Available online at www.sciencedirect.com

ScienceDirect

Comput. Methods Appl. Mech. Engrg. 285 (2015) 621–638

Computer methods in applied mechanics and engineering

www.elsevier.com/locate/cma

A manifold learning-based reduced order model for springback shape characterization and optimization in sheet metal forming

Guenhael Le Quilliec[a], Balaji Raghavan[b,*], Piotr Breitkopf[c]

[a] *Laboratoire de Mecanique et Rheologie EA 2640, Universite Francois Rabelais de Tours, 7 avenue Marcel Dassault, 37200 Tours, France*
[b] *Laboratoire de Genie Civil et Genie Mecanique EA 3913, INSA de Rennes, 20 Avenue des Buttes de Coesmes, 35708 Rennes cedex, France*
[c] *Laboratoire Roberval UMR 7337 CNRS - Universite de Technologie de Compiegne, Centre de Recherches de Royallieu BP 20529 - 60205 Compiegne cedex, France*

Received 20 June 2014; received in revised form 19 October 2014; accepted 18 November 2014
Available online 9 December 2014

Abstract

The parameters of a stamping process include the geometry of the tools, the shape of the initial sheet blank, the material constitutive law and the process parameters. When designing the overall process, one has to also take into account the springback effect that appears when the tools are removed and additional surfaces are cut-off. The goal then is to obtain a final shape *as close as possible* to the desired shape, while satisfying the admissibility constraints on the variable parameters as well as the feasibility constraints frequently expressed in the form of forming limit diagrams. In the present paper we represent the post-springback shape by a level set function. Then, rather than rely on arbitrarily selected case-dependent measurement locations as in the NUMISHEET benchmark problems, we build a reduced order "shape space" where this level set evolves, by extending our recent shape manifold approach to the problem of springback assessment for 3D shapes. Next, we propose an optimization algorithm designed to minimize the gap between the post-springback and the desired final shapes. The required level set functions are generated from a corresponding set of springback shapes predicted by Finite Element simulations. Using our approach, we determine the *minimal* number of parameters needed in order to uniquely characterize the final formed shape regardless of complexity. Finally, we demonstrate the approach using an industrial test-case: springback assessment of the deep drawing operation of an automotive strut tower.

Keywords: Model reduction; Springback; Shape optimization; Characterization; Level set; Manifold learning; Proper orthogonal decomposition; Eulerian

1. Introduction

From the view of manufacturing of structural parts, high strength steels and aluminum are very attractive materials due to their good formability, high strength characteristics, price, or quality [1]. They are commonly used for complex

[*] Corresponding author. Tel.: +33 0 620 975 226.
E-mail address: balaji.raghavan@insa-rennes.fr (B. Raghavan).

http://dx.doi.org/10.1016/j.cma.2014.11.029

sheet metal parts. One of the most important problems however with these and similar materials (*i.e.* high ratio σ_y/E) is that of springback, which is severe during the unloading phase of a sheet metal forming operation and greatly affects the dimensional accuracy of the parts. High strain steels are vulnerable due to the high yield stress while aluminum alloys due to their low Young's modulus [2]. Springback is related to forming conditions, tool and blank geometry, other material properties such as yield stress, work hardening, strain rate sensitivity, Young's modulus *etc.* [3,4].

Corrections for springback are essential during die design in order to obtain specified final shapes. When dealing with the springback effect in an optimization context, we face a high dimensional problem. Post-springback shapes are typically represented by *deformed* FE meshes, although meshless representations with a set of nodes may be used as well [5]. The initial shapes are defined by up to a few hundred CAD parameters (that are not necessarily independent), but this is not the case with deformed meshes. The dimensionality in this case depends on the number of elements and/or nodes in the mesh and may thus be prohibitively high (e.g. the parameterization in [6]) and not directly concordant when remeshing is used. An obvious, but inefficient, way is to define *a posteriori* a set of geometric parameters to describe the complex 3D post-springback shape.

For example, even in the 2D draw bending of a simple U-channel, three parameters are used to measure the amount of springback [7] as was proposed in the benchmark of the NUMISHEET 93 conference. First of all, these are not easy to measure (*e.g.* optical scanning [8]), and moreover they are essentially decided on an ad-hoc basis, and either redundant or insufficient to *fully quantify* the final shapes obtained [9,10]. Furthermore, it is significantly more difficult to apply this simplistic approach to complex 3D test cases.

Secondly, when performing a set of numerical experiments, one obtains a family of post-springback shapes corresponding to different values of design parameters. The inverse problem [11] then consists of finding the values of parameters that yield a final shape as close as possible to the desired one. This requires predicting a new shape from a set of already computed ones by defining a proper space in which we are able to measure the distances and to interpolate between shapes.

Working directly with finite element meshes is not realistic due to potentially high numbers nodes/elements involved, as has already been mentioned.

We may define a set of CAD-like parameters (NURBS, *etc.*) spanning the variety of deformed shapes (including the desired shape), but this approach is tedious, arbitrary, and most importantly, difficult to automate.

Thus, since the springback shape is not easy to characterize, and given that the form obtained after springback frequently differs from the *desired* final shape, it is difficult to predict a unique set of process parameters (tool/punch geometry, blank holding force, *etc.*) in order to obtain a final shape as close to the manufacturer-desired shape as possible.

In order to numerically evaluate the springback and to be able to characterize complex shapes, we need a universal and case-*independent* technique to find the smallest number of parameters needed to fully describe the final shape obtained regardless of complexity, and easily compare it with the desired geometry. The first effort was made by the authors in [12] using their previously introduced "shape manifold" concept [13,14] for the simple NUMISHEET 93 benchmark problem of 2-D draw bending. Here, the concept of an "admissible shape" for a forming process was introduced for the first time to distinguish between *realizable/attainable* post-springback shapes and idealized shapes for a given drawing process, and the notion of interpolation between admissible shapes was introduced. The concept of interpolating level set functions has also been studied by [15] and [16] both of whom used radial basis functions (RBFs) and in conjunction with the level set equation.

The goal now is to parameterize a general complex 3D post-springback shape (in *level set* form, *i.e.* a signed distance function φ from the shape's surface), and interpolate between level set functions in a way that implicitly satisfies all the admissibility constraints, *i.e.* by developing the "shape space" *locally*. Using this we determine the intrinsic dimensionality of the drawing problem and thus the minimum number of parameters that control the final shape obtained at the end of the drawing process, and to express the final shape as a function of the geometric parameters $G_1, G_2, \ldots$, material parameters $M_1, M_2, \ldots$, and process parameters $Pr_1, Pr_2, \ldots$ (*e.g.* blank holding force, speed, friction, *etc.*) *i.e.* $\varphi = \varphi(G_1, G_2, \ldots, M_1, M_2, \ldots, Pr_1, Pr_2, \ldots)$. By calculating the distance between two admissible shapes *i.e.* $dist(\varphi_1, \varphi_2)$ or the distance of an inadmissible shape from the surface of the manifold of admissible shapes (*i.e.* realistic post-springback shapes), we can characterize the amount of springback and express it in terms of the set of design variables. A vital component here is the meta-model used to "reduce" the level set functions representing the shapes. Meta-modeling has been widely used to approximate the physical fields associated with the design problem using a lower order meta-model *i.e.* output space [17–19], using the methods of Proper

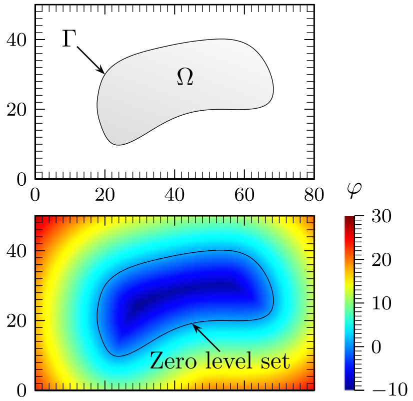

Fig. 1. Example of a level set function φ representing the boundary Γ of the 2D domain Ω.

Orthogonal Decomposition [20–23] or Proper Generalized Decomposition [24–27] or hyper-reduction [28]. In the field of metal forming, [29] presented an approach for displacement field approximation using the Proper Orthogonal Decomposition (POD) combined with kriging interpolation of projection coefficients. The authors have presented shape space meta-models for a variety of industrial problems [30,31,13] ending finally in the α-manifold [14]. Around the same time, [32,33] developed similar ideas of "slow manifolds" for the reduction of the output space of a problem in elastodynamics.

In this paper, we capitalize on our previous works and we propose an integrated approach designed to avoid above drawbacks and consisting of the following key concepts:

1. We use the level set as a mesh independent surface representation tool which allows for a variety of input data: CAD surface, finite mesh or even a pixel map from an actual physical experiment.
2. The POD-based manifold concept allows us to define a minimal unique set of design variables spanning the variety of final shapes.
3. The predictor–corrector manifold walk algorithm based on local Diffuse Approximation of the global manifold to progressively approach the desired shape.

