimize the element distortion with respect to the target mesh. The initial shape boundary is kept fixed during the optimization.

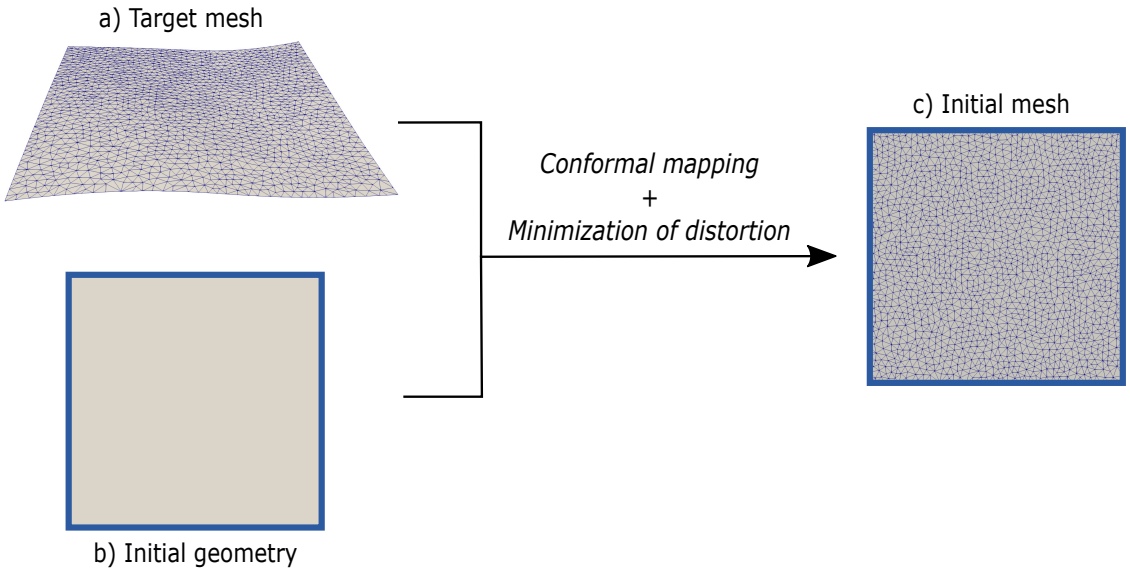

Figure 6.3 Generation of the initial mesh. The mesh of the curved target shape (a) is projected on the flat initial geometry (b) using the LSCM algorithm. The distortion of each element in the projected mesh is then minimized with respect to the target mesh using the L-BFGS algorithm. The resulting mesh (c) is used as the initial mesh during the inverse problem resolution.

The target shape is described with two fundamental forms $\mathbf{a}_{tar}$ and $\mathbf{b}_{tar}$. Our algorithm for the inverse problem resolution is iterative. Each iteration implies an adjustment of the eigenstrain pattern and subsequent computation of the new current shape provided by the adjusted pattern. The eigenstrain tensors are adjusted for each triangular element separately based on comparison of the current shape with the target shape. More precisely, on each iteration, we multiply ε^t and ε^b by the ratio between the local mean curvatures of the target shape H_{tar} and of the current shape H_c:

$$\varepsilon^{j,new} = \varepsilon^j \frac{H_{tar}}{H_c} \quad \text{for} \quad j = t, b. \tag{6.10}$$

Next, we solve the forward problem for $(\varepsilon^{t,new}, \varepsilon^{b,new})$ and thus obtain a new current shape. These two steps are repeated until stabilization of the current shape. The anisotropy coefficient χ is pre-defined with a calibration step described in Section 6.2.5 and kept constant throughout the iterations.

The initial guess is made in terms of $\mathbf{a}_{r,t}$ and $\mathbf{a}_{r,b}$ as presented in Ref. [98]. Thus, we first compute the $\mathbf{a}_{r,t}$ and $\mathbf{a}_{r,b}$ as:

$$\begin{cases} \mathbf{a}_{r,t} = \mathbf{a}_{tar} - \frac{2h}{3}\mathbf{b}_{tar}, \\ \mathbf{a}_{r,b} = \mathbf{a}_{tar} + \frac{2h}{3}\mathbf{b}_{tar}. \end{cases} \tag{6.11}$$

Then, for each layer, we take the average of the eigenstrains prescribed by $\mathbf{a}_{r,t}$ and $\mathbf{a}_{r,b}$ in each direction and use this eigenstrain as the initial guess.

This algorithm constitutes a simplified version of that described in Ref. [98]. Namely, the eigenstrain is adjusted in the way that it leads to the target *curvature*, but the *stretching* of the target shape is not taken into account. This formulation was adopted because the target shapes implied large rotations but small stretch with respect to the initial state.

The final step of the inverse problem resolution is the pattern grouping [98], which is done after the last iteration. The aim of this process is to split the peening pattern into segments treated with practically attainable peening regimes. Each attainable peening regime is related to the corresponding induced eigenstrain during the calibration phase. The grouping algorithm takes the eigenstrain assigned to each element and reassigns it the closest eigenstrain among the limited number of attainable ones. This removes gradual variation of the eigenstrain along the plate surface and divides the pattern into strictly defined peened segments. When the grouped pattern is computed, we solve the forward problem taking this pattern as input. This step serves for preliminary estimation of the error between the target shape and the final shape developed with the ready-to-use pattern.

Inverse identification of the anisotropy coefficient

This problem is similar to the inverse problem, except for the condition that the local eigenstrains ε^t and ε^b are pre-defined and the anisotropy coefficient χ must be determined. We assume that χ is a global property of the treated material, so it is uniform along the plate surface. The initial mesh is generated based on the target mesh in the same way as it is done during the inverse problem resolution.

The problem is solved through the iterative adjustment of χ based on the comparison of the current and target shapes. We compute local curvatures along the x-direction for both shapes: κ_x^c for the current shape and κ_x^{tar} for the target shape. We average their absolute values κ_{avg}^c and κ_{avg}^{tar} and then adjust χ as follows:

$$\chi^{new} = (1+\chi)\frac{\kappa_{avg}^{tar}}{\kappa_{avg}^c} - 1. \tag{6.12}$$

We then adjust the local eigenstrain tensors according to Eqn. (6.2) and solve the forward problem. Starting with an initial guess of $\chi = 0$, the process is repeated until the convergence of κ_{avg}^c and κ_{avg}^{tar}, providing us with an evaluation of the anisotropy coefficient χ.

6.2.2 The translation software

The translation software fills the gap between the numerical peening patterns and the actual peen forming process. The peening pattern that it takes as input is provided in the form of a mesh with the eigenstrain assigned to each of the elements. The pattern is divided into segments with different prescribed eigenstrain that correspond to treatment with different peening regimes. The relation between each peening regime, i.e., each set of the peening parameters, and the corresponding eigenstrain is defined during the calibration phase and saved in the software. The software first detects borders of the segments treated with the same peening regime. It then generates the nozzle paths that fill each segment and assigns the peening parameters corresponding to each segment. Finally, the software wraps up the paths and the peening parameters in a programming script written in Karel programming language [104], which is then loaded in the peening robot. The script controls all parameters of the robot operation and does not require any manual adjustment.

The translation software is able to fulfill a peened segment with two types of nozzle paths, which are presented in Figure 6.4. The first path type is called “Zigzag" and implies straight parallel nozzle paths. The second type is called “Circular". The paths generated in this way constitute a set of concentric closed lines, and the longest line is the outline of the treated segment. The “Circular" paths better fill complexly shaped segments. However, the neighboring paths of this type may not always have a constant offset, that results in a less uniform coverage.

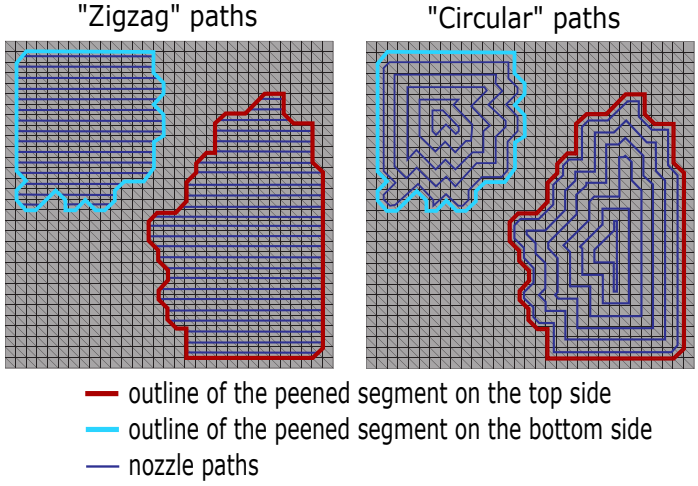

Figure 6.4 Two types of nozzle paths generated by the translation software. The example pattern implies treatment from the top and bottom sides with the same peening regime.

The software allows adjusting several common parameters for both path types. Thus, the outline of the treated segment can be smoothed with respect to its actual border prescribed by the peening pattern. This allows filtering out features too small to peen with respect to the shot stream width. Moreover, the outline can be offset with respect to the border, which allows scaling the peened segment without resorting to the simulation software. The offset between the paths can also be adjusted depending on the shot stream width, which is measured during the calibration phase. The path offset is adjusted in the way that the peening coverage in the treated segment is as uniform as possible. Figure 6.5 illustrates the adjusted outline and the concept of the path offset. The speed of movement of the nozzle along the peening paths can also be adjusted depending on the peening regime and is usually set to 50-100 mm/s. The peening robot is not able to block the shot stream during the program execution, so the nozzle moves between different peened segments and different paths inside one segment at a high speed of 1 500 mm/s.

6.2.3 The peening robot

The peening robot involved in the experiments is based on a Canablast shot peening machine. It possesses a FANUC M20iA programmable robotic arm, which guides the peening nozzle.

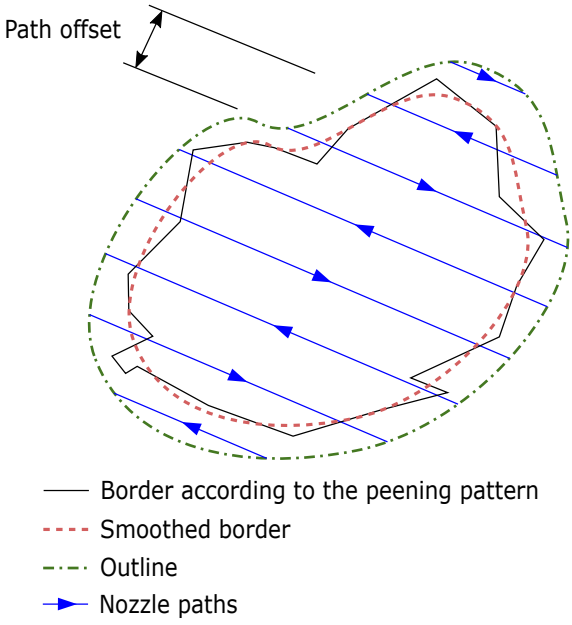

Figure 6.5 Adjustment of the peened segment executed with the translation software. The outline of the peened segment is smoothed with respect to its initial border and offset to increase the size of the treated segment. The path offset is adjusted to ensure uniform coverage.

The robotic arm and the treated component are installed inside a closed peening cell shown in Figure 6.6.

We developed a specific peen forming fixture to treat square panels of 320 × 320 mm. The fixture is designed in two main parts: the fixed and the removable frames. The first ensures a precise location of the peened part in terms of the robot's reference frame and the latter clamps the plate along its perimeter. This system eases the installation process and allows removing the plate from the shot peening cell while keeping it under the clamping constraints.

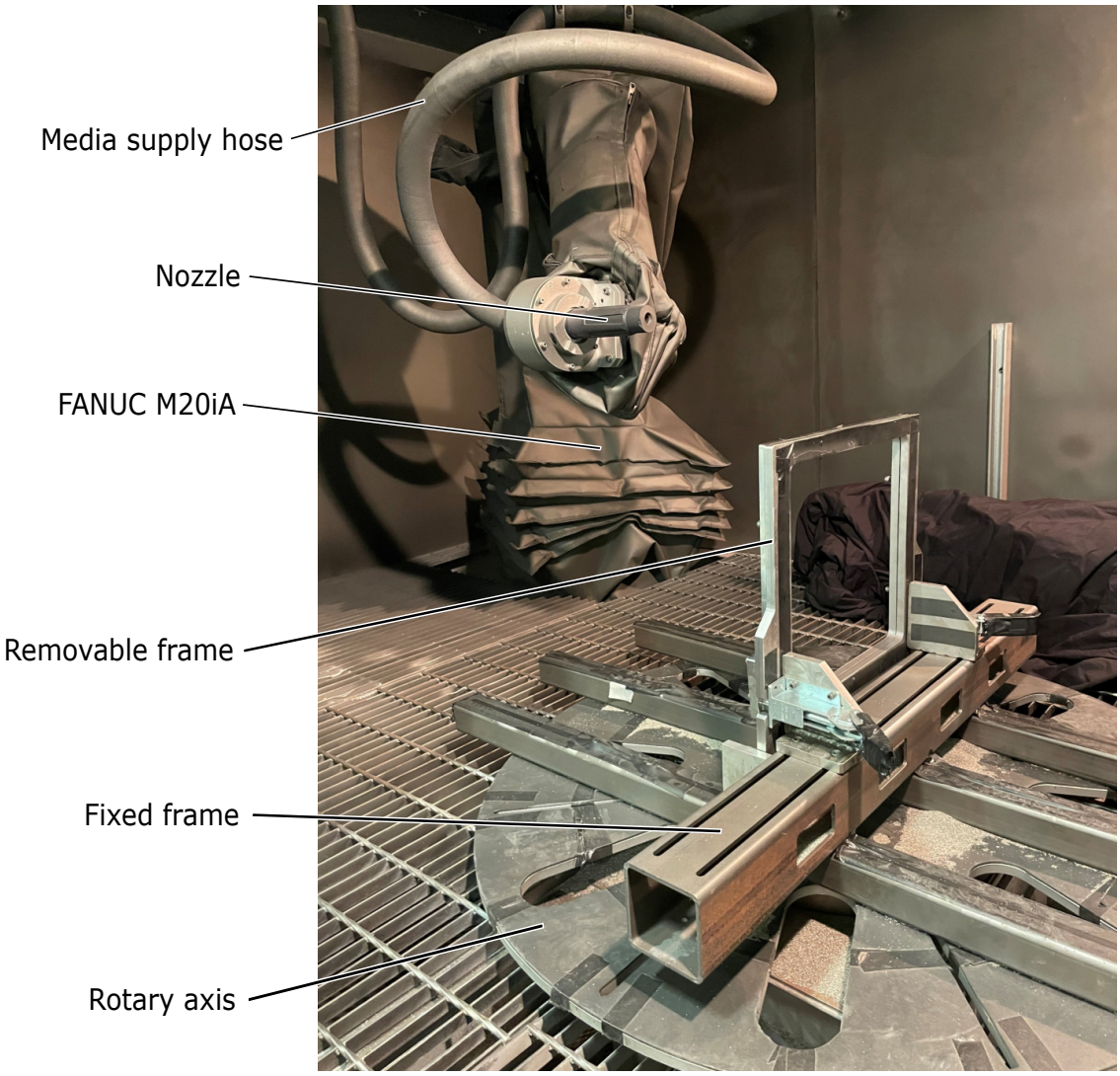

Figure 6.6 The interior on the robotic peen forming cell. The programmable robtic arm FANUC M20iA guides the peening nozzle, and the media is supplied to the nozzle through the elastic hose. The treated plate is clamped by the removable frame along its perimeter. The removable frame is mounted on the fixed frame attached to the rotary axis.

