Springback Compensation for Incremental Sheet Metal Forming Applications

J. Zettler[1], H. Rezai[1], G. Hirt[2]
[1] EADS Deutschland GmbH, München
[2] Institut für Bildsame Formgebung der RWTH Aachen, Aachen

Abstract. Incremental CNC sheet forming (ISF) and incremental sheet forming with incremental die (ISFID) are relatively new sheet metal forming process for small batch production and prototyping. In these processes, a blank is shaped by the CNC movements of simple tools in combination with a simplified die or without die at all. The standard forming strategies for the traditional ISF process normally use some kind of die in order to increase the accuracy of the final part. However the ISF process and the ISFID introduce great residual stresses to the sheet during the forming which lead to geometrical deviations after releasing the fixation of the sheet. These residual stresses vary based on material selection and geometry of the formed part.

This paper deals with a procedure to account for the geometrical deviations based on the residual stresses by applying a compensation to them with a software tool which takes care of special ISF and ISFID requirements like maximum wall angle or minimal radii. As this software tool is part of a software designed for ISFID at EADS Innovation Works, several links and specialities of this process variation are used for the compensation tool described in this paper and might not be applicable to the standard and traditional ISF process.

1 Introduction

The geometric deviations related to residual stresses are a common problem in cold sheet metal forming processes. After removing the sheet from the shaping tools or the blank holder the elastic energy inside the sheet vanishes and as a result the sheet begins to distort. This distortion, often called springback, leads to a geometrical deviation compared to the desired CAD shape.
To compensate the springback especially for deep drawing processes a lot of efforts have been spent during the last years and numerous papers and research results exist already [6]. The ISFID and the ISF process however are still in a niche but growing and some researchers already tried to reuse the basic ideas of springback compensation based on a comparison and scaled mirroring of the CAD geometry and the geometry after the forming ([3]). In this paper we will focus on a new developed software within EADS Innovation Works which allows for a compensation of the springback after ISFID and ISF. The procedure differs mainly from that of other research activities by taking care of ISFID and ISF characteristics which have been implemented in the software. These characteristics include maximum forming angle and minimum radii of the forming part. The software can either be used in a FE simulation based springback compensation loop or in an iterative trial series. In figure 1 the basic principle of the software is shown.
We have a CAD geometry which represents the target of the compensation. In ISF and ISFID we need a tool path to form our part. Therefore the first step is to generate a tool path based on the CAD geometry. After forming the part either by FE simulation with LS-DYNA (see [7]) or by real trials, the final geometry after springback is measured. For the simulation output this is not necessary but for the real trials an optimcal measurement system called ATOS from the company GOM is used. The next step, Best-Fit, is one of the most important steps in the whole procedure as it is responsible for how much deviation we will measure in different areas of the part. In section 2 we will explain in more detail the exact procedure we use here.

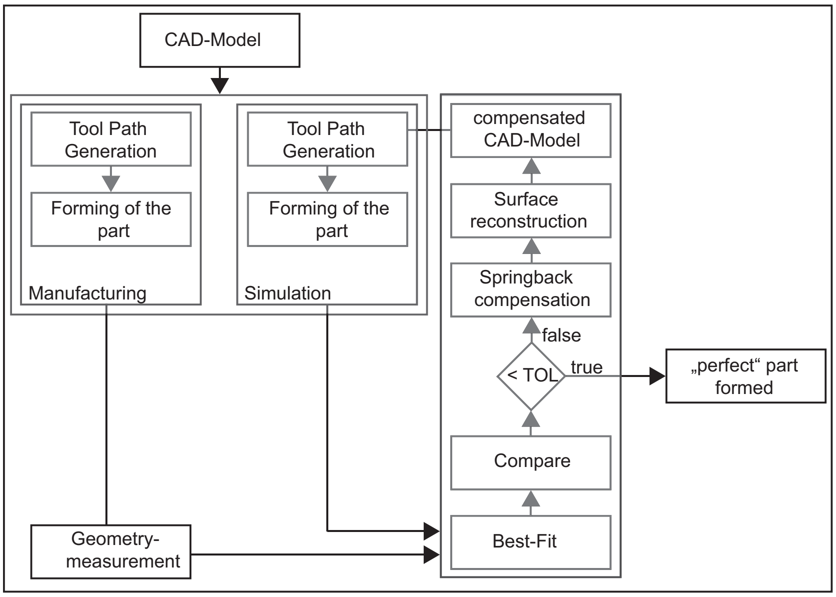


Figure 1: *Springback Compensation procedure*

During the geometrical comparison between the CAD geometry and the formed geometry a global error, which equals the discretized point to point distance between the two geometries, is calculated and compared to a given tolerance set by the user. This tolerance represents the global error the user is accepting. If the global error is not yet below the tolerance limit, the springback compensation will take place. This function is especially designed for the ISF process and implements the characteristics mentioned above. Section 3 covers this topic in detail. Before the complete springback compensation procedure is finished and a compensated geometry can be used to generate new and changed tool paths, we perform a surface reconstruction of the compensated geometry. This step is necessary because during the compensation step we work with meshed geometries. This means, that the geometries are represented by triangulated meshes and therefore don´t have an analytic representation of the surface normal which leads to an inaccurate normal vector field which results in problems during the tool path generation for ISFID (see [7] for more details on the tool path generation for ISFID). In section 4 the mathematical procedure used for the surface reconstruction is explained in detail.