This approach will allow for limiting the overall dimensionality to the *intrinsic dimensionality* of the post-springback shapes, directly linked with the varying parameters.

Since the work involves interpolation of level sets rather than a direct solution of the level set equation, this approach does not suffer from the computational effort one would expect with the upwinding methods and high order finite difference schemes normally used to solve the front propagation equation.

The methodology is presented in Section 2. We then apply this to an industrial drawing operation to characterize the geometries obtained post-springback and to identify the process parameters (force, friction coefficient) for a desired shape in Section 3. The paper ends with concluding comments and suggestions for future work.

2. Basic methodology and numerical tools

2.1. Level set representation of structural shapes

The level set functions allow to represent, to track and to analyze (hyper)surfaces. Most of the time they represent the boundaries Γ of a region Ω (Fig. 1). The evolution of Γ depends on the p variable parameters $\underline{P}$ of the underlying problem (*e.g.* time, position, dimensions, physics). The level set function φ of Γ simply corresponds to a signed distance function. In other words, the value of φ at an arbitrary point $\underline{x}$ of the space corresponds to the minimum distance between $\underline{x}$ and Γ. Thus, $\varphi = 0$ on Γ and, by convention, the sign of φ is positive outside Ω and negative inside. Therefore, the surface Γ for any given time or for any given set of values of the variable parameters $\underline{P}$ is represented as the zero level set by:

$$\varphi\left(\underline{P}\right) = 0. \tag{1}$$

The normal direction $\underline{n}$ on Γ can be obtained from φ:

$$\underline{n} = \frac{\underline{\text{grad}}\,(\varphi)}{\left\|\underline{\text{grad}}\,(\varphi)\right\|}. \tag{2}$$

The curvature κ of Γ is deduced from $\underline{n}$:

$$\kappa = \text{div}\left(\underline{n}\right). \tag{3}$$

The front propagation of φ over time can be captured by the Hamilton–Jacobi equation:

$$\frac{\partial\varphi}{\partial t} - v\left\|\underline{\text{grad}}\,(\varphi)\right\| = 0. \tag{4}$$

where v is the speed of propagation in the normal direction.

When Γ has a simple geometry, an analytical form of its corresponding level set function φ may be found. Regarding the more general level set method, φ is only defined on a fixed cloud of points, occupying a domain Π, often distributed on a cartesian grid surrounding Γ. Level set functions are then saved in a vector $\underline{\varphi}$. Eqs. (2) and (3) can then be solved by using finite differences.

Here however, we are not interested in solving the level set equation but rather generating intermediate level set functions – corresponding *exclusively* to admissible shapes for the given problem – from a set of existing level set functions corresponding to neighboring admissible shapes, while detecting the intrinsic local dimensionality of the given problem.

In other words, instead of trying to integrate (4) to get the evolution in time (or versus the variable parameters), we are interested here in performing *local interpolation* on a set of neighboring level set functions.

In the next section, we describe in detail the approach for interpolating between a set of input level set functions corresponding to calculated post-springback shapes.

2.2. *Using shape manifolds to efficiently interpolate between level set functions*

The domain of admissibility $\mathcal{F}$ is directly induced by the underlying problem with its constraints (which may be nonlinear as in [34]) and its degrees of freedom. In the case of a material forming problem, the level set function of a formed sheet after springback will be considered as admissible if it corresponds to any shape *realistically* achievable using a combination of the problem parameters (geometric parameters, material parameters and process parameters).

The approach assumes that $\mathcal{F}$ is a smooth shape space or manifold (Fig. 2), so that the set of all admissible level set functions are connected together. This fundamental hypothesis was previously made by Raghavan et al. [13,14]. The *desired* final shape (by the manufacturer, *etc.*) may in fact *not* correspond to a realizable shape in the given design problem, and in that case, will be considered as a *non-admissible* shape.

Following [14], we first generate a sample of s admissible shapes Γ_i in the vicinity of an initial design $\underline{P}_0$, as described in Fig. 3. In the second step, the corresponding level set (LS) functions $\underline{\varphi}_i \in \mathcal{F}$ of these shapes are computed, leading to high dimensionality vector representations of Γ_i. POD is performed in the third step to reduce the dimensionality as described in the Section 2.3. The relationship between the POD coefficients $\underline{\alpha}_i$ are next analyzed to deduce the local intrinsic dimensionality as described in Section 2.4. Their interpolation (described in the Section 2.5) produces an α-manifold approaching the true admissible shape space. By its very definition, the shape manifold has two important properties [13,14]:

1. All points in/on the manifold correspond to admissible shapes (*i.e.* level set functions)
2. Any point that lies outside the manifold corresponds to a **non** admissible shape for the given process.

We must comment on admissibility when shapes are represented by level sets. When we work with level sets instead of binary indicator functions, even an inadmissible point $\underline{\alpha}_{\text{IN}}$ outside the manifold **will** yield a signed distance function $\varphi(\underline{\alpha}_{\text{IN}})$ and hence a zero contour. This means that an inadmissible shape will not necessarily be visually obvious. The important point though is that the zero contour of $\varphi(\underline{\alpha}_{\text{IN}})$ will be physically infeasible (at least in the considered range of variation of the parameters) even if we are able to recover a zero contour that for all practical

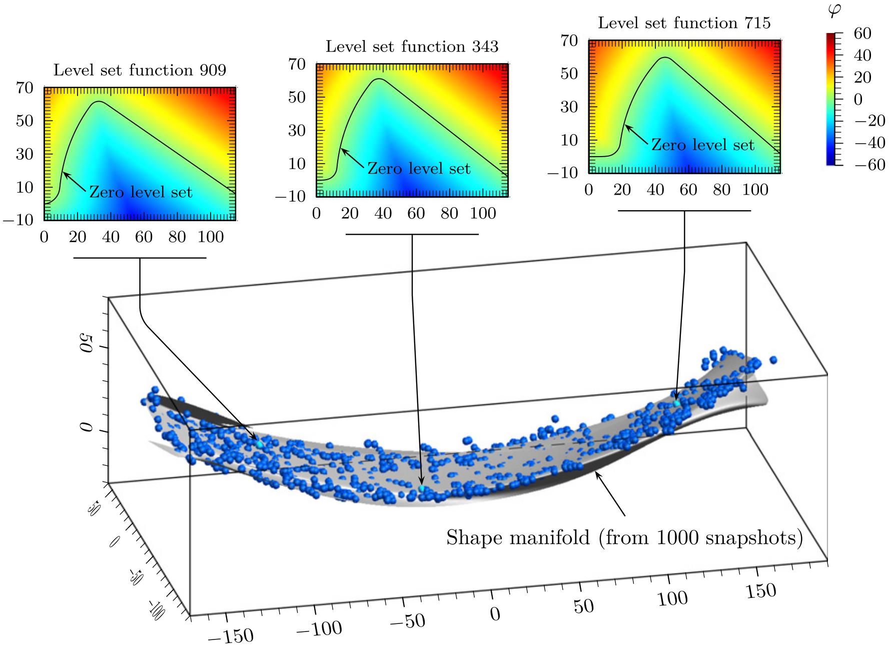

Fig. 2. Concept of shape manifold connecting all admissible shapes for a design problem.

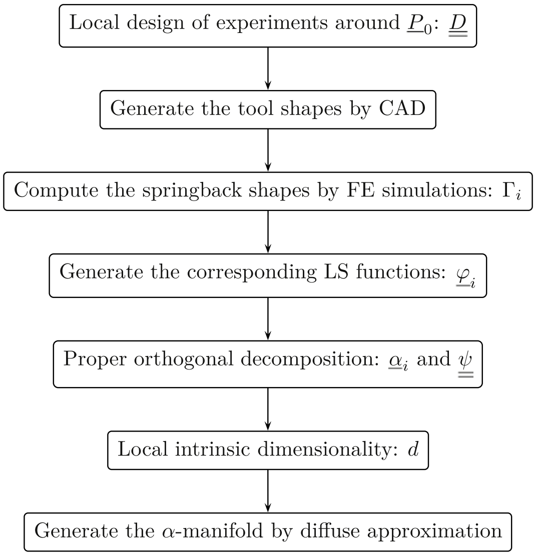
Local design of experiments around $\underline{P}_0$: $\underline{\underline{D}}$
Generate the tool shapes by CAD
Compute the springback shapes by FE simulations: Γ_i
Generate the corresponding LS functions: $\underline{\varphi}_i$
Proper orthogonal decomposition: $\underline{\alpha}_i$ and $\underline{\underline{\psi}}$
Local intrinsic dimensionality: d
Generate the α-manifold by diffuse approximation

Fig. 3. Approach proposed for the local shape manifold construction.

purposes resembles an actual post springback shape. This however is the strength of the approach when applied to level sets, since we can now identify shapes that are physically infeasible using the manifold.