The fixture is mounted on a rotary axis, so the plate can be treated from both sides with no need for manual reinstallation. The rotary axis turns the plate by 180° in 2 seconds, and during this process the programming script points the nozzle aside, that prevents undesired impacts.

6.2.4 3D scanning and error map

Once a plate is treated, we compare its curved shape with numerical prediction. We remove the plate from the frame and we scan it using a Hexagon StereoScan neo 3D scanner. Before scanning, the plate is covered with Magnaflux SKD-S2 aerosol to make its surface non-reflective. The result of 3D scanning is a meshed surface describing the plate shape.

Next, we numerically superimpose each scanned shape with the corresponding predicted shape and we compute the Hausdorff distance d_H [105] between the two shapes. We nondimensionalize the Hausdorff distance by dividing it by the deflection of the target shape Δz, which is computed as the smallest dimension of a box bounding the plate. Thus, we obtain a nondimensionalized error Ω_z:

$$\Omega_z = \frac{d_H}{\Delta z}. \tag{6.13}$$

Figure 6.7 illustrates the workflow for determining Ω_z. The optimal alignment of the two shapes is done through least squares estimation of the transformation parameters [106].

6.2.5 Calibration steps

The shot stream characterization

We first estimate the shot stream intensity for each peening regime that we use. It is done following the standard procedure involving type A Almen strips and the Almen gage [12]. Each peening regime used for the treatment of the plates is applied until saturation, and the intensity measurement is necessary to estimate the appropriate number of nozzle passages. In other words, we estimate how many times the nozzle must pass along the paths prescribed by the translation software to achieve saturation in the whole treated segment.

The nozzle paths are lines that direct the center of the nozzle, but the actual shot stream has a finite width. Hence, we measure the distribution of shot inside the stream to define the nozzle path offset. It is mainly influenced by the distance from the nozzle to the treated component. We perform this test using a 418×127 mm flat dummy specimens. We make one peening passage in the center of the specimens along the longitudinal direction translating the nozzle at a high speed of 1 500 mm/s and then scan the specimen with a document scanner.

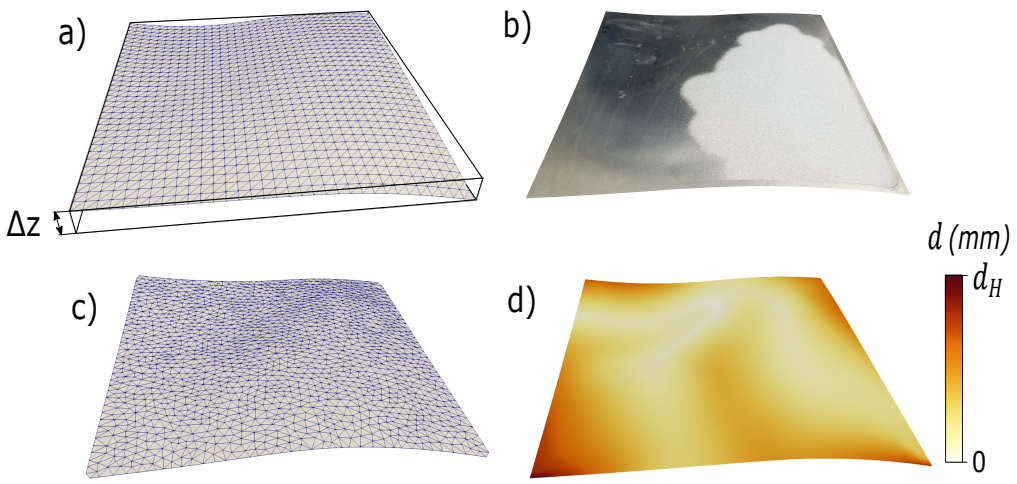

Figure 6.7 Determination of the dimensionless error $\Omega_z = d_H/\Delta z$ between the simulated shape and the shape of the treated plate. a) The simulated shape and its bounding box. The deflection Δz is computed as the smallest dimension of the bounding box. b) A photo of the plate treated following the computed pattern that leads to the simulated shape. The non-reflective surface is the treated part. c) The meshed shape scanned with a 3D scanner. d) Map of the local distances d between the simulated and the scanned shapes that were optimally aligned. The map is traced on the simulated shape. The Hausdorff distance d_H is the maximal value of local distances d.

Next, we post process the scanned image, i.e., we highlight the white pixels corresponding to impacts and we darken the background corresponding to the untreated surface. The picture resolution is 4830×1470 pixels, and we count the number of white pixels in each pixel column out of 1470. Figure 6.8 shows an example of the shot distribution measured in this way. We set the paths offset to the largest value that ensures uniform coverage of the treated segment with a maximal variation in coverage of 1%. This value equals approximately half of the shot stream width but is specific for each particular distribution.

Determination of the eigenstrain

Every combination of peening regime and treated material induces different ε^t and ε^b. We determine ε^t and ε^b experimentally using rectangular coupons of size 76×19 mm. The coupons are cut with waterjet cutting from the same aluminum plates as those that are shaped afterwards. The coupons are clamped in the Almen holder and treated until saturation.

The bending behavior of the coupon is governed by the dimensionless load Γ_B [35]:

$$\Gamma_B = \frac{3}{2}(1+\nu)\Delta\varepsilon, \tag{6.14}$$

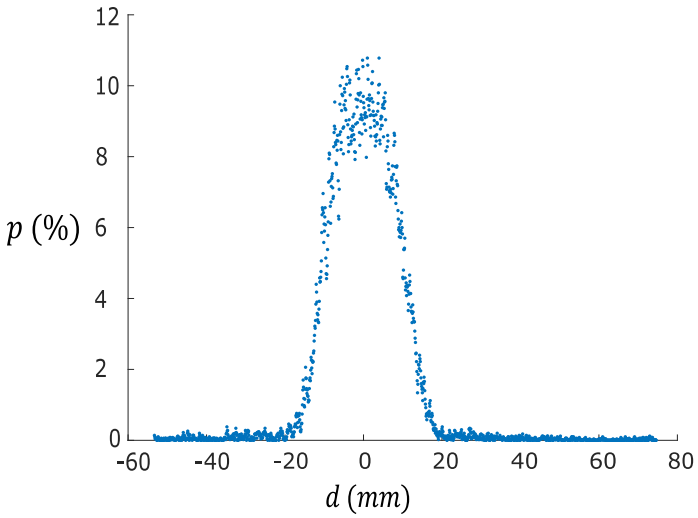

Figure 6.8 Spatial distribution of shot on a 418×127 mm flat dummy specimen that underwent one passage of the peening nozzle in the longitudinal direction. The nozzle travelled at a speed of 1500 mm/s, and the nozzle distance in this case equaled 152.4 mm. The graph indicates the percentage p of the peening-affected surface as a function of the transverse coordinate d. More precisely, we virtually divided the specimen into 1470 columns traced in the longitudinal direction, and the percentage p denotes the ratio of the peening-affected area in each column with respect to its total area.

where $\Delta\varepsilon = \varepsilon^t - \varepsilon^b$. For small values of Γ_B, the curvature κ measured at the center of the treated coupon is the same in any direction. Moreover, in this case the load Γ_B is proportional to the dimensionless curvature κh [35]. The treated coupon has a convex shape (κ is negative), and the relation takes the form:

$$\kappa h = -\mu \Gamma_B. \tag{6.15}$$

The proportionality constant μ can be determined using numerical simulations (see Appendix 6.5). Therefore, $\Delta\varepsilon$ can be deduced from κ. According to the simulations, the linear relation holds at least up to $\Gamma_B = 8 \times 10^3$.

The curvature κ, in turn, is deduced from the coupon deflection u, which is measured using the Almen gage [17]. We can approximate the shape of a curved coupon to that of an elliptic paraboloid [107], which allows to express u as:

$$u = -\kappa \frac{l_x^2 + l_y^2}{2}. \tag{6.16}$$

Here, l_x=15.87 mm and l_y=7.94 mm are the dimensions of the rectangle formed by the supports of the Almen gage [17].

The dimensionless curvature κh depends on $\Delta\varepsilon$ and does not depend on the sum $(\varepsilon^t + \varepsilon^b)$, which only influences stretching of the plates [98]. However, shot peening induces small eigenstrain [22], so we conclude that the peening-induced stretching is also small and assume that $\varepsilon^t = -\varepsilon^b$. Combining Eqns. (6.14 – 6.16) yields:

$$\begin{cases} \varepsilon^t = \dfrac{2hu}{3\mu(1+\nu)(l_x^2 + l_y^2)}; \\ \varepsilon^b = -\dfrac{2hu}{3\mu(1+\nu)(l_x^2 + l_y^2)}. \end{cases} \tag{6.17}$$

For better stability of the results, we treat four coupons with each regime and compute their average deflection u_{avg}. We then substitute u_{avg} for u in Eqn. (6.17) and thus find ε^t and ε^b.

Determination of the anisotropy coefficient

Aluminum plates are produced by rolling, and this process is one of the key factors inducing the eigenstrain anisotropy. Namely, the peening induced eigenstrain is different in the rolling direction (L) and in the in-plane transverse direction (T). This effect alters the local curvatures induced by peening in the L- and T-directions. Thus, if peen forming is simulated assuming that the material is isotropic, then in practice the local curvature in one direction is bigger than that simulated, and the local curvature in the other direction is smaller [35]. A similar effect is also observed during peen forming of metal laminates containing oriented filaments [108].

We quantify the eigenstrain anisotropy using the "wave" experiment. For this, we use the plates that have the same size and are produced in the same way as those used for the forming experiments. We treat two plates following the "wave" pattern illustrated in Figure 6.9. This pattern consists in treating half of the plate from one side and the other half from the other side. The orientation of the "wave" pattern controls the bending direction of the plate. Hence, for the first plate, the halves are separated along the T-direction thus forcing the plate to bend in the L-direction. On the contrary, the same pattern rotated by 90° makes the second plate bend along the T-direction.

We scan both treated plates using the 3D scanner and obtain their meshed numerical models. These models serve as target shapes for numerical determination of the eigenstrain anisotropy coefficient as described in Section 6.2.1. We orient the coordinate system in the way that the x-axis points in the "wave" direction, i.e., in the L-direction for the first plate and in the T-direction for the second plate. To compute the κ_{avg}^{tar}, we take into account only the elements that fall into a 200 × 200 mm square traced in the center of the plate (see Figure

6.9). The curvature of the other elements is not taken into account to reduce the influence of the edge effects.

In this way we determine two anisotropy coefficients: χ_L for the first plate and χ_T for the second plate. One of them is positive and the other is negative. We then compute the average of their absolute values $\chi_* = (|\chi_L| + |\chi_T|)/2$ and thus find the eigenstrain anisotropy coefficient for the given material. For the subsequent simulations we make a convention that the x-axis follows the principal eigenstrain direction, so that χ_* is positive.

