# Trim Curve Development in Forming Simulation

Pablo Hernandez-Becerro[1], Xinhai Zhu[1]

[1]ANSYS, Inc

## 1 Introduction

Numerical simulations are increasingly used to design and optimize sheet metal forming manufacturing processes [1]. ANSYS FORMING® [2] offers a comprehensive, process-centric solution for metal stamping simulation. It simplifies the setup of multi-stage forming simulations, making it accessible and efficient for users. Relying on the ANSYS LS-DYNA® multiphysics solver, the platform offers easy job submission and monitoring. The interface integrates post-processing capabilities that allow for detailed analysis of the resulting part.

A critical step in setting up the sheet metal forming process is designing the right trimming curves for the blanking and trimming operations. In a multi-stage forming process, blanking or trimming operations are interspersed with drawing or forming operations. The geometry of the final part is determined by a combination of the trimming curves and the material flow. Therefore, selecting the optimal trim curves to achieve the final target geometry is not a straightforward process, often requiring the iterative adjustment of the trimming curves. This process is known as trim line development. With the support of simulation tools, engineers can perform the trim line development process in a virtual environment until the part satisfies the design requirements.

ANSYS FORMING provides a dedicated module for trim line development to facilitate the design of the blanking and trimming operations. For this analysis, the user provides the target contour of the part for the selected trim curves that need to be developed. The solver iteratively adapts the trim curves until the contour of the final part is within the tolerance specified by the user from the target contour. A dedicated post-processor shows the trimming curves and deviations from the target contour for each iteration. After the analysis is completed, the user can export the optimal trim curves.

This paper presents recent enhancements related to the trimming line development module. In the first section, we present a high-level overview of the trim line development interface in the ANSYS FORMING platform. The next section of the paper introduces automatic shifting, which allows the user to automatically shift the center of the trimming curve to the optimal location. The following section presents a new functionality that facilitates the development of sections of the trimming curve instead of developing the full curve. This functionality eliminates the work required in splitting the trimming lines in the CAD editor and provides additional robustness to the solver. The final section addresses miscellaneous controls available to the user to achieve the desired results. All these enhancements are introduced with meaningful case studies that help the user understand the usage and demonstrate the accuracy of the resulting solution.

## 2 Overview of the trimming line development interface in ANSYS FORMING

ANSYS FORMING provides a convenient user interface to optimize the trimming and blanking operations. In the process tab, the user selects the analysis *Trim Line Development* under the Advance tab. Similarly to a normal forming simulation, the user defines the blank and sets up the operations of the forming process, including the blanking and trimming operations. Once the forming process is fully defined, the user can set up the trimming line development in the Advance tab. Figure 1 illustrates the interface provided to define the trimming line development within ANSYS FORMING.

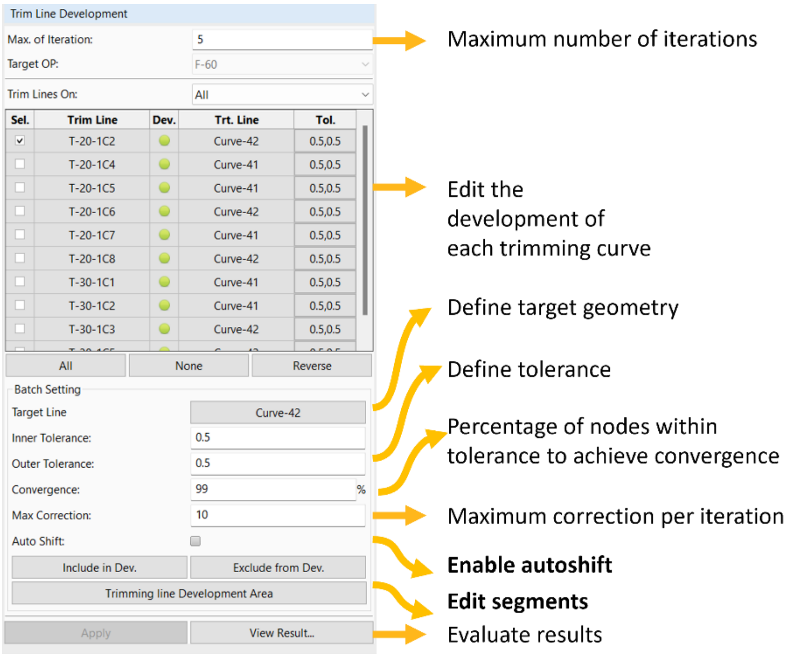


*Figure 1 Interface to set up the trim line development*

In this section, we provide an overview of the parameters available to the user to define the trimming line development. The *maximum number of iterations* refers to the maximum number of iterations allowed without convergence. The interface displays in a table all blanking and trimming curves. The corresponding parameters for the development can be edited by selecting each of the curves individually. Alternatively, several curves can be selected at once to define the trim line development parameters for all of them simultaneously. When a certain trimming curve is selected, the geometry of the curve is displayed in red and the associated target curve in a dotted blue in the main window.