2.3. Dimensionality reduction by Proper Orthogonal Decomposition

The Proper Orthogonal Decomposition method allows us to generate the lower dimensional orthogonal basis $\underline{\underline{\psi}}$, whose directions are referred to as POD modes, that is able to represent a given set of higher dimensionality solutions, called as snapshots, corresponding here to the level set functions $\underline{\varphi}_i$. Their corresponding coordinates $\underline{\alpha}_i$ in this basis $\underline{\underline{\psi}}$ are called the POD coefficients (shown in the fifth stage in Fig. 3). Here, we follow the POD method of snapshots without truncation.[1] The first step consists of calculating the mean snapshot $\underline{\varphi}_0$:

$$\underline{\varphi}_0 = \frac{1}{s}\sum_{i=1}^{s}\underline{\varphi}_i. \tag{5}$$

Then the centered snapshot matrix $\underline{\underline{S}}$ is generated:

$$\underline{\underline{S}} = \left[\underline{\varphi}_1 - \underline{\varphi}_0 \ \cdots \ \underline{\varphi}_i - \underline{\varphi}_0 \ \cdots \ \underline{\varphi}_s - \underline{\varphi}_0\right]. \tag{6}$$

The way the snapshots are linked together is given by the covariance matrix $\underline{\underline{C}}$:

$$\underline{\underline{C}} = \underline{\underline{S}}^{\mathrm{T}}\,\underline{\underline{S}}. \tag{7}$$

The eigenvectors $\underline{\underline{\phi}}$ and eigenvalues $\underline{\lambda}$ of $\underline{\underline{C}}$ are calculated and then sorted in the order of decreasing eigenvalue. These POD modes correspond to:

$$\underline{\underline{\psi}} = \underline{\underline{S}}\,\underline{\underline{\phi}}\,\underline{\underline{A}}^{-1}. \tag{8}$$

where $\underline{\underline{A}}$ is the square diagonal matrix of the singular values of $\underline{\underline{S}}$, that is $\sqrt{\lambda_i}$.

The POD coefficients of the input snapshots are now given by:

$$\left[\underline{\alpha}_1 \cdots \underline{\alpha}_i \cdots \underline{\alpha}_s\right] = \underline{\underline{A}}\,\underline{\underline{\phi}}^{\mathrm{T}}. \tag{9}$$

The POD of the input snapshots thus corresponds to:

$$\underline{\varphi}_i = \underline{\varphi}_0 + \underline{\underline{\psi}}\,\underline{\alpha}_i. \tag{10}$$

Finally, the more the chosen set of $\underline{\varphi}_i$ is sufficiently representative of the influence of the variable input parameters in their range of variation, the more any intermediate $\underline{\varphi}$ would be accurately generated from its corresponding POD coefficients $\underline{\alpha}$ when obtained by interpolation in the α-space from the s snapshots α_i:

$$\underline{\varphi} \approx \underline{\varphi}_0 + \underline{\underline{\psi}}\,\underline{\alpha}. \tag{11}$$

It is important to mention that a linear combination of a level set functions will in general not necessarily give us a valid signed distance function, as can be easily verified by the straightforward case of a level set function minus itself.

2.4. Detection of the local intrinsic dimensionality

At this point we have a set of s snapshots represented by their coordinates in α-space, *i.e.* $\underline{\alpha}_1, \ldots, \underline{\alpha}_s$.

Before we can move from a set of discrete points to a smooth manifold using interpolation, we need to first detect the dimensionality of this data set. This local intrinsic dimensionality $d \leqslant p$ may be evaluated around a given design point $\underline{\alpha}_{\mathrm{ev}}$, for a local neighborhood of $b \leqslant s$ points established by a standard nearest neighbor calculation, using the Fukunaga–Olsen algorithm [35].

The basic premise of this algorithm is simply dividing the data set into clustered regions/neighborhoods where the surface is approximately linear, and then compute the eigenvalues of the local moment matrix (by using a polynomial basis P and a matrix weighting function W to "weight" the contributions of points based on their distance from α_{ev}). The intrinsic dimensionality is then simply the number of eigenvalues that are greater than a certain threshold.

The polynomial matrix $\underline{\underline{B}}$ is assembled according to:

$$\underline{\underline{B}} = \begin{bmatrix} 1 & \underline{\alpha}_1^{\mathrm{T}} - \underline{\alpha}_{\mathrm{ev}}^{\mathrm{T}} \\ \vdots & \vdots \\ 1 & \underline{\alpha}_b^{\mathrm{T}} - \underline{\alpha}_{\mathrm{ev}}^{\mathrm{T}} \end{bmatrix}. \tag{12}$$

[1] The POD method of snapshots is faster when the number of snapshots is lower than their dimension, whereas truncation is useful when the number of snapshots is large.

The moment matrix $\underline{\underline{M}}$ is then computed from $\underline{\underline{B}}$ and a diagonal matrix $\underline{\underline{W}}$ whose components are the weighted contributions of the points $\underline{\alpha}_1, \ldots, \underline{\alpha}_b$:

$$\underline{\underline{M}} = \underline{\underline{B}}^{\mathrm{T}} \underline{\underline{W}}\, \underline{\underline{B}}. \tag{13}$$

Finally, the number of non-zero singular values of $\underline{\underline{M}}$ corresponds to the local intrinsic dimensionality d of the design problem around $\underline{\alpha}_{\mathrm{ev}}$.

2.5. *Interpolation by diffuse approximation*

At this point, we have determined the intrinsic dimensionality d for the neighborhood around α_{ev}. We now present the actual approach for making the transition from a set of b discrete snapshots $\underline{\alpha}_i$ to a smooth α-manifold

$$\underline{\mu}(\underline{\alpha}) = \underline{0} \tag{14}$$

using interpolation in α-space, either by passing through all the discrete $\underline{\alpha}_i$ points of the neighborhood or sufficiently close in a minimized weighted-error sense, e.g. moving least squares.

In order to perform the interpolation, we need to replace the implicit representation in the previous equation by an explicit representation as follows:

$$\underline{\alpha} = \underline{\alpha}(t_1, \ldots, t_d). \tag{15}$$

2.5.1. *Finding local coordinates for the local neighborhood*

The first question that we must answer is what are the parameters $t_1, \ldots, t_d$ that we can use to perform this interpolation in order to move from the b snapshots $\underline{\alpha}_i$ to a smooth representation.

The first obvious and straightforward choice for $\underline{t}$ is simply the vector of dimensionless design parameters ($\underline{P} = P_1 \cdots P_p$) representing the complete set of geometric variables ($G_1, G_2, \ldots$), material variables ($M_1, M_2, \ldots$) and process variables ($Pr_1, Pr_2, \ldots$) for the drawing operation, as mentioned in the introductory section. This is actually a viable option if $d = p$, in which case $\underline{P}$ of the original structural problem may themselves be considered as the interpolation variables (coordinates), considerably simplifying the entire procedure, since by simply varying $\underline{P}$ and calculating the local neighborhood, we seamlessly move from one local neighborhood to another.

On the other hand, when $d < p$ (*i.e.* the number of design parameters exceeds the intrinsic dimensionality) *or* if the structural parameters are unknown (as in the case of non-intrusive optimization) then it would be better to use an *implicit* parameterization of the α-manifold, *i.e.* perform the interpolation in an *implicit* manner by finding the *local coordinates* of $\underline{\alpha}_i$ *i.e.* $h, t_1, \ldots, t_d$ that describe the local shape of the manifold for that neighborhood, using tangent space construction. This local expression for the neighborhood will then be given by

$$h = \tilde{h}(t_1, \ldots, t_d) \tag{16}$$

where the smooth function $\tilde{h}(\underline{t})$ will be determined by interpolation using the natural coordinates h^i, $\underline{t}^i$ where $\underline{t}^i \in \mathcal{R}^d$ of the b neighboring snapshots.

The coordinates $\underline{t}^i$ correspond to the coordinates of the ith snapshot in the *local tangent space* of the α-manifold at the evaluation point $\underline{\alpha}_{\mathrm{ev}}$, while the complementary coordinate h^i is simply the normal distance of the snapshot from ith to this tangent space.

Thus, in order to find the local coordinates for the b neighboring points in α-space, we first calculate the covariance matrix $\underline{\underline{C}}_{\alpha}$ centered over the evaluation point $\underline{\alpha}_{\mathrm{ev}}$ for the local neighborhood:

$$\underline{\underline{C}}_{\alpha} = \frac{1}{b} \sum_{i=1}^{b} \left(\underline{\alpha}_i - \underline{\alpha}_{\mathrm{ev}}\right) \left(\underline{\alpha}_i - \underline{\alpha}_{\mathrm{ev}}\right)^{\mathrm{T}}. \tag{17}$$

The eigenvectors $\underline{v}_i$ of $\underline{\underline{C}}_{\alpha}$ correspond to the local principal directions in the local tangent space. The local coordinates for the neighborhood are then deduced by projection [36]:

$$h^i = [\underline{\alpha}_i - \underline{\alpha}_{\mathrm{cen}}]^{\mathrm{T}} \underline{v}_1 \tag{18}$$

$$\underline{t}^i_j = [\underline{\alpha}_i]^T \underline{v}_{j+1}. \tag{19}$$

where $\underline{\alpha}_{\text{cen}}$ is the centroid of the neighborhood in α-space.