Theoretically, χ_L must equal $-\chi_T$, so the "wave" treatment of one plate should be sufficient to determine χ_*. However, exact equality is rarely observed because of practical constraints. For example, the simulations do not take into account the initial stresses, which are different for each particular component, or the influence of clamping. For this reason, treatment of two

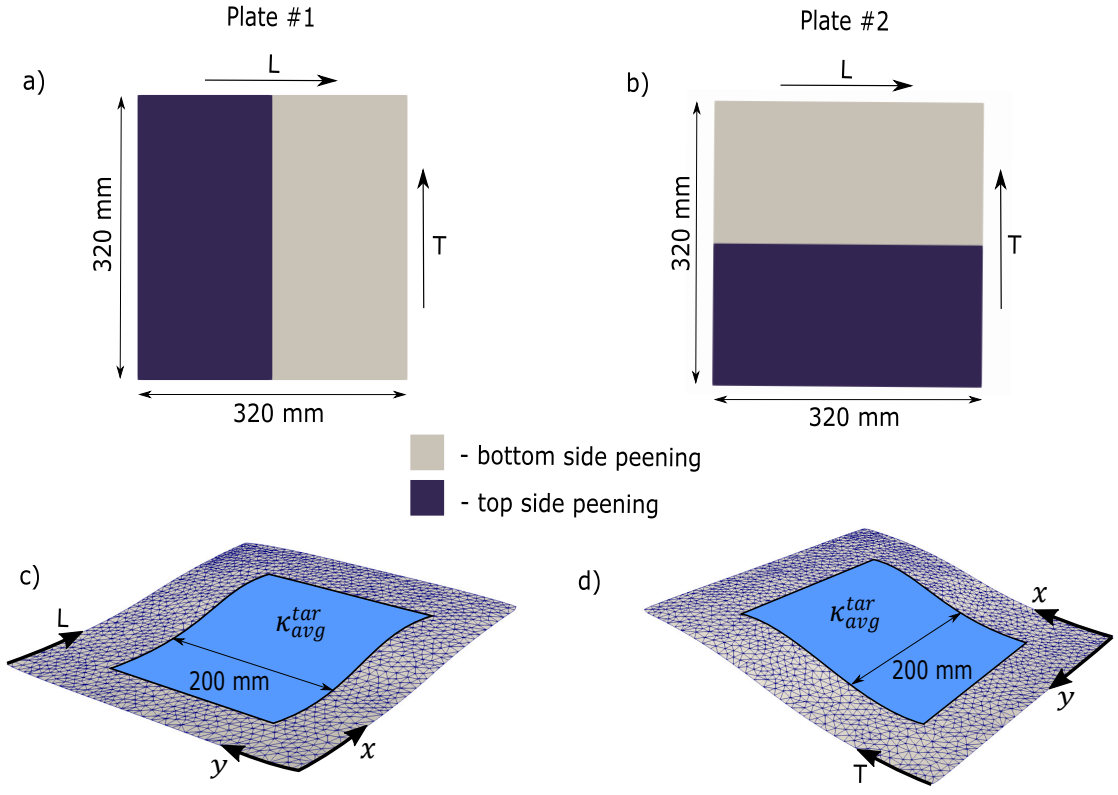

Figure 6.9 The "wave" experiment. The "wave" peening pattern is applied on two 320 × 320 mm sized aluminum plates. The pattern forces the plates #1 and #2 to bend along the L-direction (a) and the T-direction (b) respectively. The plates are scanned in 3D after peening. Figures c) and d) show the deformed state of the plates #1 and #2 respectively. For each plate, we compute the average absolute value of κ_x for all the elements contained in the 200 × 200 mm square traced in the center of the plate (blue), and thus obtain κ_{avg}^{tar}.

plates stabilizes the measured result and allows to check if ε^t and ε^b were correctly determined in advance. Moreover, the usage of more than two plates could further improve the precision of this experiment.

6.2.6 Experimental validation of the simulation software

The experiments consisted in shaping 320×320 mm aluminum plates. For this, we fixed six peening regimes described in Table 6.1. The peening equipment was charged with the cut wire CW28 media having shot diameter of 0.71 mm.

Validation of the forward problem solver

We shaped nine plates following randomly generated peening patterns and compared the developed shapes with the numerical predictions. The plates were 2.06 mm thick and were made of aluminum 6061-T6. The patterns implied one peening regime listed in Table 6.1 as #1.

To ensure that each pattern induces a different shape, we generated the patterns based on random target shapes. At first, we generated nine different shapes using the *peaks* function embedded in Matlab programming language. We then solved the inverse problem for each shape and thus obtained the peening patterns. We took into account the measured coefficient of eigenstrain anisotropy, i.e., we imposed the ratio $(1 - \chi_*)/(1 + \chi_*)$ between the induced eigenstrain in the L- and T-directions.

Table 6.1 Parameters of the peening regimes used in the experimental campaign. The nozzle was oriented perpendicularly to the treated surface. The mass flow rate determines the mass of media that passes through the nozzle. The nozzle distance is the distance from the nozzle to the treated surface. The pressure is created by compressed air. The peening intensity and the number of passes to saturation were measured following the standard procedure using type A Almen strips [12]. The speed of the nozzle travelling was fixed at 100 mm/s.

Regime	Mass flow rate (kg/min)	Nozzle distance (mm)	Pressure (kPa)	Intensity (A)	Saturation (Number of passes)
#1	2.27	152.4	248.2	23.5	5
#2		152.4	172.4	19.2	4
#3		152.4	137.9	17	4
#4		152.4	103.4	13.6	4
#5		152.4	68.9	10.9	8
#6		304.8	48.2	8	9

The target shapes were generated without any constraints on their deflection. The inverse problem solution showed that their development implied much larger eigenstrain magnitude than those achievable with regime #1. Thus, according to the forward problem resolution, the simulated shapes induced by the grouped pattern were less curved than the initially generated target shapes. Consequently, we used these less curved simulated shapes as reference for the experiments.

Validation of the inverse problem solver

The inverse problem resolution was validated using $320 \times 320 \times 1.6$ mm plates made of aluminum 2024-T3. The objective of this validation was to shape parts of the upper and lower skins of a model airplane wing. The wing had constant cross-section in the form of NACA 5430 airfoil [109] with the total chord length of 500 mm (see Figure 6.10). The two target shapes had the same area as the flat plates: 320×320 mm.

We first solved the inverse problem for both target shapes and obtained the peening patterns. We then solved the forward problem to predict the dimensionless error Ω_z with respect to the target shapes. Next, we translated the peening patterns into programs for the peening robot and treated the aluminum plates. Finally, we compared the experimentally developed shapes with the target shapes constituting the NACA airfoil in terms of Ω_z.

The peening patterns consisted of segments treated with the regimes #2-5 from Table 6.1. The plates were also uniformly treated from both sides with regime #6 before the application of the peening pattern. Such low intensity pre-treatment is commonly used in industry to enhance the fatigue life of the components. It also makes the surface finishing more uniform.

Figure 6.10 The two target shapes for the 320×320 mm sized panels made of aluminum 2024-T3. The target shapes constituted parts of the upper (blue) and lower (orange) airplane wing skins. The wing was shaped as the NACA 5430 airfoil with the chord length of 500 mm.

6.3 Results

6.3.1 Calibration steps

The width of the peening-affected band was approximately 4 cm for the nozzle distance of 152.4 mm and approximately 8 cm for the nozzle distance of 304.8 mm. The largest path offsets that provided uniform coverage corresponded to 19 mm and 38 mm respectively.

We determined the eigenstrain using 76×19 mm aluminum coupons (see Section 6.2.5). The 2.06 mm thick coupons made of aluminum 6061-T6 were treated to saturation from one side with regime #1. The 1.6 mm thick coupons made of aluminum 2024-T3 were first treated to saturation from both sides with regime #6 and then treated to saturation from one side with the regimes #2-5. The average deflection of the four coupons treated with each regime is presented in Table 6.2. Table 6.2 also shows the eigenstrain ε^t induced by each regime. It was computed as a function of deflection u_{avg} using the Eqn. (6.17). We observed a slight difference in deflection of less than 1 % of average in each bunch of the four coupons. It is explained by the presence of initial stresses coming from the manufacturing process.

To characterize the eigenstrain anisotropy (see Section 6.2.5), we treated the 6061-T6 plates following the "wave" pattern with regime #1. The 2024-T3 plates were first treated uniformly with regime #6 and then underwent the "wave" treatment with regime #2. The anisotropy coefficients revealed by the "wave" experiment for the two materials are presented in Table 6.3. The 6061-T6 plates demonstrated a 4.6 times stronger eigenstrain anisotropy than the 2024-T3 plates.

Table 6.2 Results of the eigenstrain measurements performed using the bunches of four 76×19 mm sized aluminum coupons (see Section 6.2.5). The coupons were fixed during peening with an Almen holder, and their deflection was measured with an Almen gage. The thickness of the coupons is denoted by h. The average deflection in each bunch of four coupons is denoted by u_{avg}, and the parameter σ_u describes the standard deviation in deflection. The eigenstrain induced in the top layer is denoted by ε^t. In this study we suppose that the eigenstrain ε^t equals $-\varepsilon^b$, which stands for the eigenstrain induced in the bottom layer.

Regime	Pre-treatment (regime)	Material	h (mm)	u_{avg} (mm)	σ_u (mm)	$\varepsilon^t \times 10^3$
#1	-	6061-T6	2.05	0.341	0.003	1.19
#2	#6	2024-T3	1.6	0.543	0.005	1.47
#3				0.302	0.015	1.2
#4				0.443	0.004	0.82
#5				0.174	0.007	0.47

Table 6.3 The anisotropy coefficients determined using the "wave" experiment for the two materials. The coefficient χ_* is the average of the absolute values of χ_L and χ_T. The coefficient χ_* was used as the eigenstrain anisotropy coefficient during the simulations.

Material	χ_L	χ_T	χ_*
6061-T6	-0.143	0.193	0.168
2024-T3	0.013	-0.06	0.0365

6.3.2 The forward problem solver validation

The nine random peening patterns used for validation of the forward problem solver are presented in Figure 6.11. The patterns and the corresponding simulated shapes were numbered from #1 to #9. Figure 6.12 (a-f) shows three representative cases demonstrating the smallest, the biggest and an average dimensionless error Ω_z. This figure also traces the local dimensionless mean curvature Hh of each shape, showing that the curvature was qualitatively well simulated. The simulations were performed on the same irregular mesh as that generated by the 3D scanner to allow element-wise curvature comparison. Thus, the irregular mesh was projected onto the initial geometry using the LSCM algorithm and optimized using the L-BFGS algorithm in the same way as during the inverse problem resolution (Figure 6.3).

The dimensionless error lied in a compact range between 20.5% and 28.5% for seven shapes out of nine. This means that the Hausdorff distance d_H generally increases with Δz, which is illustrated in Figure 6.12 (g). The shapes #4 and #7 demonstrated significantly smaller Ω_z than the others (8% and 12% respectively), which is explained by the following reason.

The developed jig did not completely prevent deformation of plates during treatment. It clamped the perimeter but let the interior part deform. At the same time, the peening pattern was first fully reproduced from the top side and then from the bottom side. Consequently, when the treatment was applied from the bottom side, the plates were already bent towards the top side. This created an effect of prestress that decreased the eigenstrain in the segments treated from the bottom side [97]. Accordingly, the curvature in these segments was smaller than the simulated one.

The induced prestress varies locally along the plate and is also dependent on the peening pattern. We quantified this effect with the parameter ψ that indicated the ratio of areas treated on the top and bottom sides:

$$\psi = \frac{A_{top} - A_{bot}}{A_{total}}, \tag{6.18}$$

where A_{top} and A_{bot} are the areas of the segments treated on the top and bottom sides

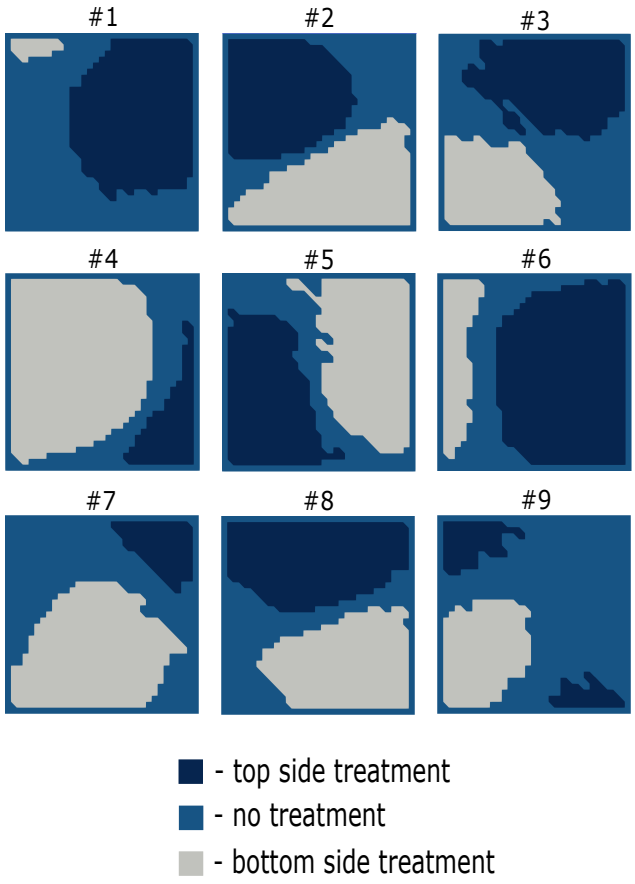

Figure 6.11 The nine peening patterns that were applied during the experimental validation of the forward problem solver. Each pattern implied treatment from both sides. The plates were clamped along their perimeter, so the clamped part did not undergo any treatment. For this reason, each pattern has margins along the plate edges. The patterns were applied using regime #1 without any pre-treatment.

respectively, and A_{total} is the total plate area. Figure 6.12 (h) plots Ω_z as a function of ψ. This graph shows that the plates #4 and #7 had the smallest ψ among the others, which means that these plates were prestressed less than the others when peened from the bottom side. Accordingly, these shapes showed the smallest Ω_z. Essentially, the problem of an undesirable prestress may be mitigated in practice by constant alternation of the peened sides during treatment. Moreover, this problem becomes less important for stiffened industrial parts, because the stiffeners block bending of the component in the longitudinal direction unless they are not treated themselves [67].