After the surface reconstruction is finished the compensated surface can be used for the next iteration loop until the global error is below the error tolerance set by the user.

2 Best Fit

After the forming, either virtually by FE simulation or in trials, we get a triangulated geometry representation of the formed part. To calculate the geometrical deviation between this geometry and our desired CAD geometry we have to make sure that the relative position of the CAD geometry and our formed geometry is aligned to each other in a proper way. We call this procedure „Best Fit“ as we search for the optimal relative position between two different geometries which have similar and representative regions. For the ISF process it is often necessary to take into account special areas of the geometry which should be preferred by the alignment procedure. An example for such a special area is the region in which the die or the blank holder is acting. With our algorithm this area can be weighted differently than the other parts of the geometry and the „Best Fit“ will lead to a slightly better positioning towards the prefered areas. For our algorithm we first have to triangulate our CAD model to speed up the whole procedure. This triangulation is directly implemented in our software as well. Afterwards we now assume two given triangulations. One from the CAD model called $\mathcal{D}_S$ with the point-set $Q = \{q_i\}_{i=1,\ldots,m}$ and $\mathcal{D}$ of the formed geometry with the point-set $P = \{p_i\}_{i=1,\ldots,n}$. The algorithm uses a weighted version of the *ICP-Method* (ICP $\hat{=}$ **I**terative **C**losest **P**oint) wich is used to find the perfect match between the both triangulations. A detailed description of this method can be found in [4]. The aim of this method is to compute a combination of rotations and translations of the point set P wich minimizes the square-distances between the two point sets Q and P. As already mentioned above, by using weights for single points of the point set P the user can influence the resulting orientation. In Fig. 2 different weights have been used and the changed „Best Fit“ position of the formed geometry can be seen.

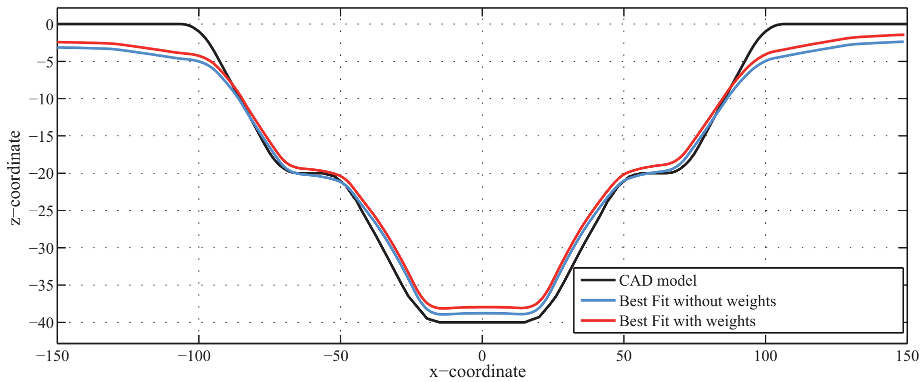


Figure 2: *Cut through a CAD geometry compared to two „Best Fit“ formed geometries with and without weights on the boundary of the geometry*

In our implementation we allow a preselection and definition of weights for single regions of the CAD model by directly selecting CAD B-Spline patches. Internally we take all the nodes of the CAD triangulation which refer to the B-Spline patch and apply the desired weight to them.
To ensure a robust and fast convergence of the algorithm, it is necessary to start with a coarse orientation of the triangulation $\mathcal{D}$ compared to $\mathcal{D}_S$. For this reason, before starting the ICP-algorithm, we compute the principal axes of the two point sets Q and P and transform $\mathcal{D}$ in a way that the pricipal axes fit to each other . The principal axes can be computed by setting up a covarianz matrix of the point sets Q and P and using the characteristic polynomial of that matrix to acquire the Eigenvectors and Eigenvalues. They already represent the principal axes.
After the coarse alignment we can compute an assignment $\psi : Q \to \hat{P} \subseteq P$ between the two point sets Q and P by using a nearest-neighbour-search-algorithm. With the given weights w_i, $i = 1, \ldots, m$ we have to solve the following nonlinear minimization problem

$$\sum_{i=1}^{m} w_i \|r_i\|_2^2 \longrightarrow \min \quad , \quad r_i := R \cdot \psi(q_i) + t - q_i$$

with a 3×3 rotational matrix R and a translation-vector $t \in \mathbb{R}^3$.

The solution is found by using the *Levenberg-Marquardt-Algorithm*. Afterwards the mesh will be transformed with the solution and the assignment ψ is computed again for the transformend point-set. The iteration will be cancelled if the average square distance error

$$E = \frac{1}{m}\sum_{i=1}^{m} \|q_i - \psi(q_i)\|_2^2$$

has no significant change in comparison to the previous iteration step. In figure 3 the speed of the algorithm is shown. Already after around 10-20 iterations the average error is nearly minimized.

