Titre: Title:	Segmentation of peen forming patterns using k-means clustering
Auteurs: Authors:	Vladislav Sushitskii, Hongyan Miao, Martin Lévesque, & Frederick Gosselin
Date:	2024
Type:	Article de revue / Article
Référence: Citation:	Sushitskii, V., Miao, H., Lévesque, M., & Gosselin, F. (2024). Segmentation of peen forming patterns using k-means clustering. Journal of Manufacturing Processes, 119, 867-877. https://doi.org/10.1016/j.jmapro.2024.04.009

URL de PolyPublie: PolyPublie URL:	https://publications.polymtl.ca/58061/
Version:	Version officielle de l'éditeur / Published version Révisé par les pairs / Refereed
Conditions d'utilisation: Terms of Use:	Creative Commons Attribution 4.0 International (CC BY)

Titre de la revue: Journal Title:	Journal of Manufacturing Processes (vol. 119)
Maison d'édition: Publisher:	Elsevier
URL officiel: Official URL:	https://doi.org/10.1016/j.jmapro.2024.04.009
Mention légale: Legal notice:	©2024 The Authors. Published by Elsevier Ltd on behalf of The Society of Manufacturing Engineers. This is an open access article under the CC BY license (http://creativecommons.org/licenses/by/4.0/).

Journal of Manufacturing Processes 119 (2024) 867–877

Contents lists available at ScienceDirect

Journal of Manufacturing Processes

journal homepage: www.elsevier.com/locate/manpro

Research article

Segmentation of peen forming patterns using k-means clustering

Vladislav Sushitskii [a,b,*], Hong Yan Miao [a,b], Martin Lévesque [a], Frédérick P. Gosselin [a]

[a] Laboratory for Multiscale Mechanics (LM2), Department of Mechanical Engineering, Polytechnique Montreal, 2500 Chemin de Polytechnique, Montreal, H3T1J4, Quebec, Canada
[b] Aluminium Research Centre – REGAL, 1065 Avenue de la Medecine, Quebec, G1V0A6, Quebec, Canada

ARTICLE INFO

Keywords:
Segmentation
Clustering
Filtering
k-means
Peen forming

ABSTRACT

Shot peen forming is a prominent manufacturing process for shaping aircraft wing skins, which mostly relies on manual intervention, where operators use their expertise to select appropriate peening parameters and apply them in a specific pattern. This paper introduces a crucial advancement towards automating shot peen forming —a segmentation strategy designed to partition peen forming patterns into uniformly treated zones. Unlike existing optimization methods that necessitate predefined peening treatments, our segmentation strategy automatically identifies optimal treatment parameters for each segment. Our approach comprises a novel clustering algorithm, which divides the pattern into segments, and a noise filtering algorithm that eliminates excessively small segments. The clustering algorithm is a unique adaptation of the k-means method, which considers interconnected centroids due to the coupling of the effects of the top and bottom treatments of the part. The filtering algorithm leverages cellular automata principles. Both algorithms underwent numerical testing using 200 randomly generated test cases. The results indicate that the segmentation strategy consistently maintained forming error within an acceptable range, and remarkably, reduced the forming error in 67 out of 200 cases. This segmentation strategy can seamlessly integrate with existing shape optimization tools and a peening treatment library, leading to a fully automated shot peen forming system. The source code for our algorithms is publicly available on GitHub, fostering accessibility and further research in this domain.

1. Introduction

Fabricating aircraft and rockets requires forming large metal plates with high precision. These plates constitute, for example, wing skins [1], fuselage shells [2] or fuel tank segments [3]. The outlined components often exhibit a nonuniform double curvature, and a cost-effective shaping method that delivers such forms is shot peen forming [4]. The treatment can be applied to both sides of the plate with a variable intensity, so developed curvatures can be locally controlled by altering the peened segments and the peening parameters.

Shot peen forming is still largely performed manually because it relies on the operator's intuition and expertise in shaping [5,6]. However, an essential element of process automation is a numerical simulation tool that can be integrated with a peening robot. Several simulation tools were developed in-house by industrial companies, but they are not publicly accessible [7]. Moreover, in the public scientific literature, no method exists for selecting shot peen forming treatments based on a given peen forming pattern. Here, we propose a novel clustering-based method for automatically selecting a peening treatment, which can be combined with the pattern optimization method published earlier [8]. No comparison is made with other methods for segmenting the peen forming patterns in the literature, because none exists that we are aware of.

In terms of peen forming, the problem of defining a necessary treatment to achieve a predefined target shape is called the *inverse problem*. The solution to the inverse problem in peen forming is a map of local peening parameters, which we term as *regimes* throughout the paper. These regimes are mapped over the initial geometry of the treated plate, which is called the *peen forming pattern*. For instance, Fig. 1 shows two peen forming patterns that allow shaping a flat square plate into a (a) target spherical shape: (b) one with continuously varying intensity or an infinity of regimes; (c) and one discrete regimes. The first steps towards numerical resolution of the inverse problem were made by VanLuchene et al. [9] and VanLuchene and Cramer [10], who simulated the effect of peening using the peening-induced forces. Later, two [11,12] presented two inverse problem solvers operating directly with the peening parameters. In all these cases, the peening parameters were experimentally related to the peening-induced forces [10] or curvatures [11,12]. The inverse problem can also be solved through formulating the peen forming pattern in terms of the peening-induced residual stresses [13]. In turn, the residual stresses can be related to

[*] Corresponding author.
E-mail addresses: vladislav.sushitskii@polymtl.ca (V. Sushitskii), frederick.gosselin@polymtl.ca (F.P. Gosselin).

https://doi.org/10.1016/j.jmapro.2024.04.009
Received 21 February 2024; Received in revised form 31 March 2024; Accepted 2 April 2024
Available online 10 April 2024

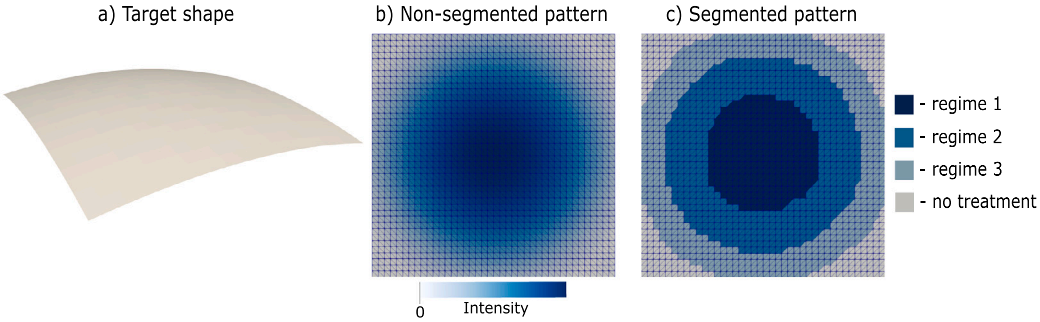


Fig. 1. Examples of a non-segmented and of a segmented peen forming pattern. Both patterns make a flat square plate deform into a spherical patch (a) when applied from both sides of the plate. The non-segmented pattern (b) prescribes a gradual variation of the peening intensity over the plate, which is hardly reproducible with peening equipment. The segmented pattern (c) is reproducible because it consists of uniformly treated segments. Both patterns are discretized with triangular finite elements, and the effect of their application can be simulated using the eigenstrain approach. For this, one must establish in advance a one-to-one correspondence between the peening regimes or the intensities, and the induced eigenstrains.

the peening parameters through direct impact simulations [14,15] or through local analytical impact models coupled with global numerical simulations [16].

Another strategy for the peen forming simulation involves formulating the peen forming pattern in terms of local eigenstrains [17,18]. The eigenstrain represents the plastic strain induced by the treatment, which triggers deformation when introduced in the component. The advantage of eigenstrains is that they are independent of the plate geometry, unlike the peening-induced forces, stresses or curvatures [19,20]. Moreover, the eigenstrains can be assumed constant during reconfiguration of the component, which was proven by means of numerical simulations and experiments with 76×19 mm shot peened metal coupons [21,22]. With the eigenstrain approach, the inverse problem can be solved using a numerical optimization algorithm [6], an artificial neural network [23] or the theory of non-Euclidean plates [8]. In turn, a one-to-one correspondence between the peening regimes and the induced eigenstrain can be established using experiments [19], small-scale numerical simulations [24,25], or a combination of both [26]. It is, however, imperative to hold the plate flat during peening and to ascertain the absence of stresses and strains in the initial state of the plate, because the established correspondence can be altered otherwise [6].