In this manner, we have determined the local coordinates $\left(h^i, \underline{t}^i\right)$ for all the $i = 1, \ldots, b$ snapshots in the local neighborhood.

2.5.2. *Diffuse approximation on the local coordinates $h^i, \underline{t}^i$*

The next step is to perform interpolation on the local co-ordinates found in the previous step to find a smooth function $\tilde{h}(\underline{t})$. Generally speaking, there are several methods that may be used to perform this interpolation (*e.g.* kriging, cokriging, radial basis functions, moving least squares, *etc.*) As we have explained in the introduction, the diffuse approximation [37] is used here mainly due to the ease and flexibility that it allows in performing a moving least-squares fit with an appropriate weighting function to interpolate between a given number of points in a *local* neighborhood.

Diffuse approximation is then performed for a given point $\underline{\alpha}^{\text{A}}$ belonging to the local α-manifold (not one of the snapshots) using the b neighboring points to get the local surface $h = \tilde{h}(\underline{t}) = \tilde{h}\,(t_1, \ldots, t_d)$ using a polynomial basis $\underline{\underline{B}}$ centered around with the diagonal weighting matrix $\underline{\underline{W}}$:

$$\underline{\underline{B}}_{\text{t}} = \begin{bmatrix} 1 \; (\underline{t}^1 - \underline{t}^{\text{A}})^T & & \\ \vdots \quad\quad \vdots & & \cdots \\ 1 \; (\underline{t}^b - \underline{t}^{\text{A}})^T & & \end{bmatrix}. \tag{20}$$

The basic idea here is to represent $h\,(t_1, \ldots, t_d)$ by a moving polynomial $\tilde{h}\left(\underline{t}\right) = \underline{b}(\underline{t})^{\text{T}}\underline{a}(\underline{t})$ where $\underline{b}(\underline{t}) = [1, (\underline{t} - \underline{t}^{\text{A}})^T, \ldots]$ and minimizing the functional given by:

$$E\left(\underline{a}\right) = \frac{1}{2}\sum_{i=1}^{b} w_i\left(\underline{t}, \underline{t}^i\right)\left[\underline{b}\left(\underline{t}^i\right)^{\text{T}}\underline{a} - \tilde{h}\left(\underline{t}^i\right)\right]^2. \tag{21}$$

where w_i is the weighting function (elements of $\underline{\underline{W}}$) for the ith neighbor snapshot, giving us [38]:

$$\left[\tilde{h}\left(\underline{t}^{\text{A}}\right), \frac{\partial\tilde{h}\left(\underline{t}^{\text{A}}\right)}{\partial t_1}, \ldots \frac{\partial\tilde{h}\left(\underline{t}^{\text{A}}\right)}{\partial t_d}\right]^{\text{T}} = \left(\underline{\underline{B}}_{\text{t}}^{\text{T}}\underline{\underline{W}}\,\underline{\underline{B}}_{\text{t}}\right)^{-1}\underline{\underline{B}}_{\text{t}}^{\text{T}}\underline{\underline{W}}\,[h_1, \ldots, h_b]^{\text{T}}. \tag{22}$$

For a Gaussian weighting function this could be of the form:

$$w_i\left(\underline{t}, \underline{t}^i\right) = e^{-c\|\underline{t}^i - \underline{t}^{\text{A}}\|^2} \tag{23}$$

where c is a constant to control the weighting factor. This equation gives us $h, t_1, \ldots, t_d$ where h is height over the centroidal plane for the neighborhood. The local parametric expression for the α-manifold in this local neighborhood of using d local parameters $t_1, \ldots, t_d$ is then given as:

$$\underline{\alpha}_1 = \underline{\alpha}_1\,(t_1, \ldots, t_d)\,, \ldots, \underline{\alpha}_s = \underline{\alpha}_s\,(t_1, \ldots, t_d)\,. \tag{24}$$

This means that the projection coefficients are controlled locally by d parameters that allow us to move along the manifold. A typical α-manifold obtained by diffuse approximation is shown in Fig. 4.

It is vital to note that truncation of the POD basis is not required and not really recommended as it could reduce the accuracy of the method, especially considering the POD truncation error does not drop off quickly enough [30].

That being said, one could truncate the basis to $m \leqslant s$ most energetic modes using the energy criterion [20] in the traditional manner

$$\epsilon\,(m) = \frac{\sum\limits_{i=1}^{m}\lambda_i}{\sum\limits_{i=1}^{s}\lambda_i} \tag{25}$$

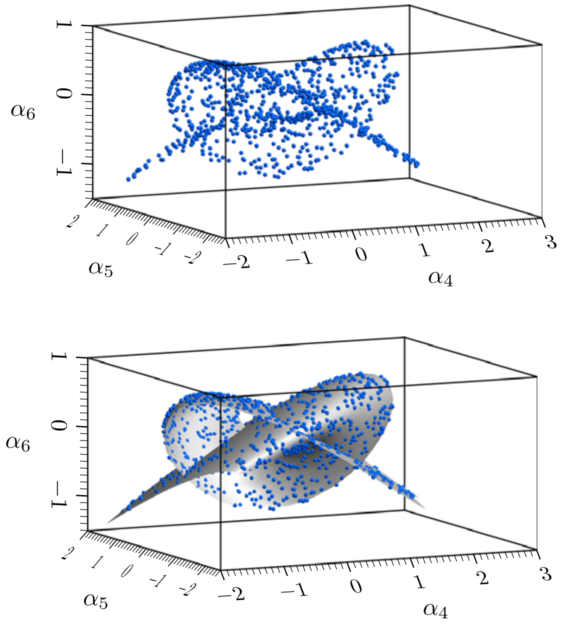

Fig. 4. Example of diffuse approximation on a set of shapes in the α-space (top) leading to a smooth α-manifold (bottom).

(where λ_i are the eigenvalues calculated in (7) and (8)) before developing the manifold in $\mathcal{R}^m$ space to further reduce computational effort.

It is also important to mention that the estimation of d needs a sufficiently close neighborhood [13,14], and the number of neighbors b needed depends on the expected intrinsic dimensionality and the order of polynomial chosen for the matrix $\underline{\underline{B}}$. For material forming problems, we can use the geometric dimensionality of the stamping/drawing tool and number of process parameters so these can give us an upper bound on d and therefore allow us to pick b.

2.6. *Manifold walking algorithm*

Simply projecting the target shape on the global α-manifold could directly lead to the desired solution, **if** the global manifold were available, and this would certainly be a way to solve the problem under certain circumstances.

However, the construction of this global α-manifold in the entire range of variation of the parameters $\underline{P}$ can become time consuming in complex cases, especially when the number of parameters p is large, due to the massive number of snapshots that would be needed. In order to reduce the total number of calls to the exact function and thus the global time cost, the α-manifold will be constructed "piece by piece", *i.e.* only locally, until the optimal solution is found.

As described in Fig. 5, the general approach starts off by choosing an initial point $\underline{P}_0$. The local shape manifold is constructed around this point (as previously described in Fig. 3). A local design of experiments $\underline{\underline{D}}$ is performed and the corresponding springback shapes Γ_i are computed. This is the **only time** we need exact function values. Next, the corresponding level set functions $\underline{\varphi}_i$ are generated for the shapes and POD is performed. After determining the local dimensionality d according to the previous sub-section, the local α-manifold is built up by the Diffuse Approximation. The coordinates $\underline{\alpha}_{\mathrm{T}}$ of the target shapes $\underline{\varphi}_{\mathrm{T}}$ are now expressed in this α-space. The projection of $\underline{\alpha}_{\mathrm{T}}$ onto this local α-manifold gives us the next set of parameters. This algorithm is repeated until convergence.

As one would expect, during this manifold walking process (Figs. 6–9), the overlapping of neighbors that increases with subsequent iterations leads to a reduction in the number of evaluations needed as we approach the target shape.

3. 3D Test-case: Post-springback shape characterization for deep drawing operation on an automotive strut tower

3.1. *Description of test-case and numerical analysis*

In this 3D test-case, we consider the deep-drawing of a metal sheet to form an automotive strut tower with a set of material parameters corresponding to a basic steel grade.