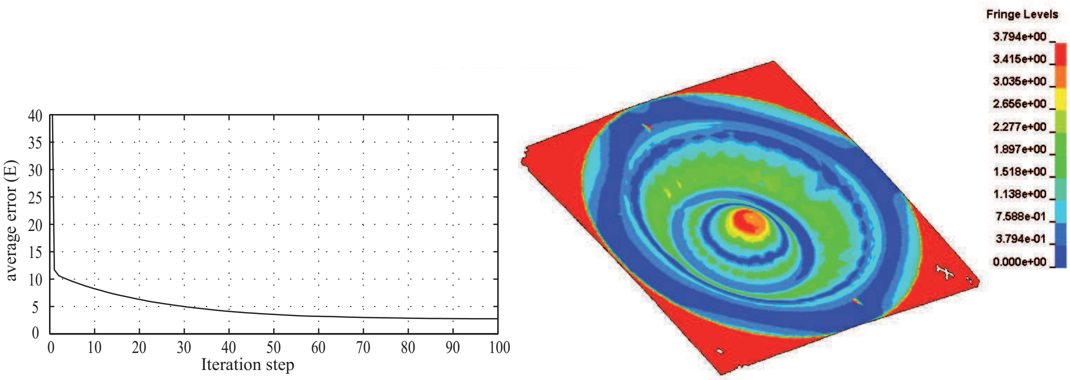


Figure 3: *Average Error in relation to the „Best Fit" iteration steps (left) and resulting overall error in [mm] of the „Best Fit" positioning for an example part with more weights on the boundary (right)*

3 Springback Compensation

After the formed geometry is „Best Fit“ towards the CAD part with the user set weights in special areas, the actual springback compensation is performed. This algorithm is optimised for the ISF process and therefore different from state of the art springback compensation methods described in [6]. The main difference is, that the geometry after the springback compensation is still formable by the ISF process without any tool change. This is achieved by taking care that the compensated geometry will never result in wall angles greater then a user defined value ϕ_{TOL} and also radii will never be less then the forming tool radius. These special ISF related restrictions can be extracted from the curvature information of the CAD model and attached to the CAD mesh $\mathcal{D}_S$. As during the compensation procedure the distance between the CAD geometry and the formed geometry has to be measured we also need the analytical surface normal of the CAD geometry and attach it to our CAD mesh $\mathcal{D}_S$ at the proper nodes $q_i \in Q$. For further explanations let $V := \{1, \dots, m\}$ contain the index values of the point set Q and $V_b \subset V$ represents the point indices within the boundary region of the CAD-model.
The single steps of the compensation procedure can be summarised as follows:

- For every knot $i \in V$ of the mesh $\mathcal{D}_S$ with the analytical unit normal vector n_i from the CAD geometry, compute a distance vector $f_i \in \mathbb{R}^3$ through intersection with $\mathcal{D}$ in normal direction.
- After computing a curvature based bound $s_i \geq 0$ for every knot $i \in V$, taking into account the maximum allowable wall angle ϕ_{TOL} , all nodal points $q_i \in Q$ will be transformed (**global** offset) using

$$q_i^* = q_i - \theta \cdot \mu_i \cdot f_i$$

  with $\theta \in [0, 1]$ and $\mu_i \in [0, 1]$. θ represents the scaling factor. If we would mirror the calculated and measured error this factor will be 1.0. Based on experiences this factor usually varies between 0.8 and 1.0 (see [3]).
- Determine connected regions of triangles based on the triangle normals which can be moved further without exceeding the angle tolerance ϕ_{TOL} (**local** offset) and transform each of these separate regions separately.
- The last step includes a smoothing procedure of the transformed triangulations $\mathcal{D}_S$ to obtain a smooth interaction between the local offsetted areas and the global offsetted areas.

3.1 Global Offset

The CAD unit normal vectors n_i at the nodal points $q_i \in Q$ determine the direction of the node movement and are also necessary to compute the distance to the triangulation $\mathcal{D}$ by intersectioning it. As already mentioned above, for accuracy reasons, these normals will be computed directly on the parametric CAD surface $S(u, v)$ with

$$n_i := \frac{S_u(u_i, v_i) \times S_v(u_i, v_i)}{\|S_u(u_i, v_i) \times S_v(u_i, v_i)\|_2}, \quad (u_i, v_i) := S^{-1}(q_i)$$

The surface parameters $(u_i, v_i) \in \mathbb{R}^2$ are stored during the triangulation process of the CAD surface and we assume them to be given.
The key aspect of the introduced algorithm is the computation of an upper bound $s_i \geq 0$ for the translation amount. Based on the principal curvatures of the CAD model at all knots $i \in V$ the upper bound s_i will be determined in a way, that the approximative computed radii of curvature of the compensated geometry will not exceed the tool radii set by the user. The paramater $\lambda_{\phi_{TOL}} \in [0, 1]$ is computed, based on the triangle normals of all knots $i \in V$. To ensure that the triangles of the deformed triangulation will not exceed the angle tolerance ϕ_{TOL} .
The distance vector f_i is always collinear to n_i and the magnitude in the first iteration step equates to the euclidean distance to $\mathcal{D}$. To ensure a continous improvement during the iterative process of the springback compensation, all precending compensation loops will be added at this stage (including the signs) to the actual distance vector f_i .