The eigenstrain approach also allows to simulate other forming processes that induce plastic strain in metal components, such as laser peen forming [27] or English wheel [28]. Moreover, eigenstrains allow to represent a peen formed plate as a laminate subjected to incompatible expansion or contraction of its layers [29], so this approach bridges the gap between peen forming and fabrication of laminated shape-shifting structures. For example, the inverse problem solver presented by Miao et al. [6] works both for peen forming and for patterned multilayer films [30], and the solver examined by Sushitskii et al. [8] uses the results developed in the field of 4D printed elastic sheets [31,32]. The practical applicability of the inverse problem solver presented in [8] was examined in [33], where it participated as an essential part of the automated peen forming process.

The numerical inverse problem resolution involves the discretization of the initial geometry with shell finite elements. The computed peen forming pattern is therefore a discrete map. In the general case, the eigenstrains prescribed to each element vary from one element to another, as it is illustrated in Fig. 1(b). On the other hand, the practical conditions of physically applying the peening treatment limit this spatial variation. Thus, the maximal eigenstrain that can be prescribed to a segment is limited by the equipment capacity. In addition, the peening parameters cannot be varied gradually over the plate. Moreover, each modification of peening parameters during treatment slows down the shaping process, which is especially the case for changing the media [34]. These reasons suggest that the pattern must consist of segments formed by multiple elements that are prescribed with the same eigenstrain. For a simpler and easier process, the segmented peen forming pattern must involve the fewest possible peening regimes while preserving the precision of the final shape.

Fig. 1(c) presents an example of a segmented pattern. In practice, such segmented patterns are usually applied using masks [35,36]. 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 problem of segmentation of the eigenstrain pattern is addressed by Miao et al. [6] and Siguerdidjane et al. [23]. The presented inverse problem solvers operate with one available eigenstrain magnitude, so the peen forming pattern is a set of treated and untreated segments. Miao et al. [6] achieved this effect by limiting the overall treated area. This condition forces the optimization algorithm to assign the maximal eigenstrain magnitude to the treated elements and thus penalizes the intermediate eigenstrain values. On the other hand, the neural network presented by Siguerdidjane et al. [23] naturally operates with one fixed eigenstrain magnitude, which is set during the training phase. Thus, the neural network computes a treatment probability for each element, and the elements with high probabilities are then assigned with the fixed eigenstrain magnitude. Although the examined solvers provide ready-to-use patterns, the use of only one eigenstrain magnitude limits the range of achievable target shapes.

The strategies for segmentation of the peen forming pattern presented by Sushitskii et al. [8], Miao et al. [6] and Siguerdidjane et al. [23] share a common drawback: they require having predefined peening regimes. This suggests that if the current peening regimes do not allow to achieve the target shape according to the simulations, then the best suitable regimes must be found by trial-and-error. Thus, if the segmentation is embedded in the inverse problem solver [6,23], then each new trial means restarting the inverse problem solver. Otherwise, if the segmentation is executed after the inverse problem resolution by means of grouping (see [8]), then the grouping algorithm must be relaunched with a new combination of peening regimes, which are defined manually. Hence, the necessity for having pre-defined peening regimes elongates the inverse problem solution and makes it less accurate, because the trial-and-error solution is not guaranteed to be the optimal.

Conversely, the problem of segmentation of a peen forming pattern is similar to the problem of clustering [37,38]. Clustering algorithms split a set of points into groups based on the point coordinates. In terms of the peen forming pattern, this means splitting the mesh into segments based on the eigenstrain values assigned to each finite element. Moreover, clustering algorithms are able to find a centroid for each cluster. With regard to the peen forming 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 able to prescribe the best suitable peening regime for each segment. This technique allowed, for example, to compute the pattern shown in Fig. 1(c) starting from the non-segmented pattern presented in Fig. 1(b).

We present in this paper a numerical strategy for segmenting the peen forming pattern without predefined peening regimes, which is designed as a post-processing tool for the inverse problem solver. The segmentation strategy relies on a novel modified k-means clustering algorithm [39], which is adapted for the peen forming case but can be applied to solve clustering problems in other domains. We chose to base our clustering algorithm on the k-means due to its high computation speed and its adaptability to sparse point sets [37,38]. The clustering algorithm is followed by a noise filtering algorithm that corrects local clustering errors and thus makes the pattern fully applicable. In turn, the practical applicability of the segmented patterns is examined by Sushitskii et al. [33].

2. Theoretical background

2.1. The bilayer eigenstrain formulation of the peen forming pattern

The eigenstrain ε prescribed to each finite element by an eigenstrain-based inverse problem solver is constant in the in-plane direction inside the element, but it varies along the through-thickness coordinate z. We assume that the treated material is isotropic and not pre-stressed, so that $\varepsilon_{xx}(z) = \varepsilon_{yy}(z) = \gamma(z)$ and $\varepsilon_{zz}(z) = -2\gamma(z)$, where x and y are the in-plane lagrangian coordinates.

A precise measurement of the through-thickness eigenstrain profile $\gamma(z)$ is challenging due to the stochastic nature of the shot projection [40]. Thus, the impacts superimpose with each other and are not necessarily normal to the surface. The shot are not perfectly spherical [35], and the peening parameters can fluctuate over time due to technical reasons. Finally, the impacts harden and heat the material [41], which alters its response to the subsequent indentations [42].

In order to overcome these complexities in the framework of peen forming simulation, we idealize the through-thickness eigenstrain profile in the way that it induces the same final shape as the original profile (when assuming plate theory, see [8]). In the general case of treatment from both sides, the simplest idealized profile involving the fewest variables is the bilayer profile γ_{bi}. This profile assumes that the plate consists of two equally thick layers assigned with the eigenstrains ε^t and ε^b as:

$$\gamma_{bi}(z) = \begin{cases} \varepsilon^t & \text{for} \quad 0 < z < \frac{h}{2}, \\ \varepsilon^b & \text{for} \quad -\frac{h}{2} < z < 0, \end{cases} \tag{1}$$

where h is the total plate thickness and z is the through-thickness coordinate having its origin at the mid-surface level. Fig. 2 illustrates the idealization of the through-thickness eigenstrain profile.

The bilayer eigenstrain profile is locally described with two variables $(\varepsilon^t, \varepsilon^b)$, and each of the two peening regimes applied from the top- and bottom-sides influences both ε^t and ε^b. Thus, the treatment from the top side induces positive ε^t and negative ε^b, while the treatment from the bottom side induces negative ε^t and positive ε^b. In case of simultaneous treatments from both sides, the values of ε^t and ε^b are superpositions of contributions made by the top and the bottom side peening regimes.

Under the assumptions of plate theory, the final shapes induced by the profiles $\gamma(z)$ and γ_{bi} coincide if the local forces and moments induced by the two profiles are equal, meaning that the conditions

$$\int_{-h/2}^{h/2} \gamma(z) dz = \frac{h}{2}(\varepsilon^t + \varepsilon^b), \tag{2}$$

$$\int_{-h/2}^{h/2} \gamma(z) z dz = \frac{h^2}{8}(\varepsilon^t - \varepsilon^b) \tag{3}$$

are met. Therefore, Eqs. (2), (3) allow to determine ε^t and ε^b element-wise and thus to reduce any through-thickness eigenstrain profile $\gamma(z)$ to the bilayer formulation.

2.2. The general approach to k-means clustering of the eigenstrain pattern

In terms of clustering, the variables $(\varepsilon^t, \varepsilon^b)$ act as coordinates of points on a plane, where each point corresponds to the peening state of a finite element in the peen forming pattern. Typical of a k-means method, our clustering algorithm uses a predefined number of centroids, or in other words, a predefined number of peening regimes. When the points are clustered, the algorithm assigns each finite element with the eigenstrains corresponding to the centroid of the cluster containing this element. This strategy allows, in particular, to constrain the maximal eigenstrain in the pattern by imposing limits on the coordinates of the centroids. The cluster centroids are related to the peening regimes, so the clustering algorithm outputs top- and bottom-side peen forming patterns formulated in terms of peening regimes.