Another issue that increased Ω_z in all the cases consisted in the shot stream width. Thus, the shot stream was not sufficiently narrow to precisely reproduce the borders of the treated segments, which is illustrated in Figure 6.13. This effect had stronger negative influence on

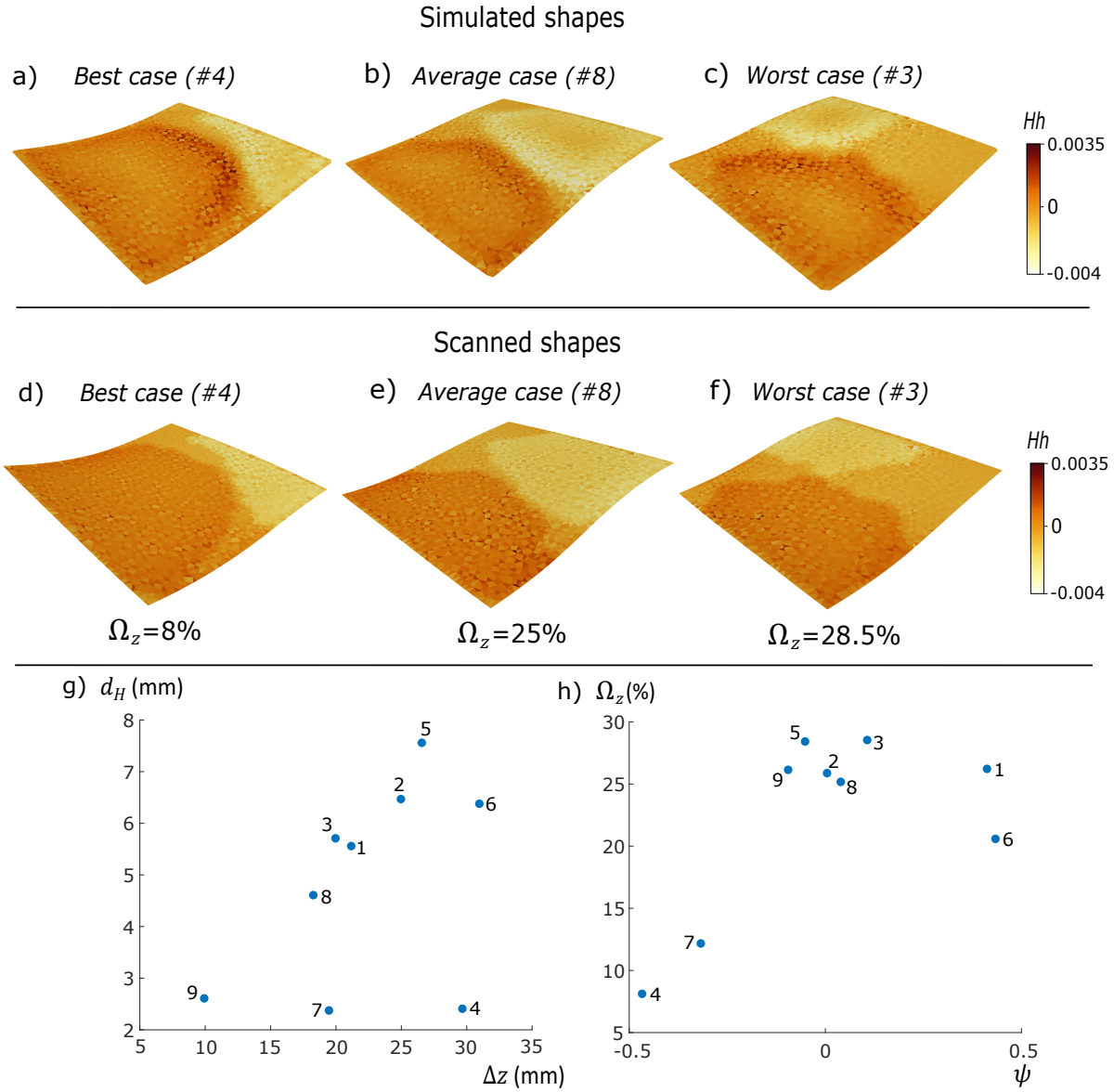

Figure 6.12 Results of the forward problem solver validation. We present three representative simulated shapes, which showed the smallest (a), an average (b) and the biggest (c) discrepancy with the experimentally developed shapes. The 3D scanned shapes developed in practice are marked as d), e) and f) respectively. The colormap indicates the local dimensionless mean curvature Hh. The discrepancy was quantified in terms of the dimensionless error Ω_z (Eqn. 6.13). The graph g) traces the Hausdorff distance d_H for all nine cases depending on the plate deflection. The graph h) traces Ω_z as a function of the parameter ψ, that indicates the difference between the treated areas on the top and bottom sides (Eqn. 6.18). Each point on the graphs is marked with the number of plate that it corresponds to.

patterns prescribing highly irregular borders to the peened segments. Hence, the plates #5 and #9 demonstrated a high Ω_z while having a relatively small ψ (see Figure 6.12 h), because the borders of the corresponding peened segments were strongly curved (see Figure 6.11). On the contrary, the plates #1 and #6 demonstrated a moderate Ω_z while having a large ψ, because the corresponding patterns prescribed smoothly shaped peening segments. The described effect may be mitigated technically with a nozzle that provides a narrower shot stream.

The third source of error consisted in the forces and the moments induced by clamping, which influenced local curvatures inside the treated areas. Thus, the simulations prescribed uniform curvature inside each treated segment, but the experiments revealed that the curvature along the perimeter of the treated segment was higher than in the middle of the segment. This effect was observed on all plates and is visible in Figure 6.12 (a-f). The influence of this effect depends on the shape of the peening pattern, and its quantification is left for further studies. Nevertheless, we expect the influence of clamping to be weaker for real size industrial panels that have much bigger radii of curvature.

6.3.3 The inverse problem solver validation

The results of the inverse problem solver validation using the 2024-T3 plates are presented in Figure 6.14. The fact that these plates were peened from only one side reduced the effect of the undesired prestress. Nevertheless, the outlines of the treated segments were blurred by the wide shot stream, and clamping of the plates altered their curvature, which caused

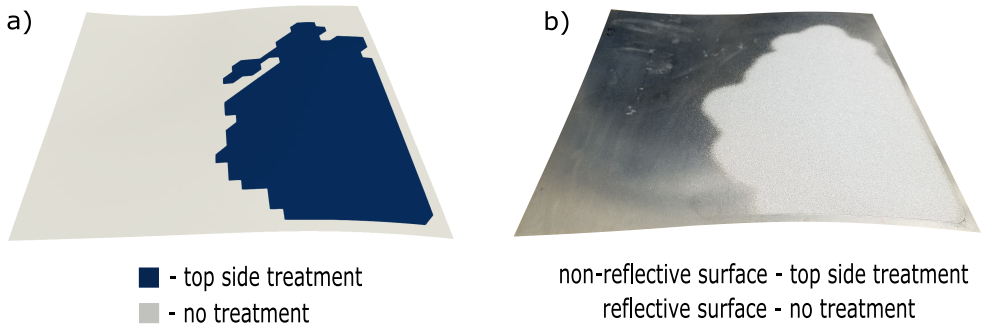

Figure 6.13 Difference between the prescribed peening pattern and the pattern that was actually applied during treatment. a) The treated area prescribed by the pattern #3 on the top side. b) A photo of the top side of the plate that was treated following pattern #3. The contour of the treated segment is smoother than the one prescribed by the pattern, because the shot stream was too wide to reproduce the contour precisely.

discrepancy between the simulations and the scanned shapes.

The discrepancy between the simulated shapes and the target shapes showed that the fixed peening regimes did not suit perfectly for the chosen target shapes. Thus, the simulated Ω_z equaled 1.2 % for the upper part and 1.4 % for the lower part with the free pattern, that was adjusted iteratively and was free of constraints, and grew up to 5.5 % and 12.9 % respectively with the grouped pattern, that implied only the fixed peening regimes. The main reason for this is that the precise reproduction of the target curvature required gradual transition of intensities from the front line of the wing towards the back line. However, having four predefined peening regimes was not sufficient to ensure smoothness of this transition. Moreover, the highest required eigenstrain for the upper (0.0023) and for the lower (0.0016) parts was higher than that provided by the most intense regime #2 (0.0015). Consequently, the curvature induced by the grouped pattern along the front line was inferior to the target one. Figure 6.15 provides details on the free and the grouped patterns for both target shapes.

The upper plate showed a bistable behaviour after treatment. Thus, the computed final shape appeared to be an unstable state of equilibrium, and the plate tended to curve along the diagonal axis in an unconstrained state. For this reason, we applied a weight on two corners of this plate during scanning, which is a common practice during the accuracy measurements in the peen forming industry. The weight the plate to keep the prescribed longitudinal symmetry, but the two corners had to be discarded from the numerical model provided by the 3D scanner (see Figure 6.14).

We used the two plates to assemble a physical model of the NACA 5430 airfoil, which is presented in Figure 6.16. The forward and the back parts of the model wing, which are colored in black in Figure 6.10, were printed in 3D using an FDM printer charged with the PLA plastic. These parts also served as supports for the aluminum plates. The plates were drilled and assembled with the printed parts using binding barrels and screws. The constraints induced by the binding have eliminated the shaping error, so the plates perfectly fit the NACA profile. This situation is often the case in industry, where the parts are shaped up to a pre-defined tolerance, and the error is entirely eliminated only when the part is installed.

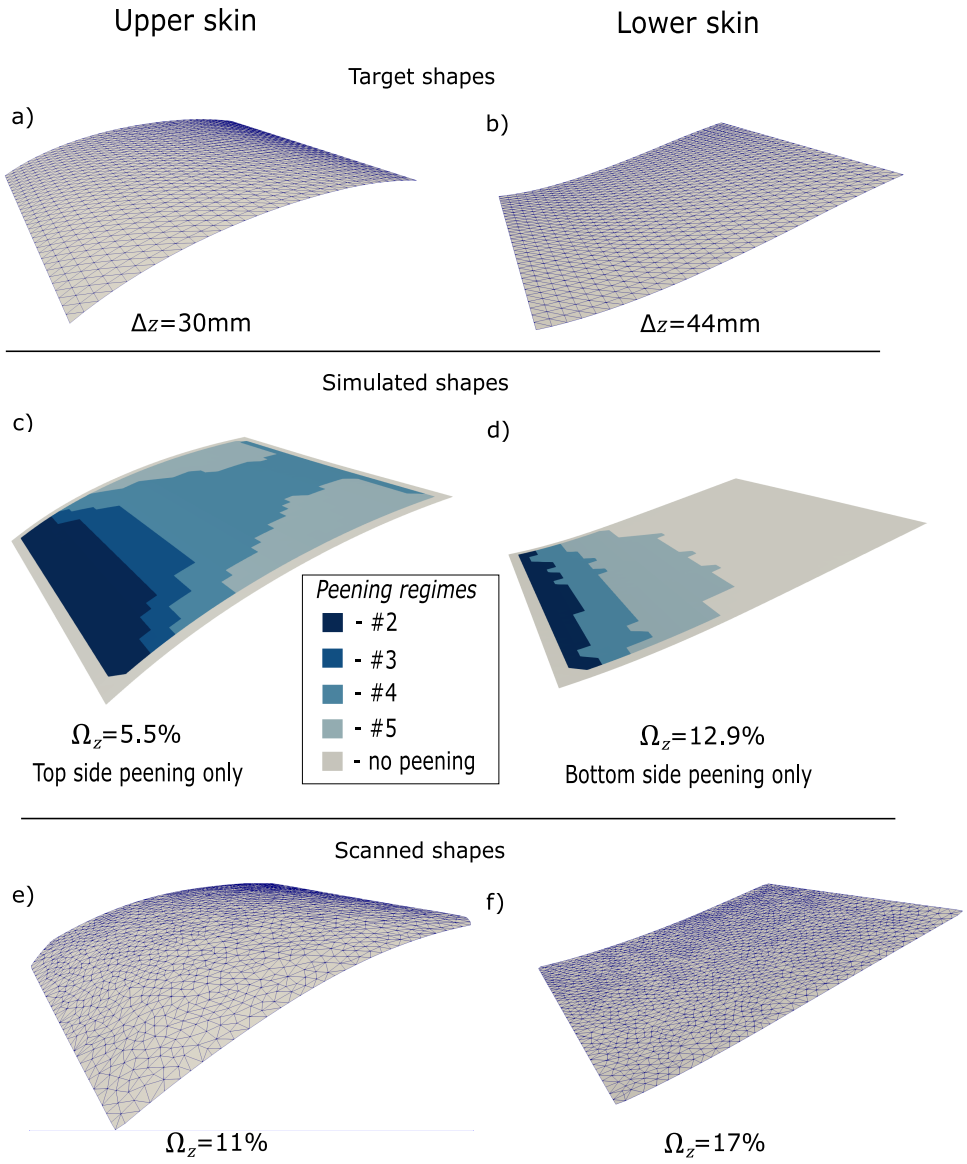

Figure 6.14 Results of the inverse problem solver validation. The target shapes constitute upper (a) and lower (b) parts of the airplane wing skin reproducing the NACA 5430 airfoil. The inverse problem was solved for both shapes. We simulated deformation induced by the prescribed patterns for the upper (c) and lower (d) skins and compared the simulated shapes with the target shapes. The aluminum plates were treated according to the patterns, and the 3D scans of the upper (e) and lower (f) skins were compared to the target shapes. The deflection Δz is computed as a function of the target shape. The parameter Ω_z describes the dimensionless error with respect to the target shape (Eqn. 6.13). The plates were made of aluminum 2024-T3 and had the size of $320 \times 320 \times 1.6$ mm. They underwent uniform treatment from both sides with regime #6 before application of the peening patterns.