We finally define the global offset for all knots $i \in V$ with

$$q_i^* = q_i - \theta \cdot \mu_i \cdot f_i \quad , \quad \mu_i := g_i \cdot \lambda_{\phi_{TOL}} \cdot \frac{\min(\|f_i\|_2, s_i)}{\|f_i\|_2} \tag{1}$$

To make sure that the outer boundary of $\mathcal{D}_S$ stays unchanged as we have to clamp it by the blank holder in further iterations, we introduce a weighting of the transformation-amount for all knots $i \in V_b$. Similar to the „Best Fit“ algorithm in section 2 we allow a preselection of regions of the CAD-model wich have to stay unchanged (see also section 5 for an application of this feature). The points inside the selected region will not be transformed at all and the boundary of that region will be handled like the outer boundary of the CAD-model and the knots of the adjacent regions will be added to V_b. The weightings g_i are defined by a smooth function over the the boundary distances d_i, l_i wich returns low values of almost zero for points near the outer boundary and values of almost one for points near the inner boundary of the CAD-model (see fig. 4)

$$g_i = \begin{cases} \frac{1}{2}\left[1 + sin\left(\left(\frac{1}{2} - \frac{d_i}{(d_i+l_i)}\right)\pi\right)\right] & \text{if } i \in V_b \\ 1 & \text{else} \end{cases} \tag{2}$$

Note: $\frac{d_i}{(d_i+l_i)} \in [0,1]\,, (2) \Rightarrow g_i \in [0,1] \quad , \quad g_i\,, \lambda_{\phi_{TOL}} \in [0,1]\,, (1) \Rightarrow \mu_i \in [0,1]$

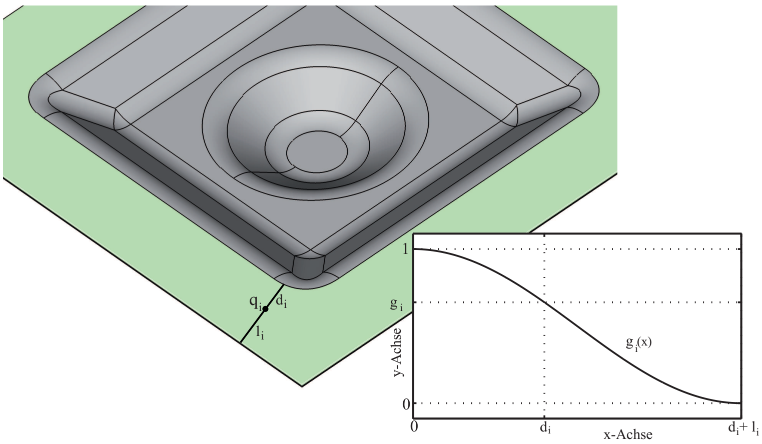


Figure 4: *Weighting function which vanishes towards the boundary*

3.2 Local Offset

After performing the global offset using (1) the next step is to determine connected subnets $V_i \subset V$, $i = 0, \ldots, n$ of the transformed mesh (region-growing) with potential of a further offset without exceeding the maximum formable angle ϕ_{TOL} . In figure 5 the white marked areas can be offsetted a little bit further while maintaining the boundary conditions like minimal radii and maximum wall angle.

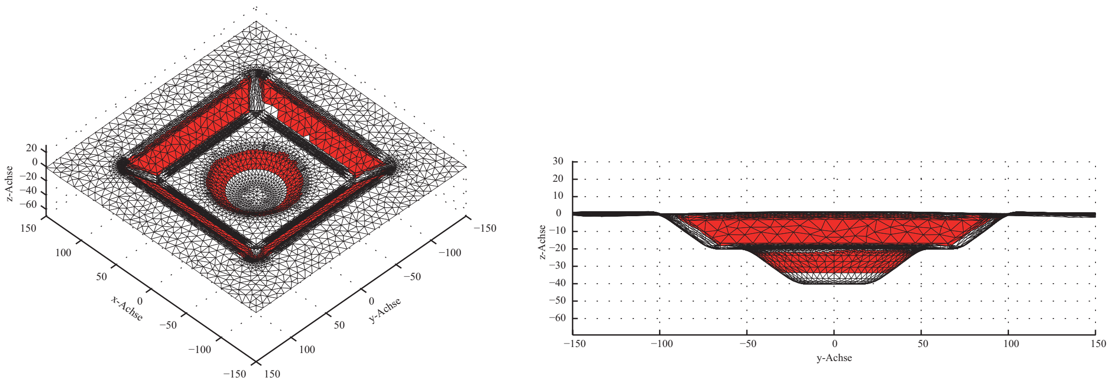


Figure 5: *White sub areas detected by region growing which allow for further movement*

The local offset of the subnet V_k will now be defined with

$$q_i^{**} = q_i^{*} - \theta \cdot h_i \cdot \lambda_{k,\phi_{TOL}} \cdot (1-\mu_i) \cdot f_i \quad , \quad h_i = \frac{1}{2}\left[1 + sin\left(\left(\frac{t_{i,k}}{t_{max,k}} - \frac{1}{2}\right)\pi\right)\right] \quad , \forall\, i \in V_k \qquad (3)$$

To avoid non smooth transitions between the different local areas and the rest of the triangulated geometry, the tranformation amount will also be weighted using a descending function to the boundary similar to (2). The function-parameter $t_{i,k} \geqq 0$ defines the minimum boundary-distance of the node $q_i \in Q$ concerning V_k while $t_{max,k} := \max_{i \in V_k} t_{i,k}$ stands for the maximum distance. Similar to $\lambda_{\phi_{TOL}}$ the parameter $\lambda_{k,\phi_{TOL}} \in [0,1]$ is computed, based on the triangle normals of the knots $i \in V_k$.
A result of the local offset is illustrated in fig. 6. A comparison between figure 5 and figure 6 illustrates very detailed the additional movement of several areas after the global offset due to the local offset.