The k-means clustering is an iterative process. On each iteration, it attributes the points to the nearest centroids in terms of squared Euclidean distance and then recomputes the centroid positions as the mean points of the corresponding clusters. The iterations stop when all centroids no longer change their positions. The result of clustering is dependent on the initial guess of the centroid positions, which are assigned randomly [37]. For this reason, the algorithm must be relaunched several times with different initial centroid positions. 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 best clustering. The number of trials is chosen the balance between the computational cost and the desired accuracy. For example, performing 100 trials of the k-means algorithm allows to decrease the clustering error by 60% with respect to a single trial on average [43]. However, the number of all possible partitions of a given set of points is finite, so, if the algorithm is guaranteed to try all partitions in a certain number of trials, then further increasing of this number does not bring any positive effect.

3. Methodology

3.1. The clustering algorithm

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. In other words, we describe the forming effect of a treatment using the bilayer model, where the treated layer expands by $\varepsilon^{(1)}$ and the other layer contracts by $\varepsilon^{(2)}$. 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 $\left(\varepsilon_i^{(1)} + \varepsilon_j^{(2)}, \varepsilon_i^{(2)} + \varepsilon_j^{(1)}\right)$. An unclustered pattern consists of such eigenstrain combinations assigned to each element, but the aim of the clustering algorithm is to find the best suitable regimes. Consequently, the clustering algorithm must split the combinations into contributions made by the top and bottom peening regimes independently.

The clustering algorithm starts by initializing 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, 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:

$$\begin{cases} 0 < \varepsilon^{(1)} < \varepsilon_{max}^{(1)}, \\ \varepsilon_{min}^{(2)} < \varepsilon^{(2)} < 0, \end{cases} \tag{4}$$

where $\varepsilon_{max}^{(1)}$ and $\varepsilon_{min}^{(2)}$ are, correspondingly, the maximal positive and the minimal negative eigenstrains achievable with the peening equipment. These values must be experimentally characterized in advance.

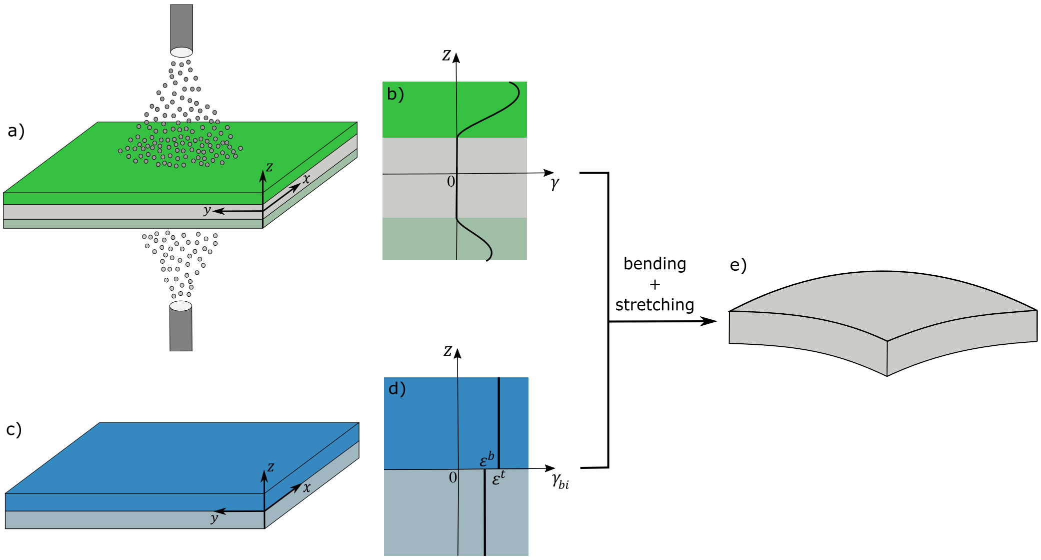


Fig. 2. Idealization of the through-thickness eigenstrain profile. In this example, we suppose that the plate is uniformly peened from both sides and that the top side peening regime is more intense. (a) The treatment induces eigenstrains in the outer layers of the plate (bright green and pale green), while the internal layer (gray) is not affected. (b) The induced eigenstrain prescribes an in-plane expansion to the outer layers, and the expansion magnitude γ is non-uniform along the through-thickness lagrangian coordinate z. (c) To idealize the eigenstrain profile, we represent the plate as a bilayer consisting of equally thick layers (bright blue and pale blue). (d) We prescribe a uniform eigenstrain to both layers and thus obtain an idealized through-thickness eigenstrain profile γ_{bi}, which is described with two variables: $(\varepsilon^t, \varepsilon^b)$. The values of ε^t and ε^b are computed in the way that γ_{bi} induces the same final shape as $\gamma(z)$ (see Eqs. (2), (3)), in the sense of plate theory. (e) The final shape induced by γ_{bi} and $\gamma(z)$ is stretched with respect to the initial state and convex towards the side where a more intense treatment was applied. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Table 1
Formation of the cluster centroids. 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

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 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. Fig. 3(a) shows the points and the centroids initialized on the plane.

The next stage is the deletion of all centroids that lie above the line $\varepsilon^t = \varepsilon^b$. Indeed, the centroids ij and ji are induced by the same peening regimes applied conversely, so they are symmetrical about the line $\varepsilon^t = \varepsilon^b$. However, the algorithm aims at optimizing the peening regimes, and the range of regimes is the same for both sides of the plate. Consequently, the algorithm deletes one centroid from each pair of symmetrical centroids.

The symmetry principle also applies to the points corresponding to the elements. Thus, the eigenstrains $(\varepsilon^t_e, \varepsilon^b_e)$ and $(\varepsilon^b_e, \varepsilon^t_e)$ 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. Accordingly, the clustering algorithm reflects all points that lie above the line $\varepsilon^t = \varepsilon^b$ across this line instead of deletion. With this, the points lying on either side of the line $\varepsilon^t = \varepsilon^b$ are fused and clustered simultaneously. Fig. 3(b) traces the reflected points and the preserved centroids.

The iterative process begins next. Similarly as during the standard $\mathtt{k}$-means clustering, each point, i.e., element, is assigned to the cluster represented by the closest centroid, which is shown in Fig. 3(c). However, the centroids cannot be relocated freely for every iteration because they are coupled. Indeed, the relocation of centroid ij means altering the eigenstrains induced by both regimes i and j. This, in turn, induces a simultaneous relocation of all centroids formed by regimes i and j in combination with another regime. For this reason, the 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 $(\varepsilon^t_e, \varepsilon^b_e)$ 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 $\left(\varepsilon^{(1)}_{e,i}, \varepsilon^{(2)}_{e,i}\right)$ and $\left(\varepsilon^{(1)}_{e,j}, \varepsilon^{(2)}_{e,j}\right)$, 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 [44]. These coordinates must minimize the sum of squared Euclidean distances f to the regimes i and j:

$$f = \left(\varepsilon^{(1)}_{e,i} - \varepsilon^{(1)}_{i}\right)^2 + \left(\varepsilon^{(2)}_{e,i} - \varepsilon^{(2)}_{i}\right)^2 + \left(\varepsilon^{(1)}_{e,j} - \varepsilon^{(1)}_{j}\right)^2 + \left(\varepsilon^{(2)}_{e,j} - \varepsilon^{(2)}_{j}\right)^2. \quad (5)$$

In addition, the combination of contributions $\left(\varepsilon^{(1)}_{e,i}, \varepsilon^{(2)}_{e,i}\right)$ and $\left(\varepsilon^{(1)}_{e,j}, \varepsilon^{(2)}_{e,j}\right)$ must equal $(\varepsilon^t_e, \varepsilon^b_e)$, which leads to:

$$\begin{cases} \varepsilon^{(1)}_{e,i} + \varepsilon^{(2)}_{e,j} = \varepsilon^t_e, \\ \varepsilon^{(2)}_{e,i} + \varepsilon^{(1)}_{e,j} = \varepsilon^b_e. \end{cases} \quad (6)$$