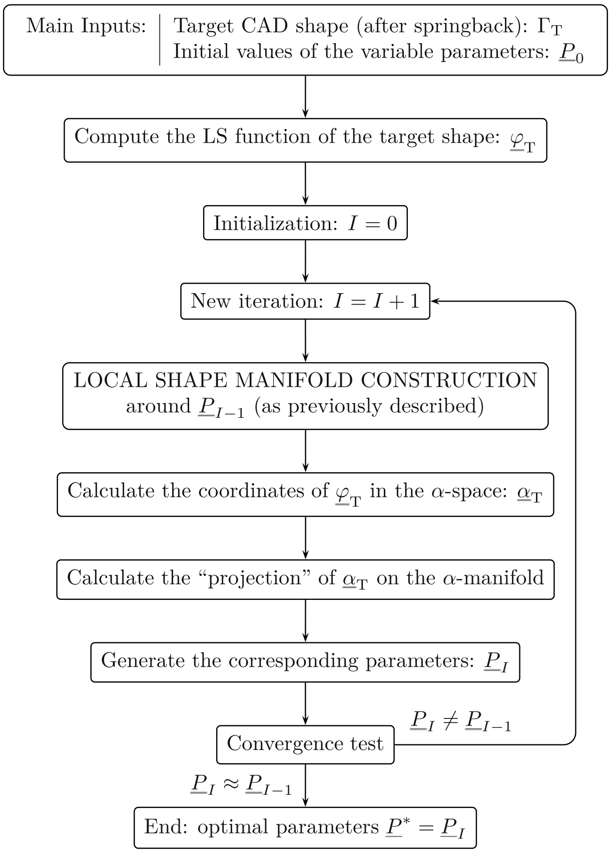
Main Inputs:
Target CAD shape (after springback): Γ_{T}
Initial values of the variable parameters: $\underline{P}_0$
Compute the LS function of the target shape: $\underline{\varphi}_{\mathrm{T}}$
Initialization: $I = 0$
New iteration: $I = I + 1$
LOCAL SHAPE MANIFOLD CONSTRUCTION around $\underline{P}_{I-1}$ (as previously described)
Calculate the coordinates of $\underline{\varphi}_{\mathrm{T}}$ in the α-space: $\underline{\alpha}_{\mathrm{T}}$
Calculate the "projection" of $\underline{\alpha}_{\mathrm{T}}$ on the α-manifold
Generate the corresponding parameters: $\underline{P}_I$
Convergence test
$\underline{P}_I \neq \underline{P}_{I-1}$
$\underline{P}_I \approx \underline{P}_{I-1}$
End: optimal parameters $\underline{P}^* = \underline{P}_I$

Fig. 5. General approach proposed for the shape optimization in metal forming after springback.

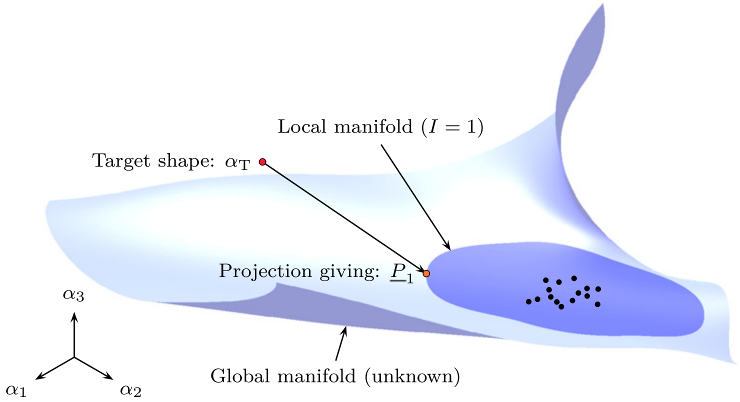
Local manifold ($I = 1$)
Target shape: α_{T}
Projection giving: $\underline{P}_1$
α_3
α_1
α_2
Global manifold (unknown)

Fig. 6. First stage of the manifold walking algorithm: neighborhood around initial point.

The geometries of the tools and the blank, shown in Fig. 10, were designed using the legacy CAD software TopSolid/Design [39]. The geometries are fully parameterized but only 1 (H) or 2 (H and R) parameters were allowed to vary according to the values of their associated dimensionless parameters $\theta_1 \in [0; 1]$ and $\theta_2 \in [0; 1]$:

$$H = 3 \cdot \theta_1 + 5 \tag{26}$$

$$R = 6 \cdot \theta_2 + 41. \tag{27}$$

The CAD geometries were translated into meshes with the finite element mesh generator GMSH [40]. The forming phase of the simulations was performed in dynamic explicit mode using the legacy finite element software

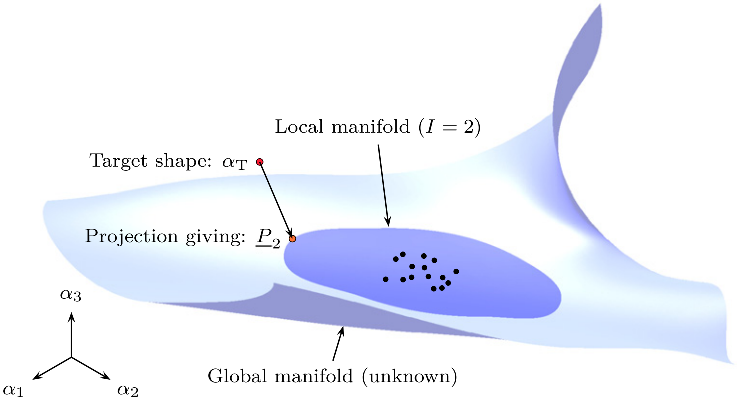

Fig. 7. Second stage of manifold walking.

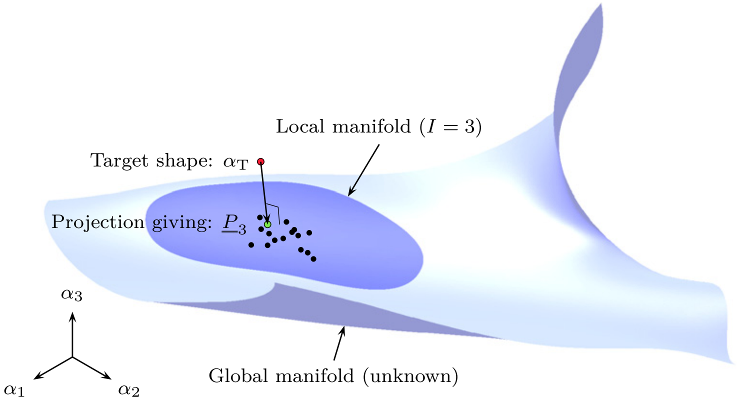

Fig. 8. Intermediate stage of manifold walking: local manifold continues to move closer to target point in α-space.

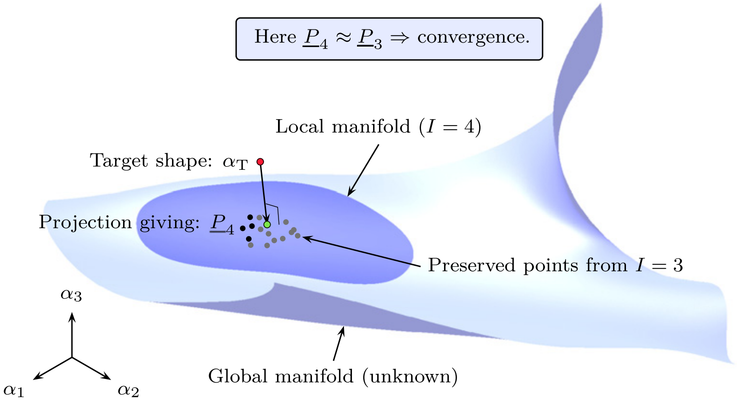
Here $\underline{P}_4 \approx \underline{P}_3 \Rightarrow$ convergence.
Local manifold ($I = 4$)
Target shape: α_T
Projection giving: $\underline{P}_4$
Preserved points from $I = 3$
α_3
α_1
α_2
Global manifold (unknown)

Fig. 9. Local neighborhood clusters overlap towards convergence.

LS-DYNA [41], using the Belytschko–Tsay shell element formulation and 5 through thickness integration points. A Python procedure was written to translate the GMSH mesh files in the LS-DYNA keyword file format.

Only one half of the problem was modeled due to its symmetric nature. Corresponding boundary conditions are imposed on the blank side. The mesh is initially composed of a total of 16 114 elements with an adaptive remeshing level limited to 2. The tools (punch, blank holder and die) were modeled as rigid body surfaces. The blank holding force was held fixed at 500 kN. The blank was modeled considering a basic steel grade with transversely anisotropic

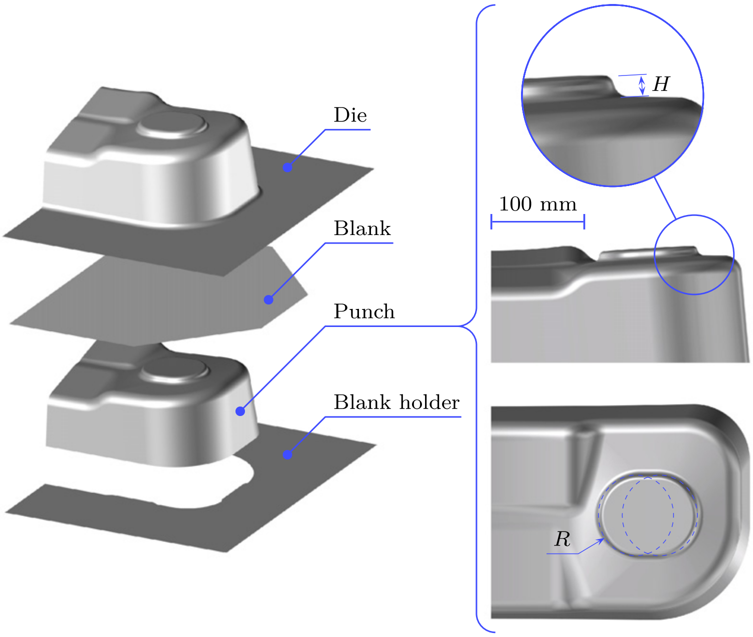

Fig. 10. Fully parameterized CAD geometries with 2 varying parameters H and R.