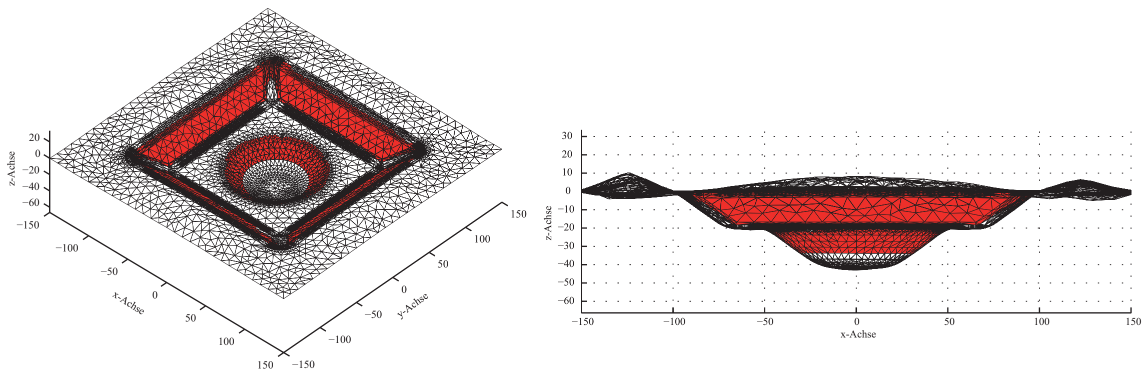


Figure 6: *Final result after global and local springback procedure.*

3.3 Mesh-Smoothing

The deformed CAD mesh resulting from the springback compensation may appear inhomogeneous, especially in areas with very scattered curvature values, and therefore suffer loss of accuracy. Therefore we perform a smoothing step on the compensated CAD mesh subsequent to the springback compensation. For the reason of simplicity for the further explanations we keep the notation $\mathcal{D}_S$ for the resulting triangulation alike the point set Q and the index set V.
The smoothing step will not proceed in a given sequence but for all nodal points simultaneous. For one single point $q_i \in Q$ the corresponding smoothed point $\hat{q}_i$ results from the orthogonal projection onto the equalization plane of the adjacent points $q_j \in Q_i$.

$$\hat{q}_i = q_i - \lambda \cdot m_i \qquad , \forall\, i \in V$$

with

$$\lambda := min(\lambda_{max}\,,\, \|m_i \cdot q_i - m_i \cdot S_i\|_2)\,, \quad S_i = \frac{1}{|Q_i|}\sum_{q \in Q_i} q$$

To control the distortion between the smoothed mesh and the mesh before the smooth step, an additional parameter λ_{max} is used which acts as the maximum allowable movement value between given and smoothed node.
We also don´t want to smooth sharp edges if they fullfill a special need in the CAD part. Therefore the smooth step is only performed inside areas which do not surpass a threshold angle β_{max}.

4 Surface reconstruction

The final step of our springback compensation is related to surface reconstruction as we need an analytical description of the surface to generate a tool path of high quality in the subsequent iterations especially for ISFID. Therefore the aim of this section is to find an analytical surface description wich approximates a given (in our case the smoothed and compensated CAD mesh from the last chapter) point-set $Q = \{q_i\}_{i=1,\ldots,m}$ under a given error-tolerance $\epsilon > 0$. For the representation we decided to use bicubic B-Spline surfaces $S : U \times V \to \mathbb{R}^3$ of the form

$$S\,(u,v) = \sum_{i=0}^{r}\sum_{j=0}^{s} N_{i,3}\,(u)\,N_{j,3}\,(v)\,\mathbf{P}_{ij} \tag{4}$$

We are looking for an approximating B-spline surface of the form (4) wich minimizes the square error

$$\mathbf{E}_d(S) = \sum_{k=1}^{m}\left(q_k - S\,(u_k,v_k)\right)^2 = \sum_{k=1}^{m}\left(q_k - \sum_{i=0}^{r}\sum_{j=0}^{s} N_{i,3}\,(u_k)\,N_{j,3}\,(v_k)\,\mathbf{P}_{ij}\right)^2$$