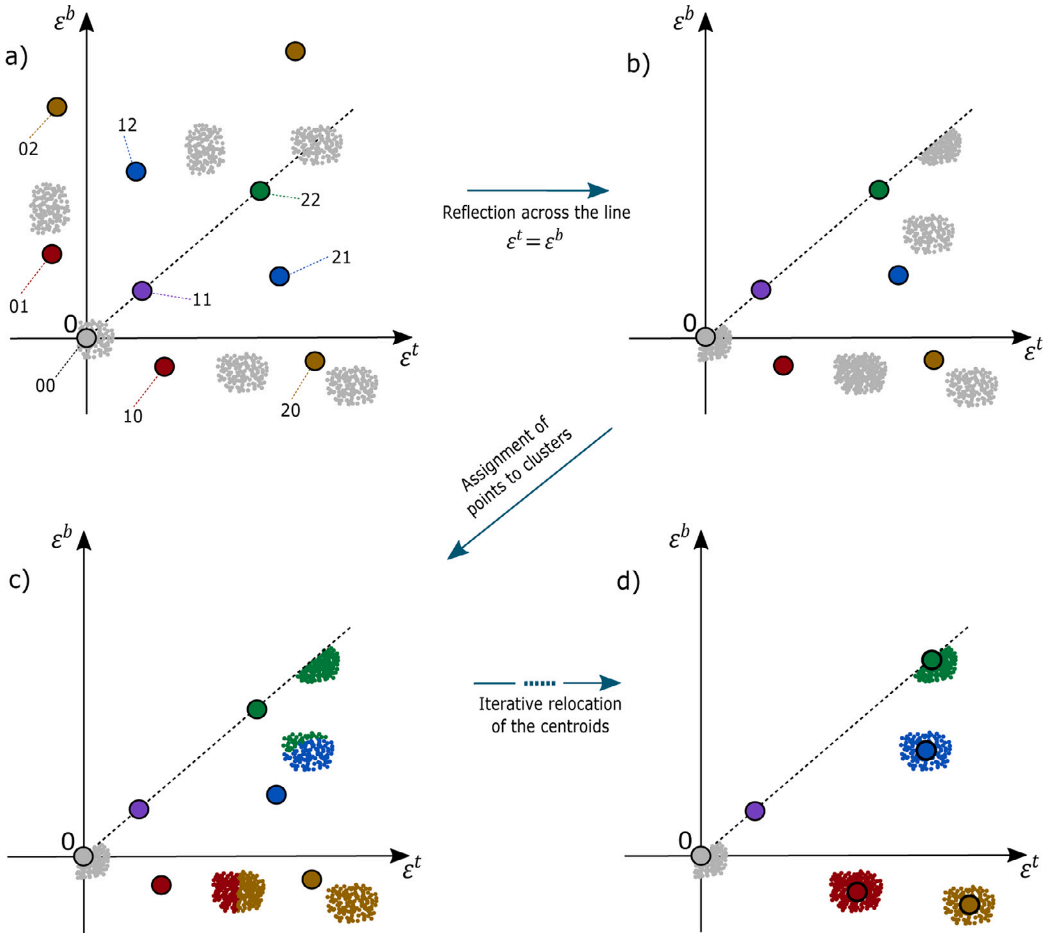

a)
b)
c)
d)
ε^b
ε^t
0
02
12
22
01
11
21
00
10
20
Reflection across the line
$\varepsilon^t = \varepsilon^b$
Assignment of points to clusters
Iterative relocation of the centroids

Fig. 3. Example of clustering with $N = 2$ available peening regimes. (a) At the beginning, the clustering algorithm traces the points (gray 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. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

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

$$\mathcal{L} = \left(\varepsilon^{(1)}_{e,i} - \varepsilon^{(1)}_{i}\right)^2 + \left(\varepsilon^{(2)}_{e,i} - \varepsilon^{(2)}_{i}\right)^2 + \left(\varepsilon^{(1)}_{e,j} - \varepsilon^{(1)}_{j}\right)^2 + \left(\varepsilon^{(2)}_{e,j} - \varepsilon^{(2)}_{j}\right)^2 - \lambda_1\left(\varepsilon^{(1)}_{e,i} + \varepsilon^{(2)}_{e,j} - \varepsilon^{t}_{e}\right) - \lambda_2\left(\varepsilon^{(2)}_{e,i} + \varepsilon^{(1)}_{e,j} - \varepsilon^{b}_{e}\right). \tag{7}$$

The stationary point of the Lagrangian function is:

$$\begin{cases} \varepsilon^{(1)}_{e,i} = 0.5\left(\varepsilon^{(1)}_{i} - \varepsilon^{(2)}_{j} + \varepsilon^{t}_{e}\right), \\ \varepsilon^{(2)}_{e,i} = 0.5\left(\varepsilon^{(2)}_{i} - \varepsilon^{(1)}_{j} + \varepsilon^{b}_{e}\right), \\ \varepsilon^{(1)}_{e,j} = 0.5\left(\varepsilon^{(1)}_{j} - \varepsilon^{(2)}_{i} + \varepsilon^{b}_{e}\right), \\ \varepsilon^{(2)}_{e,j} = 0.5\left(\varepsilon^{(2)}_{j} - \varepsilon^{(1)}_{i} + \varepsilon^{t}_{e}\right), \end{cases} \tag{8}$$

$$\begin{cases} \lambda_1 = \varepsilon^{t}_{e} - \varepsilon^{(1)}_{i} - \varepsilon^{(2)}_{j}, \\ \lambda_2 = \varepsilon^{b}_{e} - \varepsilon^{(1)}_{j} - \varepsilon^{(2)}_{i}. \end{cases} \tag{9}$$

The two contributions computed according to Eqs. (8) minimize the sum of squared Euclidean distances (5) and satisfy conditions (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 $\left(\varepsilon^{(1)}_{e,i}, \varepsilon^{(2)}_{e,i}\right)$ 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 is summarized as Algorithm 1. Fig. 3(d) illustrates the result of clustering.

3.2. *Filtering of the peen forming pattern using cellular automata*

The clustering algorithm groups the finite elements based on the eigenstrain prescribed during the inverse problem resolution. After clustering, each finite element in the peen forming 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 of only one element is not typically reproducible in practice. The presence of such small-scale segments in the clustered pattern is called the *checkerboard* problem, which is illustrated in Fig. 4. This problem arises when the eigenstrain prescribed by the inverse problem solver is on the border between two

Algorithm 1 The clustering algorithm

```
for each finite element $e$ do
    Initialize one point with coordinates $(\varepsilon^t_e, \varepsilon^b_e)$
end for
Initialize the void peening regime $(0, 0)$
Randomly initialize $N$ peening regimes in terms of $\left(\varepsilon^{(1)}, \varepsilon^{(2)}\right)$ under constraints (4)
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 $\varepsilon^t$ and $\varepsilon^b$
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 Eqs. (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 $(\varepsilon^t_e, \varepsilon^b_e)$
end for
```

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 by Selvapeter and Hordijk [45]. 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 [46]. 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 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 peen forming pattern [8], because the grouped patterns can also face the checkerboard problem.

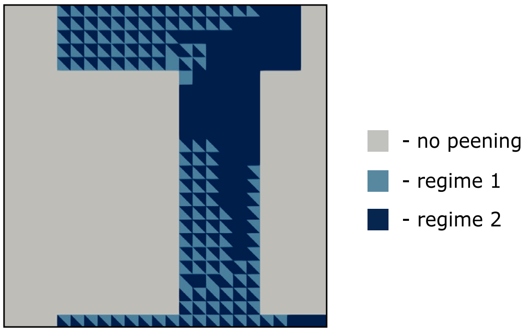


Fig. 4. Example of a peen forming pattern heavily affected by the checkerboard problem after clustering. The pattern is represented on a square plate, which is meshed with triangular finite elements. The elements assigned with different peening regimes of a close intensity are locally mixed, so the proposed pattern is not reproducible in practice.