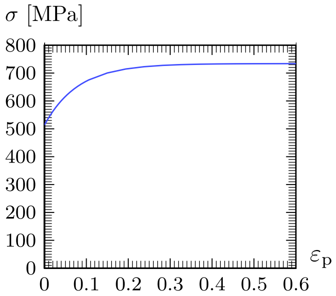

Fig. 11. Hardening curve of the blank (basic steel grade).

Table 1
Main test-case parameters and material properties of the blank for a transversely anisotropic elastic–plastic material behavior.

Test case parameters and material properties		
Blank holding force	F_b	500 kN
Friction coefficient	μ	0.1
Thickness	t	2.2 mm
Young's modulus	E	200 GPa
Poisson's ratio	ν	0.3
Density	ρ	7830 kg/m^3
Anisotropic parameter*	r	0.817

* The anisotropic parameter corresponds to the ratio of the in-plane plastic strain rate to the out-of-plane plastic strain rate.

elastic–plastic material behavior (type 37 in LS-DYNA). All the related parameters are listed in Table 1. The hardening curve of the blank is given in Fig. 11.

The springback phase of the simulations was performed in static implicit using the same legacy code. Only the blank mesh was saved for this phase. One of its nodes was fully constrained (6 degrees of freedom) in order to

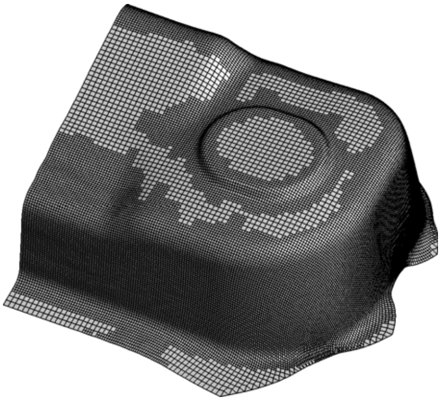

Fig. 12. Typical finite element simulation result.

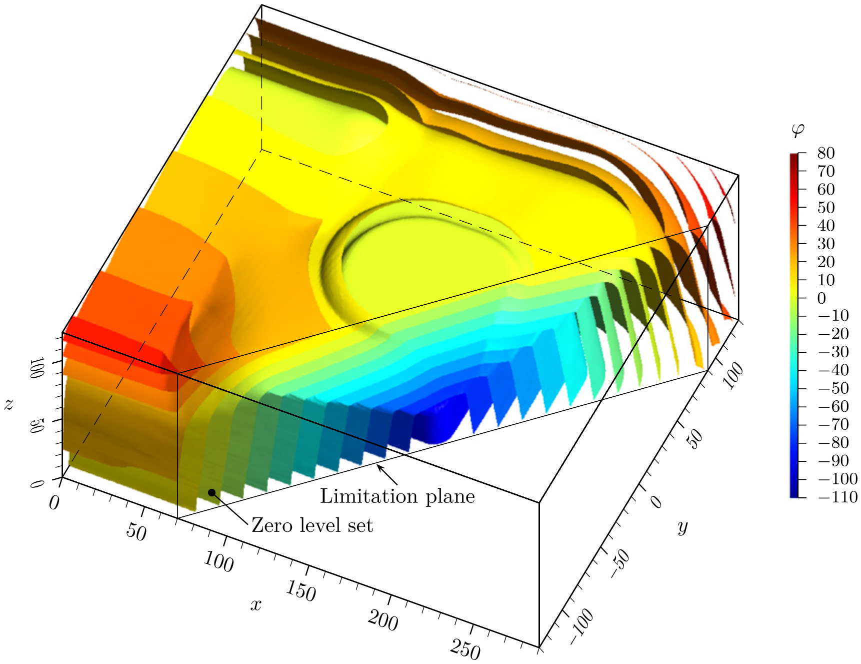

Fig. 13. Typical level set function result (grid resolution 146 × 131 × 63).

avoid any rigid body movement. Artificial stabilization was activated to achieve a progressive unloading, to allow springback to occur over 10 steps. A typical post-springback shape obtained by finite element simulation using LS-DYNA is shown in Fig. 12.

3.2. Level set representation

The level set functions φ for sample formed shapes and for the target shape are obtained quickly by calculating the signed distance function associated with the shapes and a fixed grid of points with resolution (146 × 131 × 63). To compute the level set in a fast way, we developed a dedicated procedure in C++ language able to treat CAD geometries (with help of Open CASCADE libraries [42]) as well as meshes. A typical level set function of a post-springback shape is shown in Fig. 13.

As explained earlier, φ is the smooth signed distance function from the 3D surface of the particular post springback shape used for the figure. The figure shows the various level sets corresponding to different values of this signed

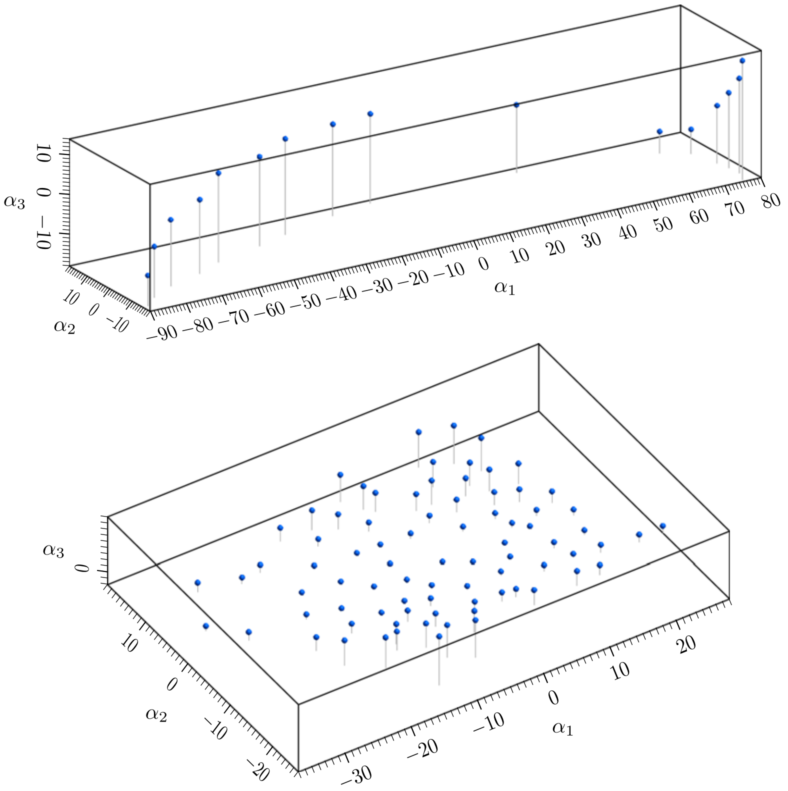

Fig. 14. Shape sampling projection in α-space in the case of 1 (top) and 2 (bottom) variable parameters of the tool geometry.

distance (represented using a continuum of color variation), while the iso-zero surface ($\varphi = 0$) corresponds to the actual post springback shape surface. Note that Fig. 13 plots the level sets against a "limitation plane" solely to show them more clearly, as in the absence of this "section" one would be unable to clearly distinguish between the individual level sets.

3.3. *Using the level-set shape manifolds for springback characterization and to estimate tool geometry required to obtain a desired final shape after springback*

As explained in the preceding section, the shape manifold is constructed *locally* around the design point. To obtain the manifolds around a design point in the range of the tool parameters, the analysis of springback for the strut tower was first performed for a sampling of input shapes in LS-DYNA by varying first a single parameter (H *via* θ_1), followed by varying 2 tool parameters (H and R *via* θ_1 and θ_2) in the neighborhood of the design point as described in the Section 3.1, to obtain the actual post-springback shapes obtained for the purpose of geometry characterization. These were then converted to the corresponding level set functions $\underline{\varphi}_1 \dots \underline{\varphi}_s$ using the methods described in Section 3.2.

This was followed by POD on the shapes $\underline{\varphi}_1 \dots \underline{\varphi}_s$ to give a set of projection coefficients $\underline{\alpha}_1 \dots \underline{\alpha}_s \in \mathcal{R}^s$ allowing us to plot the shape manifolds for a set of post-springback shapes in Fig. 14.