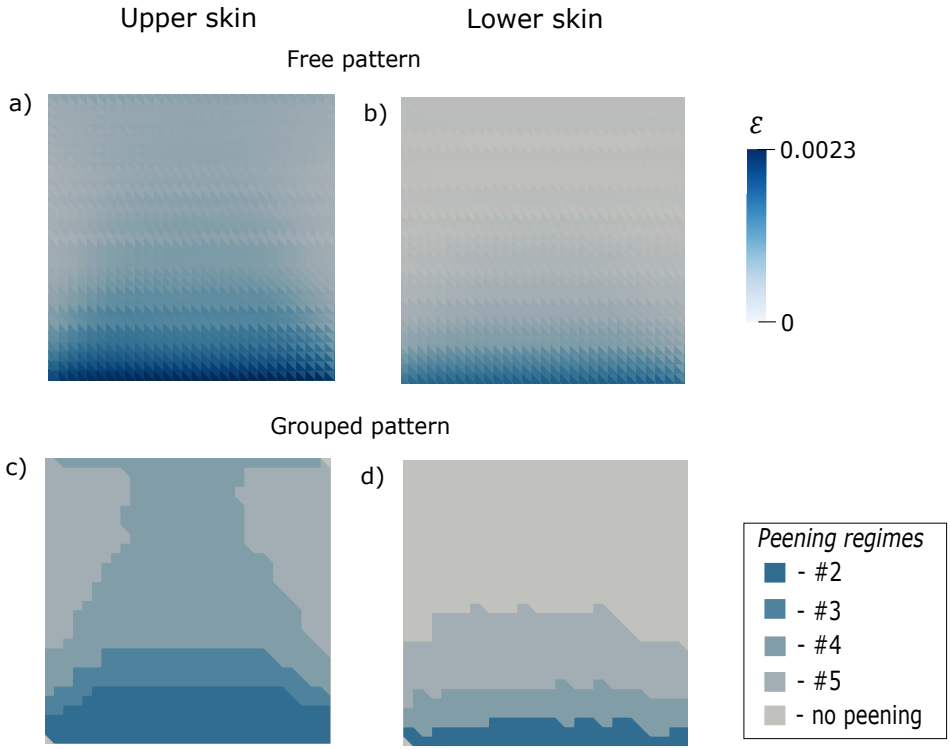

Figure 6.15 The eigenstrain patterns prescribed by the inverse problem solver to shape the airplane wing skin parts. Before grouping, the patterns assign gradual transition of the eigenstrain along the plate surface to shape the upper (a) and the lower (b) skins. The grouping process divides the patterns for the upper (c) and lower (d) parts into segments with constant eigenstrain. The eigenstrain in the grouped segments corresponds to the one induced by the fixed peening regimes.

Figure 6.16 The model of an airplane wing skin shaped as the NACA 5430 airfoil with a chord length of 500mm. The upper and lower skins were shaped using the shot peen forming, and the forward and the back parts (black) were printed in 3D. The structure is assembled using binding barrels and screws.

6.4 Conclusion

We developed a complete workflow for the shot peen forming automation. We generate the numerical peening pattern and we automatically translate it into a program for the shot peening robot. The program admits application of multiple peening regimes following complex patterns and thus allows to shape complex doubly-curved shapes. The workflow implies preliminary characterization of the shot stream effect in terms of eigenstrain and characterization of the eigenstrain anisotropy. These preliminary stages are necessary to generate the peening pattern and the nozzle paths. Simple yet efficient procedures were designed for this purpose.

The workflow was validated experimentally. The validation implied treatment following randomly generated patterns and shaping skins of a model airplane wing. The treatment was performed at high Almen intensities of 10.9-23.5A to reveal all possible errors appearing with large deformations. The validation showed promising results but revealed an error between the simulated shapes and the practically developed shapes from 8% to 28.5% of the total deflection. We expect this error to be smaller for industrial parts that have bigger radii of curvature and include stiffeners, because these components are less subjected to the undesirable prestress and to the influence of clamping supports. The inverse problem solver validation also emphasized the importance of an optimal choice of the peening regimes for each target shape.

The vast experimental campaign indicated directions for the workflow improvement. The simulation error was provoked by an incomplete consistency between the simulated and the real peening conditions. The major difference lies in the fact that the software simulates immediate application of the peening treatment from both sides, while in practice it was first fully applied from one side and then from the other. The future experiments will imply uniform application of the pattern from both sides, i.e., alternation of the peened sides after each pass of the peening nozzle. In addition, the peening nozzle will be modified in order to provide a narrower shot stream and thus to reproduce the pattern more accurately. The effect of clamping on the eigenstrain distribution will also be quantified.

Acknowledgments

The authors gratefully acknowledge financial support from The Natural Sciences and Engineering Research Council of Canada (NSERC), from The Fonds de Recherche du Québec – Nature et Technologies (FRQNT) and The Ministère de l'Économie et de l'Innovation du Québec, from The Aluminium Research Centre – REGAL, and from the industrial partner of

this project – Aerosphere Inc. The authors also acknowledge the consulting support provided by Aerosphere Inc. An important contribution to this project was made by Raphaël Paradis, who fabricated the peen forming fixture during his internship at the Aerospace Technology Center (CTA), and by Nada Abbassi, who performed the experimental manipulations for validation of the inverse problem solver during her internship at Polytechnique Montreal.

6.5 Appendix A: Proportionality between the dimensionless load and curvature

We performed a specific series of simulations in order to find the proportionality constant μ relating κh and Γ_B (Eqn. 6.15). Namely, we took a range of the dimensionless loads Γ_B and simulated its application on aluminum coupons of size 76×19 mm and various thickness h. All in all, 100 test cases for Γ_B ranging between 0 and 0.008 were simulated. For each test case, ε^t was positive and ε^b was negative, but the ratio $\varepsilon^t/\varepsilon^b$ was chosen randomly between -1 and 0. The material was considered plastically isotropic. Additionally, the thickness h ranged randomly from 2 to 4 mm for each test case. The Poisson's ratio ν was fixed equal to 0.33, that is a typical value for aluminum. We meshed the coupons with 760 triangular elements and solved the forward problem as described in Section 6.2.1. For all the test cases, the curvature κ measured at the center of the coupon varied by less than 0.015% in different directions, that proves the results presented in Ref. [35].

6.6 Appendix B: Influence of the eigenstrain anisotropy

The eigenstrain anisotropy is majorly caused by the material properties, but its influence is not always perceptible during uniform treatment of the 76×19 mm coupons. Thus, the 1.6 mm and 2.06 mm thick coupons develop the same local curvature in both directions when treated with the regimes described in Table 6.1. This is explained by the fact that the curvature of uniformly peened rectangular specimens depends on [25] [35]:

$$\overline{\Gamma}_B = \Gamma_B \frac{D^2}{h^2}, \tag{6.19}$$

where D is the characteristic length of the specimen. Namely, rectangular specimens adopt spherical shape for low values of $\overline{\Gamma}_B$ independently of the eigenstrain anisotropy. Thus, the applied treatment was not intense enough to make the 76×19 mm sized coupons develop different curvatures in the L- and T-directions. The 320×320mm sized plates have 17.7 times bigger $\overline{\Gamma}_B$ than the coupons of the same thickness treated with the same regime. Consequently, this type of plates was used for determination of the eigenstrain anisotropy.

6.7 Appendix C: Influence of pre-treatment on the induced eigenstrain

We performed additional measurements to characterize the eigenstrains induced in the 2024-T3 panels that did not undergo the preliminary double-side treatment with regime #6. First, we measured the eigenstrains induced in the 76 × 19 mm coupons cut of the same plates (see Section 6.2.5). The results of this measurement are summarized in Table 6.4. The induced deflection of the coupons was bigger than in the case with pre-treatment, so the induced eigenstrains were bigger as well. This phenomenon is explained by hardening of the surface generated by the pre-treatment, which is discussed for example in Ref. [49]. The influence of hardening was more significant for less intense peening regimes (Table 6.1). Thus, the absence of pre-treatment increased the eigenstrain induced by the regime #5 by 169%, while the eigenstrain induced by the regime #2 was only increased by 30%. The graphical comparison of the eigenstrain induced in both cases is presented in Figure 6.17.

Next, we characterized the eigenstrain anisotropy using the "wave" experiment (see Section 6.2.5). Thus, we treated two 2024-T3 panels following the "wave" pattern with regime #2 without pre-treatment. The experiment revealed a mild eigenstrain anisotropy, which is characterized in terms of the coefficient χ_* in Table 6.5. The principal eigenstrain direction has changed with respect to the case with pre-treatment, but the value of χ_* is of the order of 10^{-3}, which means that the eigenstrain is almost isotropic. Given that the eigenstrain anisotropy demonstrated by the pre-treated plates was mild as well (Table 6.3), we conclude that the pre-treatment did not have a tangible influence on this parameter.

Table 6.4 Results of the additional eigenstrain measurements performed using the bunches of four 76 × 19 × 1.6 mm sized aluminum coupons (see Section 6.2.5). The coupons were made of aluminum 2024-T3 and did not undergo any pre-treatment. They were fixed during peening with an Almen holder, and the deflection was measured with an Almen gage. The average deflection in each bunch of four coupons is denoted by u_{avg}, and the parameter σ_u describes the standard deviation in deflection. The eigenstrain induced in the top layer is denoted by ε^t. The right column shows the increase in ε^t with respect to the pre-treated coupons peened with the same regimes (Table 6.2).

Regime	u_{avg} (mm)	σ_u (mm)	$\varepsilon^t \times 10^3$	Increase in ε^t
#2	0.706	0.008	1.91	169%
#3	0.629	0.015	1.7	92%
#4	0.582	0.012	1.58	42%
#5	0.467	0.001	1.26	30%

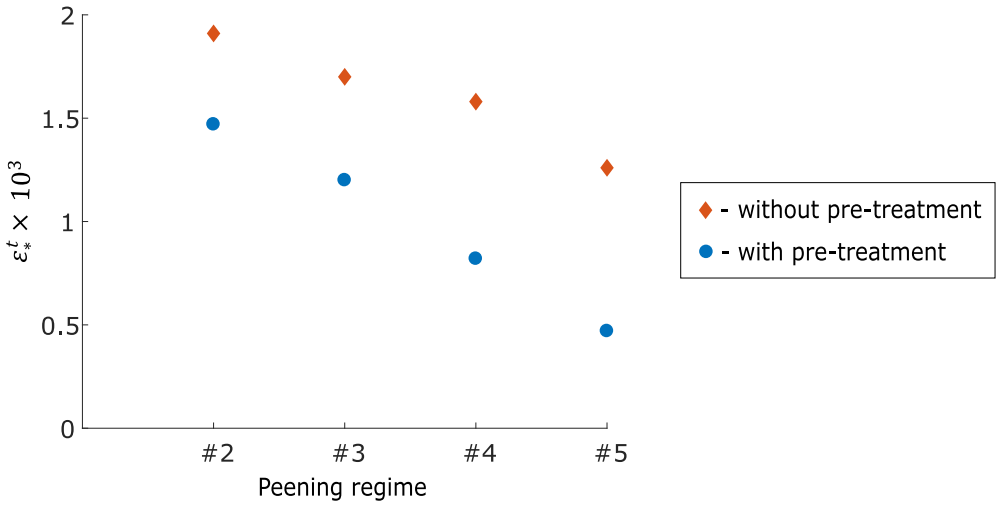

Figure 6.17 Graphical comparison of the eigenstrain ε^t induced by the same peening regimes on pre-treated and not pre-treated coupons. The coupons had the size of $76 \times 19 \times 1.6$ mm and were made of aluminum 2024-T3. The pre-treatment was done using the regime #6. The parameters describing each peening regime are presented in Table 6.1. The exact values of ε^t for each case are presented in Tables 6.2 and 6.4.

Table 6.5 The anisotropy coefficients determined using the "wave" experiment for 2024-T3 plates that were not pre-treated. The "wave" pattern was applied using the regime #2. The eigenstrain anisotropy coefficient χ_* is the average of the absolute values of χ_L and χ_T.

χ_L	χ_T	χ_*
-0.013	0.003	0.008

CHAPTER 7 GENERAL DISCUSSION

This chapter offers a critical look back at several aspects of our modeling approach. Firstly, we discuss the numerical shell model that we use to solve the forward problem and compare it with analogs. Secondly, we consider the energy-based formulation for the inverse problem solver, which can potentially be implemented instead of the analytical formulation presented in Chapter 4. We discuss its benefits and the barriers for its implementation. Thirdly, we explore the practical differences between various types of the idealized through-thickness eigenstrain profiles, and we explain our choice of the bilayer profile throughout this research.

7.1 Choice of an optimal shell model

The numerical method that we use for the forward problem resolution (see Chapter 4) is a geometric implementation of the shell finite element method in its constant-strain and constant-moment formulation. The degrees of freedom (DOF) of the shell model are the positions of vertices and the angles of inclination of the edge directors. Such a model is called the discrete Cosserat shell (DCS) [61]. A similar constant-strain and constant-moment element can also be implemented in a rotation-free formulation, meaning that the only DOF of the model are the vertex positions. A shell model of this type is called the discrete Koiter shell (DKS) [61]. The DKS model works faster on the same mesh due to lower number of optimized variables and, moreover, it is easier to implement. However, the absence of rotational degrees of freedom makes the DKS model sensitive to mesh irregularities [110], while the additional edge-director degree of freedom is specifically introduced in the DCS model cope with this problem [111]. Consequently, the DKS model is inappropriate for the the shot peen forming simulations in the general case. Indeed, complex part geometries may be meshed irregularly. In addition, the target shape can be provided by 3D scanning of a sample part, and the 3D scanners typically generate highly irregular meshes. Nevertheless, both models are suitable for regularly meshed plates [61, 62] such as those considered during the numerical validation campaign described in Chapter 4.

Weischedel et al. (see Ref. [61]) solved a series of "benchmark" FEM problems using both the DKS and the DCS models and different meshes. The solutions were also compared to those obtained in Ref. [112] using a conventional shell finite element formulation with free vertex positions and rotations (S4R elements in ABAQUS software). Figure 7.1 provides the results of one of these benchmark tests. The test simulates lifting of one end of a slit annular plate with a distributed vertical force, while the other end is clamped. The curves plotted

in Figure 7.1 show the deflection of the plate computed with different methods and different mesh resolutions. According to these results, the DKS model acts too stiff for irregular meshes thus underestimating the deflection. Nevertheless, a finer regular mesh makes both the DKS and the DCS models converge to the S4R model. At the same time, the original results of Sze et al. (see Ref. [112]) suggest that the SR4 model is less sensitive to the mesh resolution than the DKS and the DCS. Thus, the slit annular plates meshed with 180 SR4 elements (1302 DOF) and 800 SR4 elements (5346 DOF) demonstrated a negligible difference in deflection.