The filtering algorithm is summarized as Algorithm 2. We formulate the algorithm in terms of the peening regime ρ_e assigned to element e. The regimes assigned to the neighboring elements are denoted as ρ_{e1}, ρ_{e2}, ρ_{e3}.

Algorithm 2 The filtering algorithm

```
while the eigenstrain is reassigned at least for one element do
    for each finite element $e$ do
        if the element has three neighbors $e_1$, $e_2$, $e_3$, then
            if $\rho_e \neq \rho_{e1}$ and ($\rho_{e1} = \rho_{e2}$ or $\rho_{e1} = \rho_{e3}$) then
                Assign the value of $\rho_{e1}$ to $\rho_e$
            else if $\rho_e \neq \rho_{e2}$ and $\rho_{e2} = \rho_{e3}$ then
                Assign the value of $\rho_{e2}$ to $\rho_e$
            else if
                ($\rho_e \neq \rho_{e1}$ and $\rho_e \neq \rho_{e2}$ and $\rho_e \neq \rho_{e3}$) and
                ($\rho_{e1} \neq \rho_{e2}$ and $\rho_{e1} \neq \rho_{e3}$ and $\rho_{e2} \neq \rho_{e3}$)  then
                Assign the value of $\rho_{e1}$ to $\rho_e$
            end if
        end if
        if the element has two neighbors $e_1$, $e_2$, then
            if $\rho_e \neq \rho_{e1}$ and $\rho_e \neq \rho_{e2}$ then
                Assign the value of $\rho_{e1}$ to $\rho_e$
            end if
        end if
        if the element has one neighbor $e_1$, then
            if $\rho_e \neq \rho_{e1}$ then
                Assign the value of $\rho_{e1}$ to $\rho_e$
            end if
        end if
    end for
end while
```

4. Results

The clustering and the filtering algorithms were implemented in C++ programming language with the help of “Eigen” [47] library for

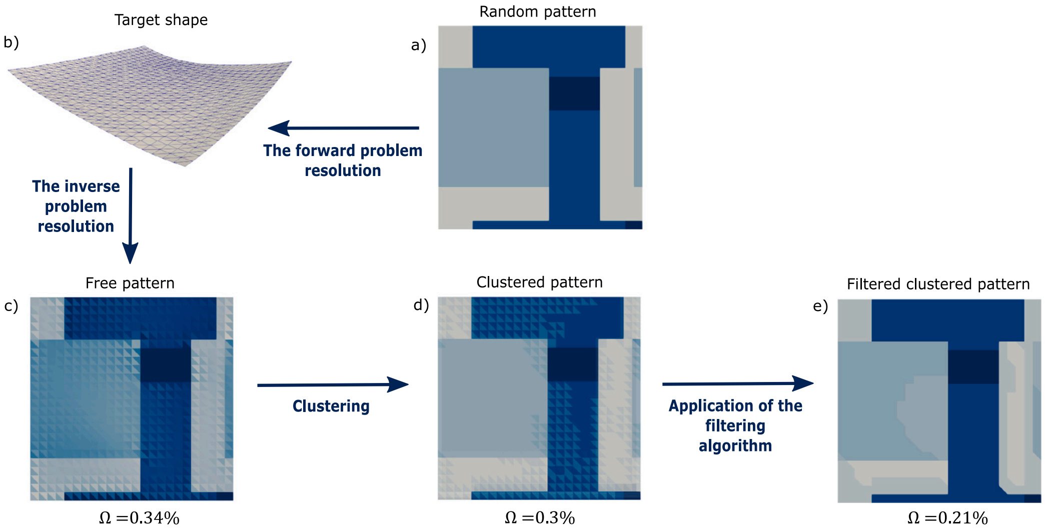

Target shape
b)
Random pattern
a)
The forward problem resolution
The inverse problem resolution
Free pattern
c)
Clustered pattern
d)
Filtered clustered pattern
e)
Clustering
Application of the filtering algorithm
Ω =0.34%
Ω =0.3%
Ω =0.21%

Fig. 5. One of the test cases from series 2 that demonstrated a marked checkerboard problem. (a) The random peen forming pattern generated on a 1 × 1 m sized plate using the algorithm presented by Sushitskii et al. [8]. Different colors on the peen forming 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 (see [8]). (d) The clustered peen forming pattern computed by application of Algorithm 1 to the free pattern. (e) The filtered clustered algorithm obtained by application of Algorithm 2 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. In the presented case, the filtered clustered pattern reproduces the original random pattern better than the free pattern, so it leads to a lower dimensionless error Ω. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

data operations and "libigl" [48] library for triangular mesh handling. The algorithms were validated numerically using 200 random target shapes. For each test case, we firstly solved the inverse problem using the algorithm presented by Sushitskii et al. [8] and obtained the *free* peen forming pattern. Secondly, we applied the clustering algorithm (Algorithm 1) and thus obtained the *clustered* peen forming pattern. Thirdly, we applied the filtering algorithm (Algorithm 2) and obtained the *filtered clustered* peen forming pattern. Finally, we simulated the application of each of these three patterns using the method presented by Sushitskii et al. [8] to evaluate discrepancy between this and that. 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 square root of the total area of the plate.

The target shapes were generated as a result of the application of random peen forming 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 peen forming patterns were generated on this mesh using the algorithm presented by Sushitskii et al. [8]. 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.

Fig. 5 illustrates the numerical validation workflow using one test case from series 2. The presented clustered pattern (Fig. 5d) leads to the highest Ω in series 2 because the pattern is significantly affected by the checkerboard problem. The filtered clustered pattern (Fig. 5e) contains irregularly shaped peening segments, but it is not subjected to the checkerboard problem, so it is applicable in practice. For the sake of completeness, two additionnal exemples are presented in Fig. B.8.

Fig. 6 summarizes the validation results in terms of Ω. The histograms demonstrate that the dimensionless error Ω did not exceed 0.4% for the final shapes provided by the free patterns. Both clustering and filtering algorithms made a slightly negative impact in terms of Ω, but Ω stayed in the same range of 0.4% for all the test cases. Moreover, Ω remained 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 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. The number of trials for the clustering algorithm was fixed as 100. The influence of clustering in terms of Ω is plotted in Fig. 7(a). Thus, the clustered pattern decreased Ω with respect to the free pattern for 78 cases in series 1 and for 32 cases in series 2. The decrease in Ω in these cases is explained by the segmented structure of the random patterns that were used to generate the target shapes. Hence, the clustering removed the gradual eigenstrain variation from the free patterns and made the clustered patterns more similar to the original ones.

On the other hand, the difference in the influence of clustering on the two series 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 the 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.

The filtered clustered pattern increased Ω with respect to the clustered pattern for 77 cases in series 1 and for 67 cases in series 2, which is shown in Fig. 7(b). The reason for this increase in most of the cases is that the filtering algorithm often enlarges the segments assigned to wrong clusters instead of eliminating them.

All in all, the successive application of the clustering and the filtering algorithms decreased Ω with respect to the free pattern for 67 cases out of 200 (see Fig. 7c). The increase in Ω for 55 cases in series 1 and for 78 cases in series 2 is explained by the fact that the segmentation strategy does not account for Ω, so its output depends only on the input pattern.

For the presented series of tests, the computational burden of the segmentation process (~0.1 s) was negligible with respect to the inverse problem resolution (~1 min), which is an iterative process involving numerical optimization on each iteration [8].