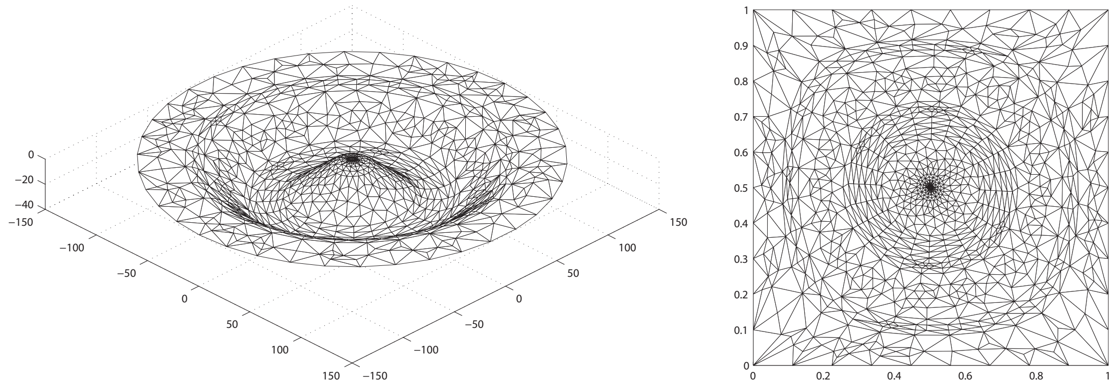
Simply minimizing the functional $\mathbf{E}_d(S)$ often leads to unwanted oscillations of the surface. In Computer Aided Graphics Design (CAGD) the smoothness of a surface is often described by using energy functionals (see [1]). In our implementation we decided to used the simplified thin plate energy

$$\mathbf{E}_g(S) = \int_0^1\int_0^1 S_{uu}^2 + 2S_{uv}^2 + S_{vv}^2 \; du\,dv \tag{5}$$

The idea is to extend the original minimization problem described by 5 with an additional weighted term $\mathbf{E}_d(S)$ with a weight $\lambda > 0$ which is simply added. This results in the following equation.

$$\mathbf{E}_d(S) + \lambda \mathbf{E}_g(S) \to \min \tag{6}$$

The functionals $\mathbf{E}_d(S)$ and $\mathbf{E}_g(S)$ are both quadratic in the unknown control points $\mathbf{P}_{ij}$ and thus the minimization (6) leads to a linear system of equations. To set up this equation system we need to estimate the parameter values (u_i, v_i) for every nodal point $q_i \in Q$ (see fig. 7).



(a) (b)

Figure 7: *Parameterization to the unit-square (b) of a 3D triangular mesh (a)*

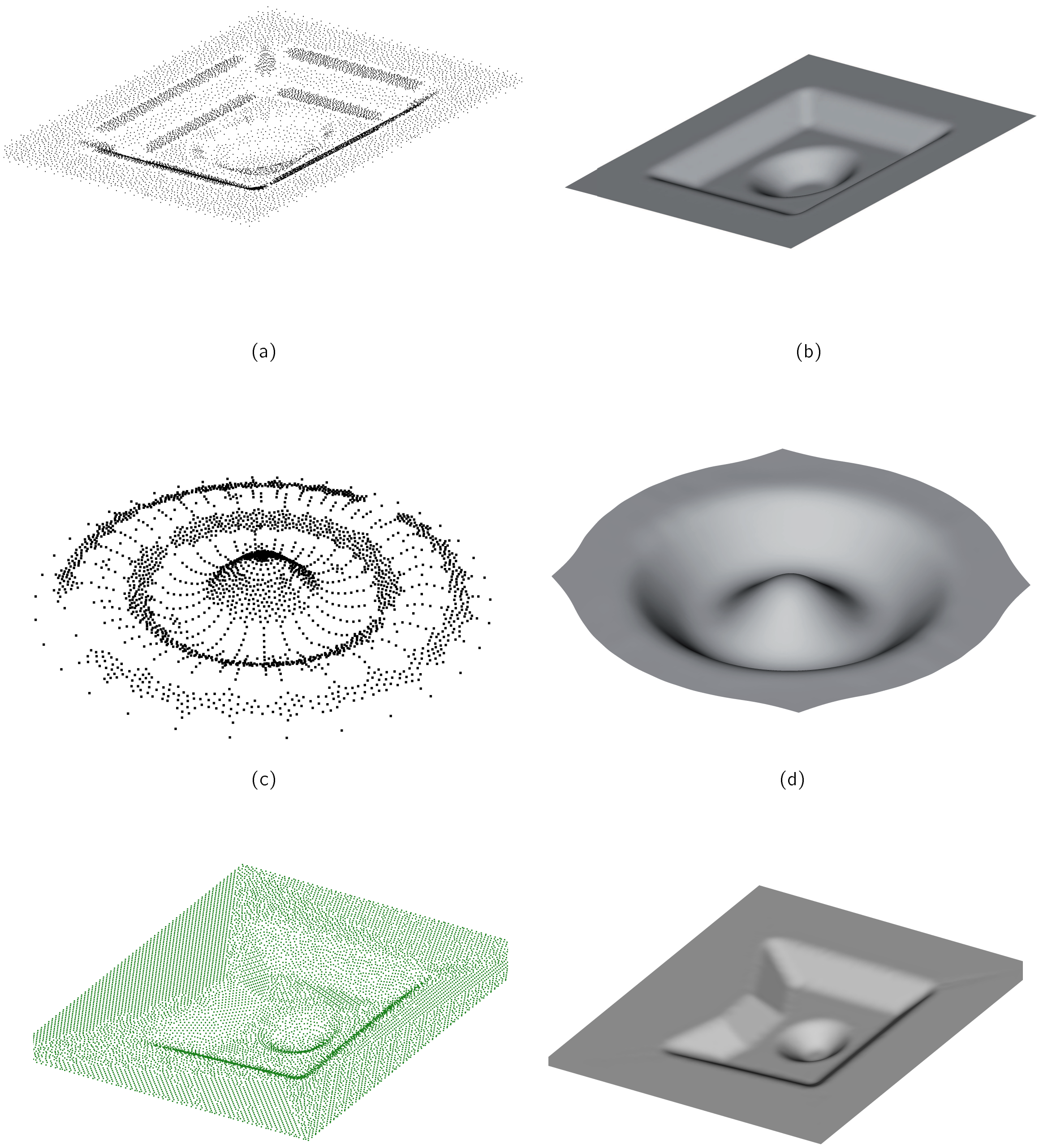
It is a difficult task to estimate the parameters and the parameter choice largely effects the result (see [5]). Therefore, iterative parameter correction procedures have been suggested (see [1]). In our application we used a parametrization method found in [2] wich requires a given triangulation over the point set Q.
This method mainly proceed in two steps:

- Map the boundary points of the mesh properly on an convex polygon (e.g. Unit-Square) wich decribes the boundary of our parameter region
- Set up and solve a linear system of equations wich demands that every internal parameter can be defined as a strict convex combination of its neighbours.

The weight λ controls the impact of the energy functional $\mathbf{E}_g$ within the minimization problem and therefore determines the smoothness of the approximating surface S. The determination of an appropriate value for λ is difficult and hence is performed iteratively. The iteration begins with a high value $\lambda \gg 1$ and will be bisected after every iteration step until the average error meets the condition $\mathbf{E}_d(S) \leqq \epsilon$.
The iteration loop of the complete algorithm includes the steps

- smooth linear approximation of the point set Q
- parameter correction
- halve the weight λ

In figure 8 the result of the iterative approximation method can be seen. These figures just represent test geometries for three different surface triangulations.

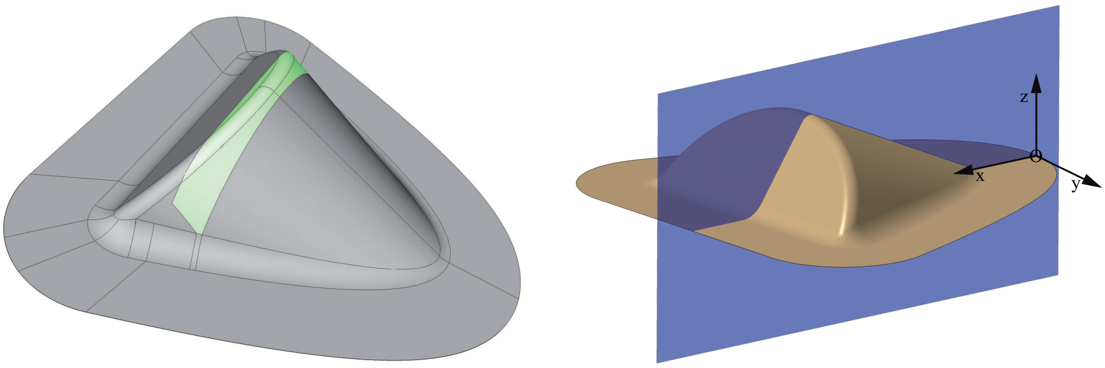


(a) (b) (c) (d) (e) (f)

Figure 8: *Approximating B-Spline-Surface (b,d,f) and the related 3D surface mesh (a,c,e)*

5 Validation

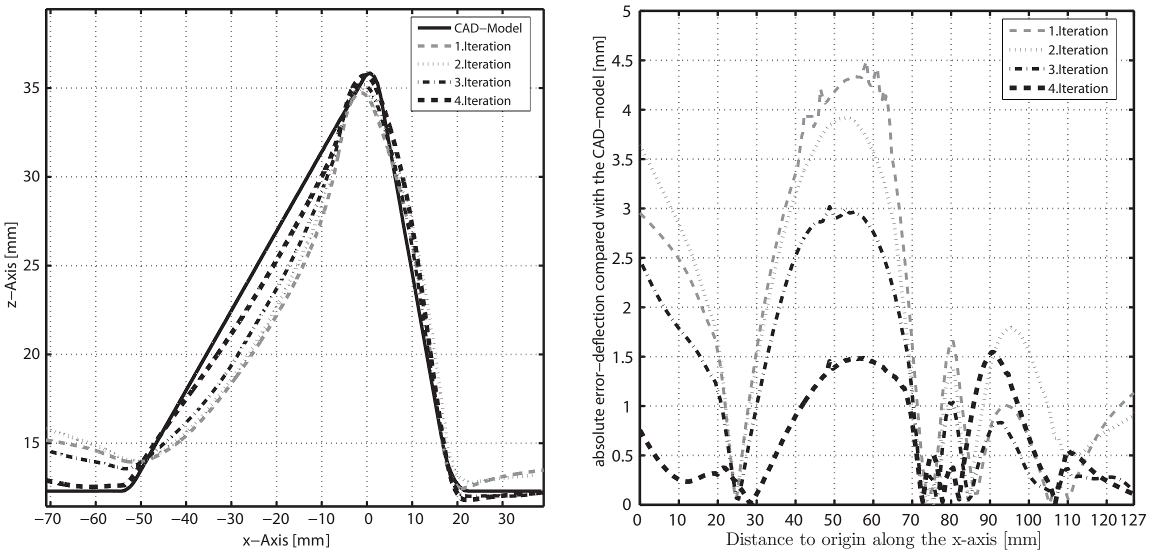
In the last sections we explained in detail the several steps of our compensation method. In this section we will focus on a benchmark part which is formed by a trial series using our springback compensation software. As the springback compensation is an iterative process the elastic deflection compared to the CAD model in every iteration step should be reduced. We evaluated our implementation with a total of four iterations using a benchmark part which is supported by a partial die. To ensure that the resulting surface of the springback compensation will not interfere with the partial die, and as a result damage the tool and the die, we preselected the specific regions to stay unchanged during the whole iteration process of the springback compensation (see fig. 9).