3.4. *Estimating tool geometry required to obtain a desired final shape after springback using manifold walking*

As mentioned in the introduction, the intended final shape differs from the actual final shape obtained post-springback. The basic idea is to locate the point on the manifold $\underline{\varphi}\left(\underline{\alpha}\left(\underline{P}\right)\right)$ that is closest to the intended final shape (which may or may not be an admissible shape) $\underline{\varphi}_{\mathrm{T}}$ without constructing the entire global manifold since that would entail additional computational effort.

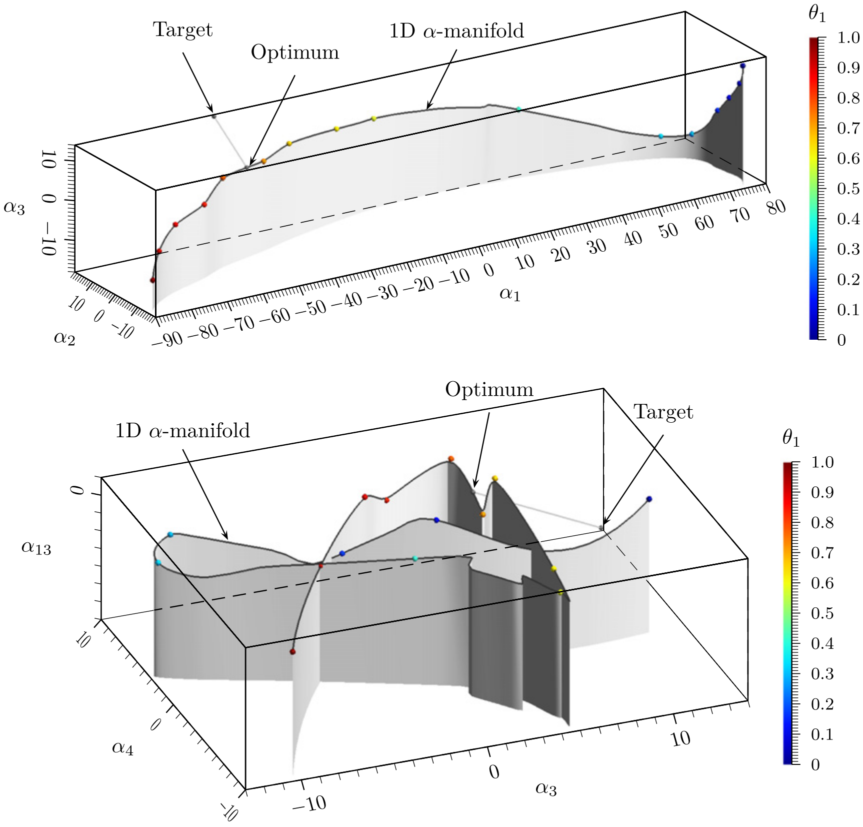

Fig. 15. Finding the optimal design point on the local α-manifold in the case of a single variable.

Let $\underline{\alpha}^*$ be the projection of the ideal (desired) shape on the manifold, *i.e.* $\varphi^* = \varphi\left(\underline{\alpha}^*\right) = \varphi\left(\underline{\alpha}\left(\underline{P}^*\right)\right)$. Then the problem may be posed in the form:

$$\underline{P}^* = \arg\min_{\underline{P}\in\mathcal{A}\subset\mathcal{R}^p} \left\|\underline{\alpha}\left(\underline{P}\right) - \underline{\alpha}_{\mathrm{T}}\right\|. \tag{28}$$

The algorithm when applied using a given desired shape gives the final solutions shown in Fig. 15 for the 1 parameter case and Fig. 16 for the 2 parameter case. As expected, these final shapes are both admissible since they lie on the manifold (the desired shape in *this particular case* on the other hand is **not** admissible since it clearly does not belong to the manifold).

We locate the design points corresponding to the tool geometries needed to achieve these final shapes, which are optimally close to the desired shapes for both cases, by locating the point on the manifold closest to the projection of the desired shape on the local neighborhood using the manifold walking algorithm in each case.

Finally, a comparison of the target desired (non admissible) shape and the zero level set corresponding to the closest admissible shape post-springback (reconstructed from the optimum α-coefficients and the POD modes) is shown in Fig. 17.

While we have varied 1 and 2 geometric parameters to demonstrate the approach it goes without saying that the approach is valid for any number of geometric and/or process parameters, and as we have seen it is independent of the complexity of the structural shape. While we have limited ourselves to the forming phase in the simulation, it is important to note that the trimming operation (to remove flash) generally has a significant influence on the springback. This operation has not been considered here but the proposed approach would then be applied to the final shapes obtained post forming *and/or* post trimming. This would of course increase the number of potential design parameters and thus the intrinsic dimensionality of the problem.

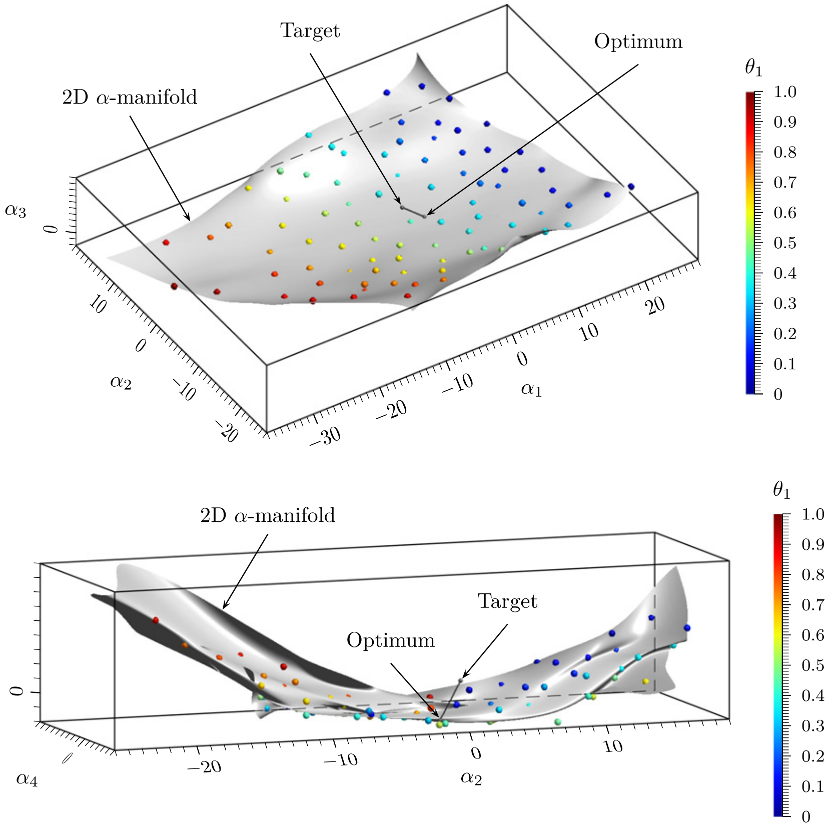

Fig. 16. Finding the optimal design point on the local α-manifold in the case of 2 tool geometry variables.

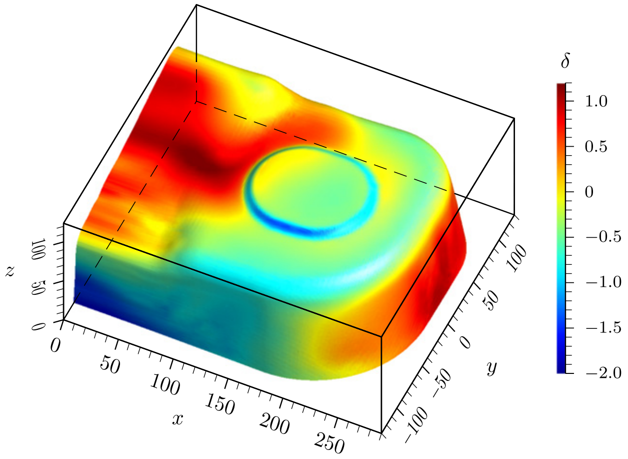

Fig. 17. Comparison of the target shape to the closest admissible shape in the case of 2 tool geometry variables.

4. Conclusions

In this paper, we have presented an objective approach to characterize the complex non-geometric 3D shapes obtained after springback in a deep drawing process, using the level set approach and a shape space meta-model based on the fundamental hypothesis of the continuous shape manifold. The transition from a "point-set manifold" to a smooth shape manifold is then achieved by interpolating between level set functions in POD coefficient-space using the Diffuse Approximation. The approach was then demonstrated in the problem of springback assessment for the deep drawing process of an automotive strut tower.

This allows us to achieve a variety of goals in a single shot: we can directly determine the tool geometry needed for a desired final post-springback shape needed by the manufacturer, thus being better able to take springback into account at the tool design stage.

Level sets, arguably the best tool to represent complex and "non geometric" shapes like those post-springback, suffer from excessive dimensionality due to the number of grid points needed for adequate precision (1,204,938 in the test case in this article) This is alleviated by the use of the POD that allows us to represent this massive information using only a handful of αs for a given shape.