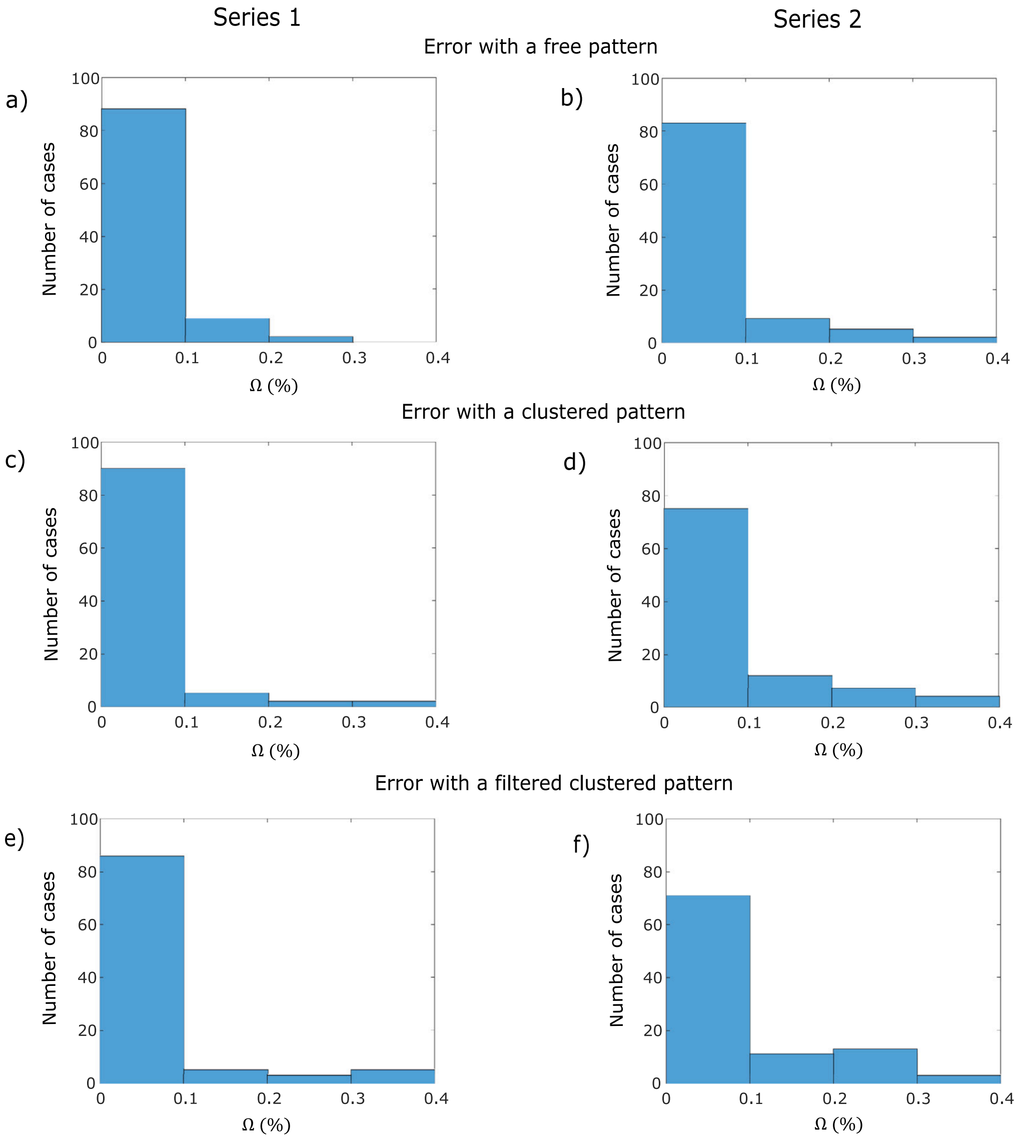


Fig. 6. Histograms plotting the dimensionless error Ω between the target shapes and the final shapes computed during the validation of the clustering and filtering algorithms. We considered 200 test cases that were divided in two series of 100 cases each. In each case, we solved the inverse problem, clustered the peen forming 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 the application of the peen forming patterns at each stage and computed the dimensionless error Ω. Figures (a) and (b) plot 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. The filtered clustered patterns induce a slightly larger Ω than the free patterns in general, but Ω does not exceed 0.4% for all the types of patterns.

5. Discussion

In our experiments, the quantity of peening regimes N was fixed for each series of tests. However, in practice N is not always predetermined. In a such case, the clustering algorithm 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 in practice. Here, more practical considerations will need to be taken into account by engineers using the method such as what kind of treatments are available and how much time can be allotted to the treatment.

Both the clustering and the filtering algorithms can either increase or decrease the Ω, which introduces uncertainty in the solution. One potential way to mitigate this problem is to implement a black-box optimization algorithm, which would optimize the eigenstrains prescribed to each peening regime after the filtering in order to reduce Ω. However, this approach would significantly increase the computational burden. In the current implementation, the time complexity of the proposed segmentation strategy scales linearly with the number of finite elements in the model, similarly to the original `k`-means algorithm.

In practice, the actual peening parameters can be related to the eigenstrains constituting the segmented pattern using experiments in combination with the eigenstrain simulations. Thus, for each couple $\left(\varepsilon^{(1)}, \varepsilon^{(2)}\right)$, one can numerically simulate a uniform treatment of a dummy specimen, such as an Almen strip, and calculate the resulting shape of the specimen. Next, the peening parameters can be adjusted in order to achieve the computed shape in practice. Furthermore, one can establish a continuous relationship between one of the peening

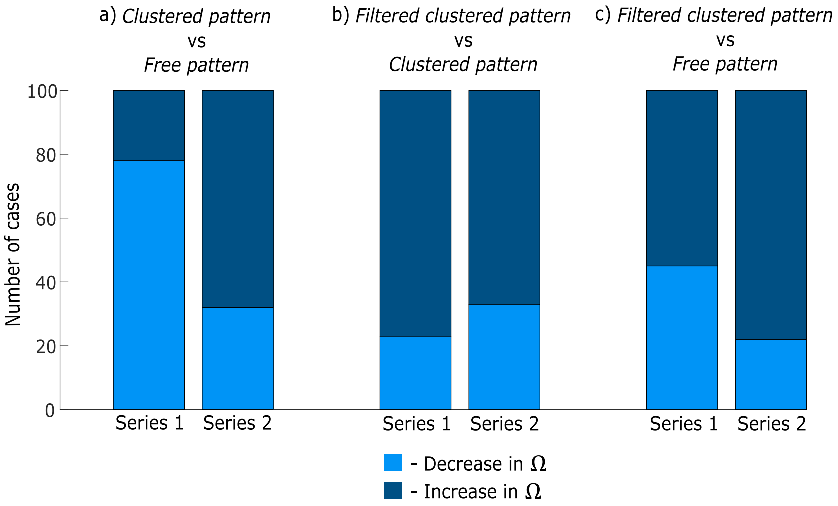


Fig. 7. Influence of the clustering and the filtering algorithms on the dimensionless error Ω. The clustered patterns were obtained by application of the clustering algorithm to the free patterns, and the filtered clustered patterns were obtained by application of the filtering algorithm to the clustered patterns. Hence, the bars (a) represent the influence of the clustering algorithm, the bars (b) represent the influence of the filtering algorithm, and the bars (c) represent the influence of the successive application of the two algorithms. The first series of tests implied one peening regime during generation of the training examples and clustering, while the second series implied four peening regimes. Both series consisted of 100 test cases. The results show that the influence of the successive application of the two algorithms is majorly negative.

parameters, such as the peening pressure, and the induced eigenstrains. However, a precise workflow for establishing this relationship is beyond the scope of this paper.

6. Conclusion

We developed a general strategy for the segmentation of the peen forming pattern, which makes every pattern applicable with shot peening equipment. This step is essential. Without this clustering algorithm, the continuous pattern that comes out of any peen forming optimization method cannot be applied directly. Our method allows discretizing the pattern into a finite number of achievable treatment segments. The segmentation strategy is a crucial step towards a fully automated forming process, which is more cost-efficient and repeatable than the manual operation. The segmentation strategy computes the best suitable peening regimes, and the segments where they should be applied. It includes a clustering algorithm, which is inspired by the k-means method, and a filtering algorithm. The clustering algorithm is designed for the bilayer eigenstrain formulation, and any other formulation of the through-thickness eigenstrain profile can be reduced to the bilayer. 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 peen forming pattern, which is suitable for industrial reproduction using masking or a narrow peening stream.

The clustering and the filtering algorithms 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 peen forming pattern, so that the original and the clustered patterns induced similar final shapes. Out of 200 tested cases, the error due to the segmentation, quantified with the normalized Hausdorff distance, did not attain more than 0.4% for any tested pattern. A clustered pattern involving one peening regime is less affected by the checkerboard problem than this involving multiple regimes, 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. Nevertheless, it efficiently deals with the checkerboard problem and rapidly eliminates local clustering errors.