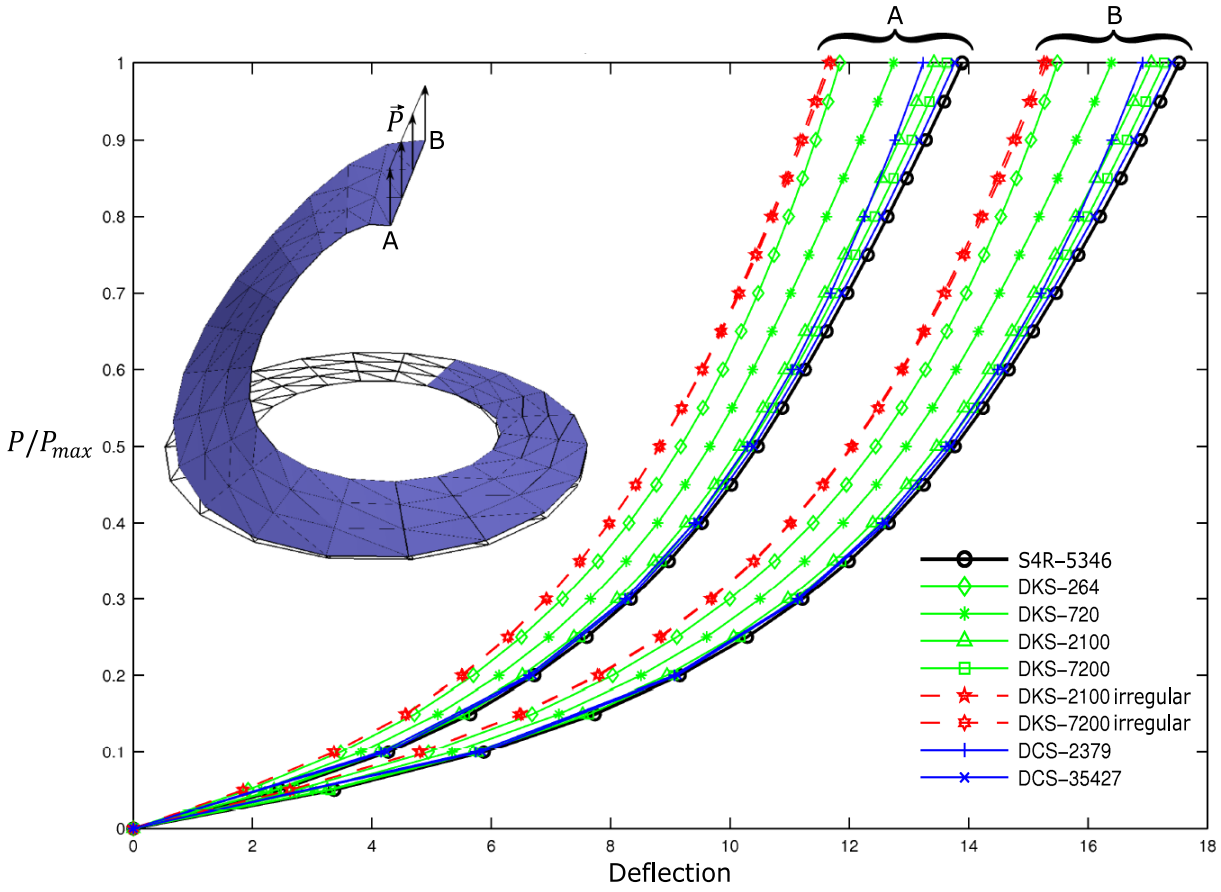

Figure 7.1 Load-deflection curves obtained with different shell models during the slit annular plate test presented in Ref. [61]. One end of the plate is lifted with a distributed force $\vec{P}$, and the other end is clamped. The annulus has thickness of 0.03, its internal and external radii equal 6 and 10 respectively, the maximal applied force per unit length P_{max} equals 0.8, and the Young's modulus of the plate equals 21×10^6 (see Ref. [112]). The curves trace deflection of the points A and B on the lifted end of the plate. The numbers in the legend stand for the number of DOF in each model. All models are meshed with a regular mesh except for those marked as "irregular" in the legend. Adapted from Ref. [61] with permission from Fraunhofer Institute for Industrial Mathematics ITWM.

The fact that the DCS and the S4R models converge to the same results with a fine mesh suggests that a forward problem solver for the peen forming case can be efficiently implemented using either of these models. The advantage of the DCS model is that it is specifically designed for the case of internal loads [59], so it does not require formulating an external load stimulus. In addition, the DCS model naturally takes into account the geometric nonlinearities due to a nonlinear elastic energy formulation (Eqn. 2.17). Finally, the DCS model incorporates the concepts of fundamental forms and curvatures, so that no supplementary commands have to be applied to estimate these values. This is an important advantage for implementation of the *inverse* problem solver (see Chapter 4), which operates with the geometrical properties of elements. However, the DCS model is resolution-dependent, which means that a finer mesh could slightly decrease the simulation error obtained during the experimental validation campaign presented in Chapter 6.

7.2 Energy-based inverse problem solver: advantages and challenges

For the sake of clarity, in this section we use a general notation applicable to all multilayer eigenstrain models. The reasoning presented in this section was developed by prof. Wim M. van Rees[1] and communicated in person on the 17$^{\text{th}}$ of April 2018.

Formulated in terms of the theory of non-Euclidean plates, the forward problem consists in finding the final configuration $\mathcal{C}_f$ induced by the rest configuration $\mathcal{R}$ prescribed to the plate. The plate adopts a $\mathcal{C}_f$ that minimizes its elastic energy E (see Chapter 4):

$$\mathcal{C}_f = \arg\min_{\mathcal{C}_c} E(\mathcal{R}, \mathcal{C}_c), \tag{7.1}$$

where $\mathcal{C}_c$ denotes a current configuration that the plate can adopt during reconfiguration. The fact that $\mathcal{C}_f$ corresponds to the global energetic minimum means that $\mathcal{C}_f$ is a state of equilibrium. The plate, however, can stay stressed in this configuration, which means that the elastic energy in its global minimum can be superior to zero. We solve the forward problem numerically using the L-BFGS algorithm [85], which minimizes E with respect to $\mathcal{C}_c$. During minimization, the gradients $\nabla_{\mathcal{C}_c} E$ are computed analytically [86].

Theoretically, the inverse problem can be formulated and solved in a similar way. This problem consists in finding the rest configuration $\mathcal{R}_{tar}$ that induces the target configuration $\mathcal{C}_{tar}$.

[1] Assistant Professor, Van Rees Lab, Department of Mechanical Engineering, Massachusetts Institute of Technology

We note that $\mathcal{C}_{tar}$ has to be a state of equilibrium, which enforces the following condition:

$$\mathcal{C}_{tar} = \arg\min_{\mathcal{C}_c} E(\mathcal{R}_{tar}, \mathcal{C}_c), \tag{7.2}$$

or equivalently

$$\nabla_{\mathcal{C}_c} E(\mathcal{R}_{tar}, \mathcal{C}_c)\Big|_{\mathcal{C}_c=\mathcal{C}_{tar}} = 0. \tag{7.3}$$

This condition is satisfied if $\mathcal{R}_{tar}$ is computed as:

$$\boxed{\mathcal{R}_{tar} = \arg\min_{\mathcal{R}} \left[\nabla_{\mathcal{C}_c} E(\mathcal{R}\ , \mathcal{C}_c)\Big|_{\mathcal{C}_c=\mathcal{C}_{tar}}\right].} \tag{7.4}$$

Eqn. 7.4 provides a concise energy-based formulation for the inverse problem solver. Same as the forward problem solver, it can be based on a numerical optimization algorithm. A possibility to compute the gradients $\nabla_{\mathcal{R}}\nabla_{\mathcal{C}_c} E$ algebraically is an advantage with respect to the methods proposed by Faucheux [25] and Miao et al. [5]. Moreover, the energy-based solver is more precise than the analytical solver presented in Chapter 4. Indeed, as we state in Chapter 4, "because of the local nature of the eigenstrain adjustment, the solution does not converge to the exact target shape but to a shape which is close to the target" (see Figures 4.9 and 4.10). The energy-based formulation 7.4 overcomes this problem, because the numerical optimization algorithms such as L-BFGS converge to the exact minimum of the optimized function.

Nonetheless, the energy-based approach has robustness issues. Thus, the target shape $\mathcal{C}_{tar}$ has to be the *exact* state of equilibrium, or in other words, $\mathcal{C}_{tar}$ has to be perfectly achievable in the framework of the implied eigenstrain model. Otherwise, if $\mathcal{C}_{tar}$ deviates from equilibrium, then the formulation 7.4 does not force the optimization algorithm to compute $\mathcal{R}_{tar}$ leading to the closest to $\mathcal{C}_{tar}$ state of equilibrium. This means that the result of computations in this case is barely predictable. Moreover, even if $\mathcal{C}_{tar}$ is a state of equilibrium, an optimization algorithm can get stuck at a local minimum of the optimized function on its way to the global minimum [7]. For these reasons, the robust analytical solver described in Chapter 4 was deemed more relevant for the peen forming application.

7.3 Choice of the eigenstrain formulation

The eigenstrain formulation characterizes local peening-induced loads with the help of two quantities: the total eigenstrain Γ and the first eigenstrain moment Γ^1 (see Section 2.1.3). The value of Γ determines the stretching force and the value of Γ^1 determines the bending

moment. Both quantities involve through-thickness integration, so the shape of the through-thickness eigenstrain profile does not influence the forming behavior. Therefore, we idealize the eigenstrain profile for the simulation purposes. An optimal idealized profile involves a low number of variables but allows to reproduce all practically achievable Γ and Γ^1 (see Section 2.2.1). There are two eigenstrain profiles that satisfy these requirements the best: the trilayer (Eqn. 2.10) and the bilayer (Eqn. 2.11). Each of them has its own advantages.

The trilayer profile is essentially the "step" profile (Eqn. 2.9) generalized for the case of double-side treatment. It reflects well the mechanical nature of the forming process because the eigenstrain in this formulation is concentrated near the surfaces (see Figure 2.3). This allows, in particular, to roughly estimate the residual stress induced by the treatment. Another benefit of the trilayer formulation is that the active layers do not superimpose. This simplifies the pattern segmentation algorithm and makes the segmentation more precise. Indeed, the formulation does not require to optimally split the eigenstrain into contributions made by the top and the bottom side peening regimes because these contributions are characterized with different variables (see Chapter 5).

On the other hand, the bilayer profile is characterized with a lower number of variables than the trialyer — two against four. Multiplication of this number by the number of finite elements in the numerical model gives the total quantity of variables for the inverse problem resolution. Consequently, this difference becomes significant if the inverse problem is solved through numerical optimization as in Ref. [5]. In addition, the fact the number of local variables characterizing the bilayer profile (two) corresponds to the number of peening-induced loads (Γ and Γ^1) is favourable for the inverse problem solver presented in Chapter 4. The solver is based on the comparison of the local stretching and curvature of the current shape with those of the target shape. This comparison locally provides two equations that are solved for the two eigenstrain variables, so the solution is exact. The use of the trilayer profile in this context requires the solution of two equations against four variables, so an optimization technique has to be involved. In this case the optimization has to be done on each iteration of the adjustment phase, while the bilayer profile faces this problem only when splitting the peening regimes during segmentation.

All in all, we conclude that the bilayer profile is more efficient in the general case due to a lower number of variables. However, if the treatment is applied from one side only, then the "step" profile is more advantageous. Indeed, it is also described by two variables but makes the simulation more realistic in terms of residual stress.

CHAPTER 8 CONCLUSION AND RECOMMENDATIONS

8.1 Summary of findings

This thesis aimed at modeling, optimizing and automating of the shot peen forming process. We developed a simulation software that allows to rapidly design an optimal forming procedure. The software is able to solve the forward and the inverse problems and to segment the peening pattern after the inverse problem resolution. We also conceived a workflow allowing to execute the designed procedure with an automated equipment.

Our simulation software combines the eigenstrain approach with the recent advances in the theory of non-Euclidean plates. The eigenstrain approach allows to model the peening-induced loads without simulating their mechanical cause, which is challenging and computationally expensive. The loads are represented as incompatible plastic strains (eigenstrains), and a plate loaded in this way falls within the definition of a non-Euclidean plate. The theory of non-Euclidean plates provides a concise energy-based formulation for the forward problem solver. We implemented it numerically using the constant-strain and constant-moment shell finite element model, which is described in Chapter 4. Additionally, the theory of non-Euclidean plates provides efficient theoretical instruments for the inverse problem resolution. Our inverse problem solver presented in Chapter 4 is iterative, and each iteration implies adjustment of the eigenstrain and resolution of the forward problem. The advantages of this solver include a precise initial guess, a low number of iterations required and an instantly executed adjustment phase. Therefore, the solver provides a precise solution within several minutes, that was proven during the numerical validation campaign described in Chapter 4.

A practically applicable peening pattern must consist of uniformly treated segments. To fulfill this requirement, we developed a pattern segmentation strategy. We apply it to the pattern provided by the inverse problem solver. The strategy consists of two stages: the first stage involves grouping or clustering, and the second stage implies correction of the checkerboard problem. The grouping algorithm described in Chapter 4 is applied if the set of peening regimes is fixed in advance and cannot be adjusted. In this case, the one-to-one correspondence between the peening regimes and the eigenstrains must be established in advance. On the other hand, the clustering procedure presented in Chapter 5 automatically computes optimal eigenstrains and segments the pattern into zones prescribed with a uniform eigenstrain. The peening regimes can be adjusted afterwards to reproduce the prescribed eigenstrains. In the absence of practical restrictions, the use of clustering is preferable to grouping, because it allows to avoid the trial and error search for optimal peening regimes.

Both grouping and clustering algorithms can produce local segmentation errors, which are corrected by the filtering algorithm formulated in Chapter 5.

The influence of grouping, clustering and filtering algorithms on the quality of the inverse problem solution was examined during the numerical validation campaign presented in Chapters 4 and 5. It was revealed that the influence of all three algorithms can be either positive or negative. However, the error for a series of tests stayed in the same range as before the application of the algorithms. The clustering algorithm is specifically designed to minimize the influence on the solution, while in the grouping case this influence can only be minimized by a thorough adjustment of the fixed peening regimes. Concerning the filtering algorithm, we conclude that its application is optional and reserved for the cases when the segmentation errors are too numerous to be manually corrected.