(a) (b)

Figure 9: *CAD model of the benchmark part. Green marked regions represent the supported area by the partial die for the ISF forming process (Fig. 9a)*

Every iteration step includes a best fit orientation, a springback compensation, mesh smoothing, surface reconstruction and a tool-path-generation using the corrected compensated CAD geometry. The absolute deflection error for the benchmark part has been measured in every iteration step along the X-axis (Fig. 10).



(a) (b)

Figure 10: *Cross section cut along the X-axis (Fig. 10a) and the absolute error deflection in each iteration(Fig. 10b)*

After the 4th iteration the overall error of the benchmark part is already below 1mm.

6 Summary and Outlook

The main focus of this paper is the introduction of a new springback compensation procedure especially designed for incremental sheet forming with an incremental die (ISFID) [7]. The software is able to compensate geometries which are formed either by FE simulation or by real trials. In every step of the compensation procedure special features related to the ISFID process are used. These include at first a weighted „Best Fit“ procedure to account for special areas in a forming geometry. Afterwards the core algorithm, the springback compensation, is taking special care of the maximum wall angle and the minimal radii of the resulting comepensated geometry. Without these special features an industrial application for the ISFID process is very time consuming as the user has to double check after every iteration loop if the compensated geometry is still formable or might lead to cracks due to high wall angles. Also the minimum radii feature we implemented is very useful because no tool changes are necessary and self intersections in the final compensated geometry are not possible at all. To use the software also for the traditional ISF process with partial die or full die, we implemented several features to select the areas which are supported by a die. This is necessary as otherwise the toolpath based on the compensated geometry may lead to a damage either in the forming tool or in the die. In the validation section 5 we tested our software on a partial die supported traditional ISF part and the results show very clear that we can reach an overall accuracy of around 1mm for the benchmark part by using our new developed procedures and routines.
There are still some drawbacks in using this kind of springback procedure. One of the most important is the fact that from iteration to iteration the final result will reach the desired accuracy but also become „wavyness“. This is due to the overbending we do in each iteration. Some ideas to compensate that already exist and will be implemented in the near future. Another topic related specifically to our procedure is the fact that the springback limit for several parts is reached quite fast if we take care of maximum wall angle. Imagine a part which already has a 60° wall angle somewhere. Now the material specific „crack“ wall angle is given with around 70°. After the first iteration with our sofware the software can only perform a compensation of around 5% to not overcome the wall angle limit, even if a lot more would have been necessary. On the one hand side this is a good result because we have to avoid cracks during the forming, but on the other hand side this means that the part will still be distorted after the following iterations as well. The springback compensation will therefore never give the desired accuracy and as a result this part cannot be formed easily by using the standard ISF process and our software.

7 Literatur

[1] Dietz, U. : *Geometrie-Rekonstruktion aus Meßpunktwolken mit glatten B-Spline-Flächen*. Shaker Verlag, 1998

[2] Floater, M. S.: Parameterization and smooth approximation of surface triangulations. In: *Computer Aided Geometric Design* 14 (1997), S. 231–250

[3] Hirt, G. ; Bambach, M. ; Chouvalova, I. ; Junk, S. : FEM modelling and optimisation of geometric accuracy in incremental CNC sheet forming. In: *CIRP Int. Conference on Accuracy in Forming Technology ICAFT*, Verlag Wissenschaftliche Scripten, 2003 (10. Sächsische Fachtagung Umformtechnik SFU 2003), S. 293–304

[4] Horn, B. K. P.: Closed-form solution of absolute orientation using quaternions. In: *Optical Society of America* 4 (1987), S. 629–642

[5] Kruth, J. P. ; Weiyin, M. : Parametrization of randomly measured points for the least squares fitting of B-spline curves and surfaces. In: *Computer Aided Design* 27 (1995), S. 663–675

[6] Lemke, T. ; Roll, K. : Compensation of Springback - State of the Art and next steps. In: *Forming Technology Forum 2007 - Application of Stochastics and Optimization Methods*, 2007, S. 49 – 54

[7] Zettler, J. ; Rezai, H. ; Taleb-Araghi, B. ; Bambach, M. ; Hirt, G. : Incremental Sheet Forming - Process Simulation with LS-DYNA. In: *6. LS-DYNA Anwenderforum, Frankenthal 2007*, DYNAmore GmbH, 2007