One limitation of the filtering strategy is the dependence of the minimal size of the treated segment on the size of elements. Hence, the smallest treated segment can still be narrower than the shot stream in the case of a dense mesh, which can be mitigated by narrowing the peening nozzle. In addition, the proposed segmentation strategy is limited to isotropic eigenstrain models, because the ability to vary the principal growth direction from one element to another is not yet included in the algorithms. Finally, the proposed segmentation strategy is axed on delivering a segmented pattern as close as possible to the free peen forming pattern in order to make the forming treatment practically applicable. This means that the range of available target shapes and the forming quality cannot be enhanced with the segmentation process, but rather depend on the peening equipment and the inverse problem solver.

The described segmentation strategy can be applied as a post-processing tool for any eigenstrain-based inverse problem solvers that deal with forming of metal plates or with laminated shape-shifting structures. Such laminated structures are present, for example, in the fields of Micro-Electro-Mechanical Systems (MEMS) or 4D printed composites. In addition, the modified k-means method with the interconnected centroids presents an advancement in the filed of clustering algorithms. The future work implies an experimental validation, which will evaluate the efficiency of the automated computation of the optimal peening regimes with respect to a fixed set of peening regimes determined through manual calibration.

CRediT authorship contribution statement

Vladislav Sushitskii: Conceptualization, Data curation, Formal analysis, Investigation, Methodology, Software, Validation, Visualization, Writing – original draft. **Hong Yan Miao:** Conceptualization, Resources, Software, Visualization, Writing – review & editing. **Martin Lévesque:** Conceptualization, Formal analysis, Funding acquisition, Investigation, Supervision, Writing – original draft, Writing – review & editing. **Frédérick P. Gosselin:** Conceptualization, Formal analysis, Funding acquisition, Project administration, Resources, Supervision, Writing – original draft, Writing – review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

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.

Funding information

Funding was provided by:

- The Natural Sciences and Engineering Research Council of Canada (NSERC) (funding reference numbers RGPIN-2019-07072 and RGPIN-06412-2016);
- The Fonds de Recherche du Québec – Nature et Technologies (FRQNT) and The Ministère de l'Économie et de l'Innovation du Québec (funding reference number LU-210888);
- The Aluminium Research Centre – REGAL;
- Aerosphere Inc.

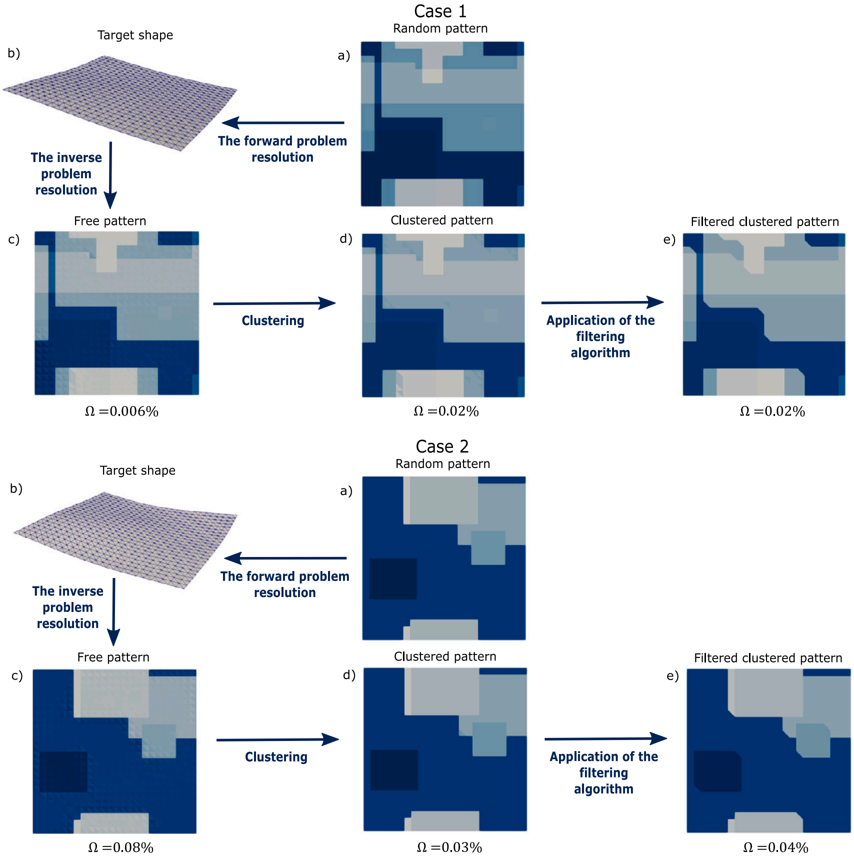
Appendix A. k-means clustering of the trialyer eigenstrain pattern

Another common way to idealize the through-thickness eigenstrain profile is to represent it as a trilayer $\gamma_{tri}(z)$. The trilayer formulation implies the assignment of eigenstrain to the two outer layers of a variable thickness:

$$\gamma_{tri}(z) = \begin{cases} \varepsilon_*^t \quad \text{for} \quad \left(\frac{h}{2} - h_*^t\right) < z < \frac{h}{2}, \\ 0 \quad \text{for} \quad \left(h_*^b - \frac{h}{2}\right) < z < \left(\frac{h}{2} - h_*^t\right), \\ \varepsilon_*^b \quad \text{for} \quad -\frac{h}{2} < z < \left(h_*^b - \frac{h}{2}\right), \end{cases} \tag{A.1}$$

where ε_*^t and ε_*^b stand for the eigenstrains assigned to the top and bottom layers respectively, while h_*^t and h_*^b denote the thicknesses of these layers. Hence, in this formulation, the eigenstrain assigned to each finite element e is described with four variables $(\varepsilon_{*e}^t, h_{*e}^t, \varepsilon_{*e}^b, h_{*e}^b)$.

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 $(\varepsilon_{*e}^t, h_{*e}^t)$ and $(\varepsilon_{*e}^b, h_{*e}^b)$. In this case, the points can be clustered using the standard k-means algorithm [39], 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 using the *Maxmin* method, which proves itself as the most efficient initialization technique for the k-means algorithm [43]. With this method, the centroids are initialized in sequence, and each subsequent centroid is placed at the location of the furthest point from the previous centroid. The first centroid in this sequence is the zero centroid.



Case 1
Random pattern
Target shape
b)
a)
The forward problem resolution
The inverse problem resolution
Free pattern
c)
Clustered pattern
d)
Filtered clustered pattern
e)
Clustering
Application of the filtering algorithm
Ω =0.006%
Ω =0.02%
Ω =0.02%
Case 2
Random pattern
Target shape
b)
a)
The forward problem resolution
The inverse problem resolution
Free pattern
c)
Clustered pattern
d)
Filtered clustered pattern
e)
Clustering
Application of the filtering algorithm
Ω =0.08%
Ω =0.03%
Ω =0.04%

Fig. B.8. Two test cases from series 2. (a) The random peen forming pattern generated on a 1 × 1 m sized plate using the algorithm presented by Sushitskii et al. [8]. Different colors on the peen forming pattern denote different peening regimes. (b) The target shape generated by application of the random pattern. The deflections are at the original scale. (c) The free pattern computed using the inverse problem resolution algorithm (see [8]). (d) The clustered peen forming pattern computed by application of Algorithm 1 to the free pattern. (e) The filtered clustered algorithm obtained by application of Algorithm 2 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. In case 1, the free pattern reproduces the original random pattern the best. This is due to several elements attributed to wrong clusters during clustering, so the dimensionless error Ω is higher for the clustered pattern. In case 2, the lowest Ω is provided by the clustered pattern, which almost exactly replicates the random pattern. Although the filtering algorithm makes the pattern fully reproducible, in case 2 the filtered clustered pattern is further from the random pattern than the clustered one. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