The automated peen forming workflow described in Chapter 6 regroups the theoretical reasoning and the practical manipulations that are necessary for the shot peen forming automation. Namely, it explains in detail how to calibrate the simulation software, how to automatically convert a peening pattern into a program for the peening robot, how to efficiently execute this program and how to estimate the quality of forming. Particular attention is given to a simple yet efficient experimental procedure for establishing a one-to-one correspondence between the peening regimes and the eigenstrains. We also present a procedure for the eigenstrain anisotropy characterization, which enhances the quality of simulations. We applied the automated workflow to validate the simulation software and we characterized the simulation error. It was revealed that the error is caused by the practical aspects that did not correspond to the simulated peening conditions, so we propose ways to correct this discrepancy from the practical side. All in all, it was demonstrated that the proposed workflow allows to shape an airplane wing skin.

8.2 Limitations and recommendations for future work

Primarily, we suggest ways for improving the automated peen forming workflow in terms of precision and cost-effectiveness. Thus, the simulation error can be reduced by accounting for peening-induced stretching in the simulations and by decreasing the influence of clamping on the eigenstrain distribution. Also, the “wave” experiment that characterizes the eigenstrain anisotropy can be optimized in terms of precision and material waste. In addition to the workflow improvement, we propose to validate our clustering method experimentally and to adapt our simulation software for integrally stiffened panels.

Determination of the peening-induced stretching

In Chapter 6, we determine the peening-induced eigenstrains using small uniformly treated coupons. The coupons are made of the same material and have the same thickness as the full-size plates that are supposed to be shaped. We measure the deflections of the coupons and deduce the dimensionless load Γ_B, which is essentially a nondimensionalized version of the first eigenstrain moment Γ^1. Given that the peening-induced eigenstrain is small, we neglect the total eigenstrain Γ and set this value to zero. However, the value of Γ determines the peening-induced stretching, and its accurate determination can be necessary for high-precision industrial applications.

The relation between the peening-induced stretching and Γ can be found in a similar fashion as the relation between the peening-induced deflection and Γ_B, i.e., using numerical simulations. In turn, measurement of the stretching is a technical problem. If the coupon is treated from one side only, this can be done with a strain gauge installed on the untreated side. The mid-surface stretching can then be determined using the Kirchhoff-Love straight normals assumption and the data on deflection. The strain gauge provides high precision, but its implication requires special care and slows down the calibration process. Another approach, which was used in Ref. [68], consists in peening the coupon with the same intensity from both sides. Given that the coupon stays flat after such a treatment, its stretching can be measured with a caliper. This method is simpler but less precise, because a high-intensity treatment can significantly deform the edges of the coupon. Nonetheless, an optimal method for determination of the peening-induced stretching is yet to be found and tested.

Decreasing the influence of clamping

The forward problem solver validation presented in Chapter 6 revealed that the clamping forces created by the fixture influenced the local curvatures. The local curvatures are determined by local eigenstrains, so these forces essentially influenced the plastic flow. As a result, the eigenstrain induced in uniformly treated segments was higher along the borders of these segments than in the center. This effect was not taken into account during the numerical simulations and, consequently, increased the simulation error. Accounting for the influence of clamping on the plastic flow in the simulations is left for future studies.

The described problem can be mitigated from the practical side practice by decreasing the compressing force between the two parts of the removable frame. This would let the treated plate expand while preventing it from bending. In other words, this would eliminate the external contracting force acting in the plane of the plate, which is the main cause of problem

according to our hypothesis. The removable frame can also be modified to let the fixed plate expand freely. All in all, an optimal practical method for decreasing the influence of clamping during forming is yet to be found. It should be noted that industrial parts have larger sizes and thicknesses, so the magnitude of peening-induced expansion in this case is lower, and the described issue is less severe.

Optimizing characterization of the eigenstrain anisotropy

A uniformly peened aluminum plate demonstrates anisotropic bending behavior only when its size and the peening intensity are sufficiently large with respect to its thickness, i.e., when the dimensionless value $\overline{\Gamma}_B$ is sufficiently large [35]. Our experiments with uniformly treated 76×19 mm coupons cut in different directions with respect to the rolling direction (the L- and T- directions) have confirmed this observation (see Chapter 6). Accordingly, we characterized the eigenstrain anisotropy using 320×320 mm plates, for which the value of $\overline{\Gamma}_B$ was larger. We applied the "wave" pattern thus forcing the plates to bend either in the L- or in the T- direction. The plates have exhibited an anisotropic behavior, but their size was defined by the size of the removable frame. Hence, this size can be decreased in the future to reduce the material waste. This would involve a similar series of simulations as those conducted in Ref. [35]. Namely, one has to find the minimal value of $\overline{\Gamma}_B$ that reveals the eigenstrain anisotropy for plates treated with the "wave" pattern. Moreover, the pattern itself can be further optimized to explicitly reveal the effect of anisotropy on small plates. Finally, we suggest repeating the anisotropy measurement (see Chapter 6) several times with the same parameters to improve the stability of results.

Experimental validation of the clustering method

The inverse problem solver validation described in Chapter 6 showed that the predefined peening regimes were not perfectly suited for shaping the upper skin of the NACA profile. Namely, the induced eigenstrain was not high enough to precisely shape the front edge. In this connection, we developed the clustering algorithms presented in Chapter 5 to avoid this issue in the future. Our clustering algorithms compute the eigenstrain that *must* be induced by each of the peening regimes involved. However, an optimal strategy for choosing the actual peening parameters as a function of the prescribed eigenstrain is yet to be found. This can be done by trial and error measurements, but it would be more efficient to establish a continuous relation between one of the peening parameters and the induced eigenstrain. For example, the peening regimes #1-5 described in Chapter 6 differ only by the peening pressure. This parameter allows to vary the peening intensity and, consequently, the induced

eigenstrain. Hence, a continuous function would allow to easily find the pressure inducing the required eigenstrain, but this hypothesis is yet to be verified in practice.

Modeling of shot peen forming of stiffened plates

Our forward and inverse problem solvers described in Chapter 4 are designed for the case of smooth plates and shells. In the industry, however, shot peen forming is also applied for shaping integrally stiffened panels, such as that shown in in Figure 8.1. Although the DCS model that we use admits variation of thickness and Young's modulus over the surface (see Appendix to Ref. [7]), it does not allow to include the stiffeners in the structure directly. More precisely, the formulation is based on the concept of edge directors, which is not adapted for the case when *three* finite elements share the same edge.

The presence of stiffeners can potentially be simulated by the prescription of a large thickness or a large Young's modulus to the elements situated along the line of attachment of stiffeners. This hypothesis is yet to be verified. Given that the stiffeners can be shot peened themselves, such a formulation would require additional calibration steps in order to simulate this effect. Otherwise, the DCS model can be replaced with a more standard finite element formulation with 6 degrees of freedom for each vertex, which corresponds for example to the S3R or the S4R models in ABAQUS software. In this case, the stiffeners can be physically included in the structure, but the inverse problem solver has to be reformulated in order to operate with the eigenstrains induced in the stiffeners.

Figure 8.1 An integrally stiffened skin panel undergoing automated shot peening treatment. © KSA Kugelstrahlzentrum Aachen GmbH. Reproduced with permission.

REFERENCES

[1] S. Ramati, S. Kennerknecht, and G. Levasseur, "Single piece wing skin utilization via advanced peen forming technologies," *Proceedings of the 7th International Conference on Shot Peening (ICSP7), Warsaw*, 1999.

[2] P. A. Faucheux, F. P. Gosselin, and M. Lévesque, "Simulating shot peen forming with eigenstrains," *Journal of Materials Processing Technology*, vol. 254, pp. 135–144, 2018.

[3] R. VanLuchene and E. Cramer, "Numerical modeling of a wing skin peen forming process," *Journal of materials engineering and performance*, vol. 5, no. 6, pp. 753–760, 1996.

[4] T. Wang, M. Platts, and J. Wu, "The optimisation of shot peen forming processes," *journal of materials processing technology*, vol. 206, no. 1-3, pp. 78–82, 2008.

[5] H. Y. Miao, M. Lévesque, and F. P. Gosselin, "Shot peen forming pattern optimization to achieve cylindrical and saddle target shapes: The inverse problem," *CIRP Journal of Manufacturing Science and Technology*, vol. 36, pp. 67–77, 2022.

[6] W. Siguerdidjane, F. Khameneifar, and F. P. Gosselin, "Efficient planning of peen-forming patterns via artificial neural networks," *Manufacturing Letters*, vol. 25, pp. 70–74, 2020.

[7] W. M. van Rees, E. Vouga, and L. Mahadevan, "Growth patterns for shape-shifting elastic bilayers," *Proceedings of the National Academy of Sciences*, vol. 114, no. 44, pp. 11 597–11 602, 2017.

[8] R. Kopp and J. Schulz, "Optimising the double-sided simultaneous shot peen forming," *Proceedings of the 8th International Conference on Shot Peening (ICSP8), Garmisch-Partenkirchen*, pp. 227–233, 2002.

[9] L. Rokach and O. Maimon, "Clustering methods," in *Data mining and knowledge discovery handbook*. Springer, 2005, pp. 321–352.

[10] F. Wüstefeld, W. Linnemann, and S. Kittel, "Towards peen forming process automation," *Shot Peening*, p. 44, 2006.

[11] Metal Improvement Company, "Shot Peening Applications," 2020. [Online]. Available: https://www.cwst.co.uk/wp-content/uploads/2020/01/new-english-green-book.pdf

[12] D. Kirk, "Shot peening," *Aircraft engineering and aerospace technology*, 1999.

[13] S. Sharp, M. Ashby, and N. Fleck, "Material response under static and sliding indentation loads," *Acta metallurgica et materialia*, vol. 41, no. 3, pp. 685–692, 1993.

[14] L. Samuels and T. Mulhearn, "An experimental investigation of the deformed zone associated with indentation hardness impressions," *Journal of the Mechanics and Physics of Solids*, vol. 5, no. 2, pp. 125–134, 1957.

[15] H. Wohlfahrt, "The influence of peening conditions on the resulting distribution of residual stress," in *Proceedings of the Second International Conference on Shot Peening*. Chicago, 1984, pp. 316–331.

[16] M. Guagliano, "Relating almen intensity to residual stresses induced by shot peening: a numerical approach," *Journal of Materials Processing Technology*, vol. 110, no. 3, pp. 277–286, 2001.

[17] "Test strip, holder, and gage for shot peening," SAE International, Standard SAE J442, 2009.

[18] A. Gariépy, "Finite element modelling of shot peening and peen forming processes and characterisation of peened aa2024-T351 aluminium alloy," Ph.D. dissertation, École Polytechnique de Montréal, 2012.

[19] J. Cammett, "Shot peening coverage — the real deal," *The Shot Peener magazine*, vol. 21, no. 3, pp. 8–14, 2007.

[20] E. de los Rios *et al.*, "Fatigue crack initiation and propagation on shot-peened surfaces in a316 stainless steel," *International Journal of Fatigue*, vol. 17, no. 7, pp. 493–499, 1995.

[21] T. Klotz *et al.*, "Surface characteristics and fatigue behavior of shot peened inconel 718," *International Journal of Fatigue*, vol. 110, pp. 10–21, 2018.

[22] Z. Chen, F. Yang, and S. Meguid, "Realistic finite element simulations of arc-height development in shot-peened almen strips," *Journal of Engineering Materials and Technology*, vol. 136, no. 4, 2014.

[23] A. M. Korsunsky, "The modelling of residual stresses due to surface peening using eigenstrain distributions," *The Journal of Strain Analysis for Engineering Design*, vol. 40, no. 8, pp. 817–824, 2005.

[24] A. M. Korsunsky, "Eigenstrain analysis of residual strains and stresses," *The Journal of Strain Analysis for Engineering Design*, vol. 44, no. 1, pp. 29–43, 2009.

[25] P. A. Faucheux, "Simulating shot peen forming with eigenstrains," PhD dissertation, Polytechnique Montreal, December 2019.

[26] S. Al-Hassani, "An engineering approach to shot peening mechanics," *Proceedings of the 2nd International Conference on Shot Peening (ICSP2), Chicago*, 1984.

[27] J. F. Flavenot, "Niku-lari, a," *La Mesure des Contraintes Residuelles: Methode de la (Fleche) Methode de la (Source de Contraintes). Les Mémoi-res Techniques du CETIM (31)*, 1977.

[28] T. Mura, "Micromechanics of defects in solids. vol. 3," *Springer Science & Business Media*, vol. 580, p. 21, 1987.

[29] T. Wang, M. Platts, and A. Levers, "A process model for shot peen forming," *Journal of Materials Processing Technology*, vol. 172, no. 2, pp. 159–162, 2006.

[30] S. Al-Hassani, "The shot peening of metals — Mechanics and structures," *SAE transactions*, pp. 4513–4525, 1982.

[31] J. Badreddine *et al.*, "Cad based model of ultrasonic shot peening for complex industrial parts," *Advances in Engineering Software*, vol. 76, pp. 31–42, 2014.

[32] S. Rouquette *et al.*, "Coupled thermo-mechanical simulations of shot impacts: Effects of the temperature on the residual stress field due to shot-peening," *Journal of Materials Processing Technology*, vol. 209, no. 8, pp. 3879–3886, 2009.

[33] A. Benallal, P. Le Gallo, and D. Marquis, "An experimental investigation of cyclic hardening of 316 stainless steel and of 2024 aluminium alloy under multiaxial loadings," *Nuclear Engineering and Design*, vol. 114, no. 3, pp. 345–353, 1989.

[34] M. B. Prime, "Amplified effect of mild plastic anisotropy on residual stress and strain anisotropy," *International Journal of Solids and Structures*, vol. 118, pp. 70–77, 2017.