But from a storage point of view, this requires us to store the 15–20 snapshots in level set form, even though this basis stays FIXED for a neighborhood allowing us to generate an infinite number of "non snapshot" shapes using the manifold. In any case, the storage issue can conceivably be resolved by using octree meshes to limit the number of grid points while maintaining precision in the zones susceptible to variation, and this is an area of future work.

Finally, the biggest challenge is still finding a way to integrate our level set interpolation approach into the Hamilton–Jacobi level set equation to yield an ODE in the projection (α) coefficients in order to actively track the deformation of the contour using "moving" local snapshots.

Acknowledgments

This research was conducted as part of the OASIS project, supported by OSEO within the contract FUI no. F1012003Z.

The authors acknowledge the support of Labex MS2T, which was funded by the French Government, through the program "Investments for the future" managed by the National Agency for Research (Reference ANR-11-IDEX-0004-02).

References

[1] H. Hayashi, T. Nakagawa, Recent trends in sheet metals and their formability in manufacturing automotive panels, J. Mater. Process. Technol. 46 (1994) 455–487.

[2] P.-A. Eggertsen, K. Mattiasson, On the modelling of the bending–unbending behaviour for accurate springback predictions, Int. J. Mech. Sci. 51 (2009) 547–563.

[3] K. Li, W. Carden, R. Wagoner, Simulation of springback, Int. J. Mech. Sci. 44 (2002) 103–122.

[4] Y. Moon, S. Kang, J. Cho, T. Kim, Effect of tool temperature on the reduction of the springback of aluminum sheets, J. Mater. Process. Technol. 132 (2003) 365–368.

[5] I. Alfaro, J. Yvonnet, E. Cueto, F. Chinesta, M. Doblare, Meshless methods with application to metal forming, Comput. Methods Appl. Mech. Engrg. 195 (2006) 6661–6675. Computational Metal Forming.

[6] T. Gao, W. Zhang, P. Duysinx, A bi-value coding parameterization scheme for the discrete optimal orientation design of the composite laminate, Int. J. Numer. Methods Eng. 91 (2012) 98–114.

[7] R. Teimouri, H. Baseri, B. Rahmani, M. Bakhshi-Jooybari, Modeling and optimization of spring-back in bending process using multiple regression analysis and neural computation, Int. J. Mater. Form. (2012) 1–12.

[8] L. D'Acquisto, L. Fratini, An optical technique for springback measurement in axisymmetrical deep drawing operations, J. Manuf. Process. 3 (2001) 29–37.

[9] G. Le Quilliec, P. Breitkopf, J. Roelandt, P. Juillard, Semi-analytical approach for plane strain sheet metal forming using a bending-under-tension numerical model, Int. J. Mater. Form. (2013) 1–12.

[10] J. Lebon, G. Le Quilliec, P. Breitkopf, R. Filomeno Coelho, P. Villon, A two-pronged approach for springback variability assessment using sparse polynomial chaos expansion and multi-level simulations, Int. J. Mater. Form. (2013) 1–13.

[11] Y.Q. Guo, J.L. Batoz, H. Naceur, S. Bouabdallah, F. Mercier, O. Barlet, Recent developments on the analysis and optimum design of sheet metal forming parts using a simplified inverse approach, Comput. Struct. 78 (2000) 133–148.

[12] B. Raghavan, G. Le Quilliec, P. Breitkopf, A. Rassineux, J.-M. Roelandt, P. Villon, Numerical assessment of springback for the deep drawing process by level set interpolation using shape manifolds, Int. J. Mater. Form. (2013).

[13] B. Raghavan, P. Breitkopf, Y. Tourbier, P. Villon, Towards a space reduction approach for structural shape optimization, Struct. Multidiscipli. Optim. (2013).

[14] B. Raghavan, L. Xia, P. Breitkopf, A. Rassineux, P. Villon, Towards simultaneous reduction of both input and output spaces for interactive simulation-based structural design, Comput. Methods Appl. Mech. Engrg. (2013).

[15] X. Xie, M. Mirmehdi, Radial basis function based level set interpolation and evolution for deformable modelling, Image Vis. Comput. 29 (2011) 167–177.

[16] S. Wang, K. Lim, B. Khoo, M. Wang, An extended level set method for shape and topology optimization, J. Comput. Phys. 221 (2007) 421.

[17] A. I. J. Forrester, A.J. Keane, Recent advances in surrogate-based optimization, Prog. Aerosp. Sci. 45 (2009) 50–79.

[18] P. Breitkopf, H. Naceur, A. Rassineux, P. Villon, Moving least squares response surface approximation: Formulation and metal forming applications, Comput. Struct. 83 (2005) 1411–1428.

[19] K. Willcox, J. Peraire, Balanced model reduction via the proper orthogonal decomposition, AIAA J. 40 (2002) 2323–2330.

[20] G. Berkooz, P. Holmes, J.L. Lumley, The proper orthogonal decomposition in the analysis of turbulent flows, Annu. Rev. Fluid Mech. 25 (1993) 539–575.

[21] L. Cordier, B.A. El Majd, J. Favier, Calibration of pod reduced order models using tikhonov regularization, Internat. J. Numer. Methods Fluids 63 (2010) 269–296.

[22] C. Audouze, F. De Vuyst, P. Nair, Reduced-order modeling of parameterized pdes using time-space-parameter principal component analysis, International Journal of Numerical Methods in Engineering 80 (2009) 1025–1057.

[23] B. Raghavan, P. Breitkopf, Asynchronous evolutionary shape optimization based on high-quality surrogates: application to an air-conditioning duct, Eng. Comput. (2012).

[24] F. Chinesta, A. Ammar, E. Cueto, Proper generalized decomposition of multiscale models, Int. J. Numer. Methods Eng. 83 (2010) 1114–1132.

[25] F. Chinesta, P. Ladeveze, E. Cueto, A short review on model order reduction based on proper generalized decomposition, Arch. Comput. Methods Eng. 18 (2011) 395–404.

[26] C. Ghnatios, F. Chinesta, E. Cueto, A. Leygue, P. Breitkopf, P. Villon, Methodological approach to efficient modeling and optimization of thermal processes taking place in a die: Application to pultrusion, Composites A 42 (2011) 1169–1178.

[27] C. Ghnatios, F. Masson, A. Huerta, E. Cueto, A. Leygue, F. Chinesta, Proper generalized decomposition based dynamic data-driven control of thermal processes, Comput. Methods Appl. Mech. Engrg. 213 (2012) 29–41.

[28] D. Ryckelynck, A priori hypereduction method:an adaptive approach, J. Comput. Phys. 202 (2005) 346–366.

[29] M. Hamdaoui, G. Le Quilliec, P. Breitkopf, P. Villon, POD surrogates for real-time multi-parametric sheet metal forming problems, Int. J. Mater. Form. (2013).

[30] B. Raghavan, P. Breitkopf, P. Villon, Pod-morphing: an a posteriori reparametrization approach for shape optimization, Eur. J. Comput. Mech. 19 (2010) 671–697.

[31] B. Raghavan, M. Xiao, P. Breitkopf, P. Villon, Implicit constraint handling for shape optimization using pod-morphing, Eur. J. Comput. Mech. 21 (2012) 325–336.

[32] D. Millan, A. Rosolen, M. Arroyo, Nonlinear manifold learning for meshfree finite deformation thin-shell analysis, Int. J. Numer. Methods Eng. 93 (2013) 685–713.

[33] D. Millan, A. Rosolen, M. Arroyo, Nonlinear manifold learning for model reduction in finite elastodynamics, Comput. Methods Appl. Mech. Engrg. 261–262 (2013) 181–131.

[34] A. Ibrahimbegovic, I. Gresovnik, D. Markovic, S. Melnyk, T. Rodic, Shape optimization of two-phase material with microstructure, Int. J. Eng. Comput. 22 (2005) 605–645.

[35] K. Fukunaga, D. Olsen, An algorithm for finding intrinsic dimensionality of data, IEEE Trans. Comput. 20 (1971) 176–183.

[36] I.T. Jolliffe, Principal Component Analysis, Springer Verlag, Berlin, 2002.

[37] B. Nayroles, G. Touzot, P. Villon, Generalizing the finite element method: diffuse approximation and diffuse elements, Comput. Mech. 10 (1992) 307–318.

[38] P. Breitkopf, A. Rassineux, G. Touzot, P. Villon, Explicit form and efficient computation of mls shape functions and their derivatives, International Journal For Numerical Methods In Engineering 48 (2000) 451–466.

[39] www.topsolid.com 2013.

[40] C. Geuzaine, J.-F. Remacle, Gmsh: a three-dimensional finite element mesh generator with built-in pre- and post-processing facilities, Int. J. Numer. Methods Eng. 79 (2009) 1309–1331.

[41] J.O. Hallquist, LS-DYNA Theory Manual, Livermore Software Technology Corporation, 2006.

[42] www.opencascade.org 2013.