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 the zero centroid at the origin
Initialize $N$ centroids at random positions in coordinates $(\varepsilon_*, h_*)$
using the Maxmin method
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 $(\varepsilon^t_*, h^t_*)$ corresponding
    to the centroid of the cluster containing point $(\varepsilon^t_{*e}, h^t_{*e})$
    Assign the bottom layer with a couple $(\varepsilon^b_*, h^b_*)$ corre-
    sponding to the centroid of the cluster containing point
    $(\varepsilon^b_{*e}, h^b_{*e})$.
end for
```

Appendix B. Additional test cases

Additionnal test cases are presented in Fig. B.8.

References

[1] Ramati S, Kennerknecht S, Levasseur G. Single piece wing skin utilization via advanced peen forming technologies. In: Proceedings of the 7th international conference on shot peening. ICSP7, Warsaw; 1999.

[2] Russig C, Bambach M, Hirt G, Holtmann N. Shot peen forming of fiber metal laminates on the example of glare (r). Int J Mater Form 2014;7.

[3] Merino J, Patzelt A, Steinacher A, Windisch M, Heinrich G, Forster R, Bauer C. Ariane 6-tanks and structures for the new European launcher. Deutsche Gesellschaft für Luft-und Raumfahrt-Lilienthal-Oberth eV; 2017.

[4] Baughman D. Peen forming. Mach Des 1970;42:156–60.

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

[6] Miao HY, Lévesque M, Gosselin FP. Shot peen forming pattern optimization to achieve cylindrical and saddle target shapes: The inverse problem. CIRP J Manuf Sci Technol 2022;36:67–77.

[7] Wüstefeld F, Linnemann W, Kittel S. Towards peen forming process automation. In: Shot peening. 2006, p. 44.

[8] Sushitskii V, van Rees WM, Levesque M, Gosselin FP. Determination of optimal shot peen forming patterns using the theory of non-Euclidean plates. J Manuf Sci Eng 2022;145.

[9] VanLuchene R, Johnson J, Carpenter R. Induced stress relationships for wing skin forming by shot peening. J Mater Eng Perform 1995;4:283–90.

[10] VanLuchene R, Cramer E. Numerical modeling of a wing skin peen forming process. J. Mater Eng Perform 1996;5:753–60.

[11] Kopp R, Schulz J. Optimising the double-sided simultaneous shot peen forming. In: Proceedings of the 8th international conference on shot peening. ICSP8, Garmisch-Partenkirchen; 2002, p. 227–33.

[12] Wang T, Platts M, Wu J. The optimisation of shot peen forming processes. J Mater Process Technol 2008;206:78–82.

[13] Essa Y, Laspalas M, Garcia E, Escolán A, Hernández-Gascón B, Martin de la Escalera F. Numerical analysis to optimize peen-forming process parameters. In: Simulia community conference proceedings. 2015, p. 12–21.

[14] Gariépy A, Cyr J, Levers A, Perron C, Bocher P, Lévesque M. Potential applications of peen forming finite element modelling. Adv Eng Softw 2012;52:60–71.

[15] Wang C, Zhao Z, Jiang J, Wang Y, Yang W, Yang J. Experimental and simulation study of multi-region peen forming and mechanical property. J Manuf Process 2022;84:198–215.

[16] Xiao X, Tong X, Sun Y, Li Y, Wei S, Gao G. An analytical model for predicting peening stresses with general peening coverage. J Manuf Process 2019;45:242–54.

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

[18] Korsunsky AM. The modelling of residual stresses due to surface peening using eigenstrain distributions. J Strain Anal Eng Des 2005;40:817–24.

[19] Zhang S, Venter A, Vorster W, Korsunsky A. High-energy synchrotron X-ray analysis of residual plastic strains induced in shot-peened steel plates. J Strain Anal Eng Des 2008;43:229–41.

[20] Coratella S, Sticchi M, Toparli M, Fitzpatrick M, Kashaev N. Application of the eigenstrain approach to predict the residual stress distribution in laser shock peened AA7050-t7451 samples. Surf Coat Technol 2015;273:39–49.

[21] Chen Z, Yang F, Meguid S. Realistic finite element simulations of arc-height development in shot-peened almen strips. J Eng Mater Technol 2014;136.

[22] Chaise T, Li J, Nélias D, Kubler R, Taheri S, Douchet G, Robin V, Gilles P. Modelling of multiple impacts for the prediction of distortions and residual stresses induced by ultrasonic shot peening (USP). J Mater Process Technol 2012;212:2080–90.

[23] Siguerdidjane W, Khameneifar F, Gosselin FP. Efficient planning of peen-forming patterns via artificial neural networks. Manuf Lett 2020;25:70–4.

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

[25] Ohta T, Sato Y. Numerical analysis of peen forming for high-strength aluminum alloy plates. Mater Trans 2021;P–M2021818, advpub.

[26] Korsunsky AM. Residual elastic strain due to laser shock peening: modelling by eigenstrain distribution. J Strain Anal Eng Des 2006;41:195–204.

[27] Luo M, Hu Y, Hu L, Yao Z. Efficient process planning of laser peen forming for complex shaping with distributed eigen-moment. J Mater Process Technol 2020;279:116588.

[28] Fann K. Finite element study on forming metal sheets with an english wheel. 2022;1222:012006.

[29] Wang T, Platts M, Levers A. A process model for shot peen forming. J Mater Process Technol 2006;172:159–62.

[30] Pajot JM, Maute K, Zhang Y, Dunn ML. Design of patterned multilayer films with eigenstrains by topology optimization. Int J Solids Struct 2006;43:1832–53.

[31] van Rees WM, Vouga E, Mahadevan L. Growth patterns for shape-shifting elastic bilayers. Proc Natl Acad Sci 2017;114:11597–602.

[32] van Rees WM, Matsumoto EA, Gladman AS, Lewis JA, Mahadevan L. Mechanics of biomimetic 4D printed structures. Soft Matter 2018;14:8771–9.

[33] Sushitskii V, Dubois P-O, Miao HY, Levesque M, Gosselin F. Simulation and automation of aluminum panel shot peen forming. International Journal of Material Forming 2024;17:25.

[34] Faucheux PA. Simulating shot peen forming with eigenstrains (Phd dissertation), Polytechnique Montreal; 2019.

[35] Kirk D. Shot peening. Aircraft engineering and aerospace technology; 1999.

[36] Hornauer K, Kohler W. Development of the peen forming process for spherical shaped components. 1990;500:585–594.

[37] Rokach L, Maimon O. Clustering methods. In: Data mining and knowledge discovery handbook. Springer; 2005, p. 321–52.

[38] Xu D, Tian Y. A comprehensive survey of clustering algorithms. Ann Data Sci 2015;2:165–93.

[39] Lloyd S. Least squares quantization in PCM. IEEE Trans Inf Theory 1982;28:129–37.

[40] Badreddine J, Remy S, Micoulaut M, Rouhaud E, Desfontaine V, Renaud P. Cad based model of ultrasonic shot peening for complex industrial parts. Adv Eng Softw 2014;76:31–42.

[41] Rouquette S, Rouhaud E, François M, Roos A, Chaboche J-L. Coupled thermo-mechanical simulations of shot impacts: Effects of the temperature on the residual stress field due to shot-peening. J Mater Process Technol 2009;209:3879–86.

[42] Benallal A, Le Gallo P, Marquis D. An experimental investigation of cyclic hardening of 316 stainless steel and of 2024 aluminium alloy under multiaxial loadings. Nucl Eng Des 1989;114:345–53.

[43] Fränti P, Sieranoja S. How much can k-means be improved by using better initialization and repeats? Pattern Recognit 2019;93:95–112.

[44] Hoffmann LD, Bradley GL, Rosen KH. Calculus for business, economics, and the social and life sciences. McGraw-Hill; 2004.

[45] Selvapeter PJ, Hordijk W. Cellular automata for image noise filtering. 2009, p. 193–7.

[46] Zawidzki M. Application of semitotalistic 2D cellular automata on a triangulated 3D surface. Int J Des Nature Ecodynamics 2011;6:34–51.

[47] Guennebaud G, Jacob B, et al. Eigen V3. 2018.

[48] Jacobson A, Panozzo D, et al. Libigl: A simple C++ geometry processing library. 2018.