[35] H. Y. Miao, M. levesque, and F. Gosselin, "Peen forming and stress peen forming of 2024-T3 aluminum sheets. Part 2: eigenstrain analysis," Apr 2021.

[36] F. Bron and J. Besson, "A yield function for anisotropic materials application to aluminum alloys," *International Journal of Plasticity*, vol. 20, no. 4-5, pp. 937–963, 2004.

[37] K. Han *et al.*, "A combined finite/discrete element simulation of shot peening processes — Part II: 3D interaction laws," *Engineering Computations*, 2000.

[38] K. Han, D. Owen, and D. Peric, "Combined finite/discrete element and explicit/implicit simulations of peen forming process," *Engineering Computations*, 2002.

[39] T. Wang, J. Platts, and A. Levers, "Finite element impact modelling for shot peen forming," *Proceedings of the 8th international conference on shot peening, Garmisch-Partenkirchen, Germany*, 2002.

[40] X. Kang, T. Wang, and J. Platts, "Multiple impact modelling for shot peening and peen forming," *Proceedings of the Institution of Mechanical Engineers, Part B: Journal of engineering manufacture*, vol. 224, no. 5, pp. 689–697, 2010.

[41] S. Zhang *et al.*, "High-energy synchrotron X-ray analysis of residual plastic strains induced in shot-peened steel plates," *The Journal of Strain Analysis for Engineering Design*, vol. 43, no. 4, pp. 229–241, 2008.

[42] M. E. Fitzpatrick *et al.*, "Determination of residual stresses by X-ray diffraction." 2005.

[43] N. Rossini *et al.*, "Methods of measuring residual stresses in components," *Materials & Design*, vol. 35, pp. 572–588, 2012.

[44] A. M. Korsunsky, "Residual elastic strain due to laser shock peening: modelling by eigenstrain distribution," *The Journal of Strain Analysis for Engineering Design*, vol. 41, no. 3, pp. 195–204, 2006.

[45] E. Efrati, E. Sharon, and R. Kupferman, "Elastic theory of unconstrained non-euclidean plates," *Journal of the Mechanics and Physics of Solids*, vol. 57, no. 4, pp. 762–775, 2009.

[46] M. Pezzulla *et al.*, "Geometry and mechanics of thin growing bilayers," *Soft matter*, vol. 12, no. 19, pp. 4435–4442, 2016.

[47] E. Efrati, E. Sharon, and R. Kupferman, "The metric description of elasticity in residually stressed soft materials," *Soft Matter*, vol. 9, no. 34, pp. 8187–8197, 2013.

[48] A. Levers and A. Prior, "Finite element analysis of shot peening," *Journal of Materials Processing Technology*, vol. 80, pp. 304–308, 1998.

[49] T. Chaise *et al.*, "Modelling of multiple impacts for the prediction of distortions and residual stresses induced by ultrasonic shot peening (usp)," *Journal of Materials Processing Technology*, vol. 212, no. 10, pp. 2080–2090, 2012.

[50] T. Chaise, "Mechanical simulation using a semi analytical method: from elasto-plastic rolling contact to multiple impacts," Ph.D. dissertation, INSA de Lyon, 2011.

[51] D. Gallitelli *et al.*, "Simulation of shot peening: From process parameters to residual stress fields in a structure," *Comptes Rendus Mécanique*, vol. 344, no. 4, pp. 355–374, 2016, computational simulation of manufacturing processes.

[52] L. Grasty and C. Andrew, "Shot peen forming sheet metal: finite element prediction of deformed shape," *Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture*, vol. 210, no. 4, pp. 361–366, 1996.

[53] P. A. Faucheux *et al.*, "Peen forming and stress peen forming of rectangular 2024–T3 aluminium sheets: Curvatures, natural curvatures and residual stresses," *Strain*, vol. 58, no. 2, p. e12405, 2022.

[54] E. Sharon and E. Efrati, "The mechanics of non-euclidean plates," *Soft Matter*, vol. 6, no. 22, pp. 5693–5704, 2010.

[55] P. G. Ciarlet, "An introduction to differential geometry with applications to elasticity," *Journal of Elasticity*, vol. 78, no. 1, pp. 1–215, 2005.

[56] G. A. Holzapfel, "Nonlinear solid mechanics: A continuum approach for engineering science," *Meccanica*, vol. 37, pp. 489–490, 2000.

[57] M. Pezzulla *et al.*, "Morphing of geometric composites via residual swelling," *Soft Matter*, vol. 11, no. 29, pp. 5812–5820, 2015.

[58] M. Pezzulla *et al.*, "Curvature-driven morphing of non-euclidean shells," *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, vol. 473, no. 2201, p. 20170087, 2017.

[59] W. M. van Rees *et al.*, "Mechanics of biomimetic 4D printed structures," *Soft matter*, vol. 14, no. 43, pp. 8771–8779, 2018.

[60] L. Morley, "The constant-moment plate-bending element," *Journal of Strain Analysis*, vol. 6, no. 1, pp. 20–24, 1971.

[61] C. Weischedel *et al.*, "Construction of discrete shell models by geometric finite differences." Fraunhofer ITWM, Kaiserslautern, Germany, 05 2012.

[62] A. Quaglino, "A framework for creating low-order shell elements free of membrane locking," *International Journal for Numerical Methods in Engineering*, vol. 108, no. 1, pp. 55–75, 2016.

[63] H.-Y. Chen *et al.*, "Physical simulation of environmentally induced thin shell deformation," *ACM Transactions on Graphics (TOG)*, vol. 37, no. 4, pp. 1–13, 2018.

[64] F. Momeni *et al.*, "A review of 4D printing," *Materials & design*, vol. 122, pp. 42–79, 2017.

[65] A. S. Gladman *et al.*, "Biomimetic 4D printing," *Nature materials*, vol. 15, no. 4, pp. 413–418, 2016.

[66] Y. Klein, E. Efrati, and E. Sharon, "Shaping of elastic sheets by prescription of non-euclidean metrics," *Science*, vol. 315, no. 5815, pp. 1116–1120, 2007.

[67] W. Wang *et al.*, "Transformative appetite: shape-changing food transforms from 2D to 3D by water interaction through cooking," *Proceedings of the 2017 CHI conference on human factors in computing systems*, pp. 6123–6132, 2017.

[68] R. VanLuchene, J. Johnson, and R. Carpenter, "Induced stress relationships for wing skin forming by shot peening," *Journal of Materials Engineering and Performance*, vol. 4, no. 3, pp. 283–290, 1995.

[69] P. E. Gill, W. Murray, and M. H. Wright, *Practical optimization*. SIAM, 2019.

[70] Y. Essa *et al.*, "Numerical optimization tool of peen-forming process for aerospace industry," 11 2016.

[71] I. I. Dikin, "Iterative solution of problems of linear and quadratic programming," *Soviet Mathematics. Doklady*, vol. 8, pp. 674–675, 1967.

[72] E. J. Haug, K. K. Choi, and V. Komkov, "Design sensitivity analysis of structural systems," 1986.

[73] J. M. Pajot *et al.*, "Design of patterned multilayer films with eigenstrains by topology optimization," *International journal of solids and structures*, vol. 43, no. 6, pp. 1832–1853, 2006.

[74] T. H. Cormen *et al.*, *Introduction to algorithms*. MIT press, 2009.

[75] M. P. Bendsoe and O. Sigmund, *Topology optimization: theory, methods, and applications*. Springer Science & Business Media, 2003.

[76] D. Xu and Y. Tian, "A comprehensive survey of clustering algorithms," *Annals of Data Science*, vol. 2, no. 2, pp. 165–193, 2015.

[77] S. Lloyd, "Least squares quantization in pcm," *IEEE transactions on information theory*, vol. 28, no. 2, pp. 129–137, 1982.

[78] H. Diep, H. Bae, and M. Ramulu, "Characterization of manual shot peening process: Preliminary results," *Proceedings of the 10th International Conference on Shot Peening (ICSP10), Tokyo*, 2008.

[79] A. Friese, "Ksa develops new automated peen forming for wing skins," *The Shot Peener magazine*, vol. 16, 2006.

[80] S. Kittel *et al.*, "Tight tolerance peen forming with on-line shape control," *Proceedings of the 7th International Conference on Shot Peening (ICSP7), Warsaw*, pp. 301–307, 1999.

[81] M. Luo *et al.*, "Efficient process planning of laser peen forming for complex shaping with distributed eigen-moment," *Journal of Materials Processing Technology*, vol. 279, p. 116588, 2020.

[82] H. Aharoni *et al.*, "Universal inverse design of surfaces with thin nematic elastomer sheets," *Proceedings of the National Academy of Sciences*, vol. 115, no. 28, pp. 7206–7211, 2018.

[83] M. B. Prime and M. R. Hill, "Residual stress, stress relief, and inhomogeneity in aluminum plate," *Scripta Materialia*, vol. 46, no. 1, pp. 77–82, 2002.

[84] M. Spivak, *A comprehensive introduction to differential geometry*. Boston: Publish or Perish, 1975, vol. 3.

[85] D. C. Liu and J. Nocedal, "On the limited memory bfgs method for large scale optimization," *Mathematical programming*, vol. 45, no. 1, pp. 503–528, 1989.

[86] R. Tamstorf and E. Grinspun, "Discrete bending forces and their jacobians," *Graphical models*, vol. 75, no. 6, pp. 362–370, 2013.

[87] W. M. van Rees, "Code accompanying the 2018 soft matter paper "mechanics of biomimetic 4D printed structures"," https://github.com/wimvanrees/growth_SM2018, 2018.

[88] B. Lévy *et al.*, "Least squares conformal maps for automatic texture atlas generation," *ACM transactions on graphics (TOG)*, vol. 21, no. 3, pp. 362–371, 2002.

[89] K. Hornauer and W. Kohler, "Development of the peen forming process for spherical shaped components," in *Proceedings of the 4th International Conference on Shot Peening (ICSP4), Tokyo*, vol. 500, 1990, pp. 585–594.

[90] L. D. Hoffmann, G. L. Bradley, and K. H. Rosen, *Calculus for business, economics, and the social and life sciences*. McGraw-Hill, 2004.

[91] P. J. Selvapeter and W. Hordijk, "Cellular automata for image noise filtering," in *2009 World Congress on Nature & Biologically Inspired Computing (NaBIC)*. IEEE, 2009, pp. 193–197.

[92] M. Zawidzki, "Application of semitotalistic 2D cellular automata on a triangulated 3D surface," *International Journal of Design & Nature and Ecodynamics*, vol. 6, pp. 34–51, 01 2011.

[93] J. Merino *et al.*, *Ariane 6-Tanks and structures for the new european launcher*. Deutsche Gesellschaft für Luft-und Raumfahrt-Lilienthal-Oberth eV, 2017.

[94] A.-M. Yan and I. Klappka, "Springback in stretch forming process of aeronautic panel production by finite element simulation," *International Journal of Material Forming*, vol. 1, pp. 201–204, 04 2008.

[95] S. Kurukuri *et al.*, "Simulation of stretch forming with intermediate heat treatments of aircraft skins: A physically based modeling approach," *International Journal of Material Forming*, vol. 4, 06 2011.

[96] C. Russig *et al.*, "Shot peen forming of fiber metal laminates on the example of glare (r)," *International Journal of Material Forming*, vol. 7, 12 2014.

[97] H. Y. Miao *et al.*, "Residual stress measurements of conventional and stress peen formed 2024-T3 aluminum sheets," Apr 2021.

[98] V. Sushitskii *et al.*, "Determination of optimal shot peen forming patterns using the theory of non-euclidean plates," 2021.

[99] A. Levers and A. Prior, "Finite element analysis of shot peening," *Journal of Materials Processing Technology*, vol. 80, pp. 304–308, 1998.

[100] V. K. Barnwal *et al.*, "Effect of plastic anisotropy on forming behavior of AA-6061 aluminum alloy sheet," *The Journal of Strain Analysis for Engineering Design*, vol. 51, no. 7, pp. 507–517, 2016.

[101] F. Bron and J. Besson, "A yield function for anisotropic materials application to aluminum alloys," *International Journal of Plasticity*, vol. 20, no. 4-5, pp. 937–963, 2004.

[102] B. O'neill, *Elementary differential geometry*. Academic Press: Elsevier, 2006.

[103] M. P. Do Carmo, *Differential geometry of curves and surfaces: revised and updated second edition*. New York: Courier Dover Publications, 2016.

[104] G. Garg, V. Kuts, and G. Anbarjafari, "Digital twin for fanuc robots: Industrial robot programming and simulation using virtual reality," *Sustainability*, vol. 13, p. 10336, 09 2021.

[105] R. T. Rockafellar and R. J.-B. Wets, *Variational analysis*. Berlin, Heidelberg: Springer Science & Business Media, 2009, vol. 317.

[106] S. Umeyama, "Least-squares estimation of transformation parameters between two point patterns," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 13, no. 4, pp. 376–380, 1991.

[107] D. Kirk, "The curvature of peened almen strips and its applications," *The Shot Peener magazine*, vol. 32, no. 4, 2018.

[108] H. Li *et al.*, "The shot peen forming of fiber metal laminates based on the aluminum-lithium alloy: Deformation characteristics," *Composites Part B: Engineering*, vol. 158, pp. 279–285, 2019.

[109] J. Moran, *An introduction to theoretical and computational aerodynamics*. Courier Corporation, 2003.

[110] M. Gärdsback and G. Tibert, "A comparison of rotation-free triangular shell elements for unstructured meshes," *Computer Methods in Applied Mechanics and Engineering*, vol. 196, no. 49, pp. 5001–5015, 2007.

[111] E. Grinspun *et al.*, "Computing discrete shape operators on general meshes," in *Computer Graphics Forum*, vol. 25, no. 3. Wiley Online Library, 2006, pp. 547–556.

[112] K. Sze, X. Liu, and S. Lo, "Popular benchmark problems for geometric nonlinear analysis of shells," *Finite Elements in Analysis and Design*, vol. 40, no. 11, pp. 1551–1569, 2004.