Under the table that lists all the trimming curves, we can edit the settings for the development. Firstly, we select the *target line* from the curves that the user imports from the geometry editor. The *target line* should represent the desired contour of the part after the last forming operation. It is important that the target line lies on the surface of the die of the last forming operation. This facilitates the accurate calculation of the deviation between the final part contour and the target contour, which is later used for the correction of the trimming curve during the optimization process.

The *inner and outer tolerance* define the maximum permissible deviation between the target line and the part contour. Once the contour of the part lies within the tolerance range from the target line, the trim line development determines that a convergent solution is achieved and stops iterating. The *convergence* parameter determines how many points, expressed as a percentage, need to be within the tolerance range to consider that the solution has converged. For quick trim line development analyses, such as feasibility tests, the user can lower the *convergence* parameter or, alternatively, increase the tolerance range.

The parameter *maximum correction* limits how much the trim lines can be modified during the development process. When the initial guess of the trimming curve is far off from the optimal trimming curve, this parameter ensures that for each iteration, the correction does not exceed a certain value. This helps avoiding potential instabilities in the correction. The parameter *auto shift* is explained in the next section of this paper in more detail. The buttons *include* and *exclude from development* determine whether the trim curves are going to be modified during the analysis. If a certain trimming curve is included in the development, a green point is displayed in the table listing all trimming lines. On the contrary, if a trimming is excluded from the development, a red point is shown in the table.

The button *trimming line development* area leads to a new window where each of the trimming curves can be split into segments. This paper presents the segmented trim line development in more detail in a subsequent section.

Once the trim line development analysis is set up, the user can select the job runner and start the simulation. The ANSYS FORMING Job Manager prepares all the input files up to the maximum number of iterations. When a convergent solution is achieved, the Job Manager stops, informing the user that the trim line development analysis concluded successfully. At this point, the user can select the button View Results, which opens a dedicated postprocessor.


*Figure 2 Interface to view the results of the trim line development*

Figure 2 displays the result viewer for the trim line development. In the result viewer, the user can select which iteration to visualize. The table shows the list of curves included for development. For each of the curves, it displays the percentage of points within the tolerance. The green or red coloring illustrates whether the curve has converged at the selected iteration, according to the tolerance range and convergence criteria defined when setting up the simulation. In the main window, the contours of the parts corresponding to each of the developed trimming curves are shown. These points are color-coded to illustrate whether they are within tolerance (green or blue) or out of tolerance (red or brown). In the same view, the user can display the trim lines for each iteration by selecting the option *Show Trim Lines*.

## 3 Automatic shifting of trimming curves

The standard trim line development algorithm computes the difference between the contour of the part after the last forming operation and the target contour. Then, it uses this information to adjust the trimming curve so that the difference is minimized. For this standard case, the solver adjusts the shape of the trimming line to generate an optimal curve, which is successful when centroid of the initial guess of the trimming curve is placed close to the centroid of the optimal trimming curve. However, it might struggle in cases where the centroid of the initial guess is far from the centroid optimal curve. Therefore, the trimming line development module provides a new functionality to automatically move the centroid of the trimming curve to its optimal location. Once the shifting occurs, the next iterations continue to adjust the shape of the trimming curve using the conventional development approach.

Figure 3 illustrates an application example of the automatic shifting functionality. The user provides an initial guess of the trimming curve to start the trim line development analysis. This curve leads to a contour of the part depicted in red, as shown in the initial iteration of Figure 3. The centroid of the resulting contour of the part is far off from the target contour, depicted in gray. By applying the auto shift option, the algorithm proposes a new trim curve by shifting the centroid of the trimming curve in space. The updated trimming curve of Iteration 1 results in a better fit than the initial guess. However, the contour of the resulting part remains still out of tolerance. In the following iterations, the solver continues to adjust the shape of the trim curve until the contour is within the tolerance range specified by the user.

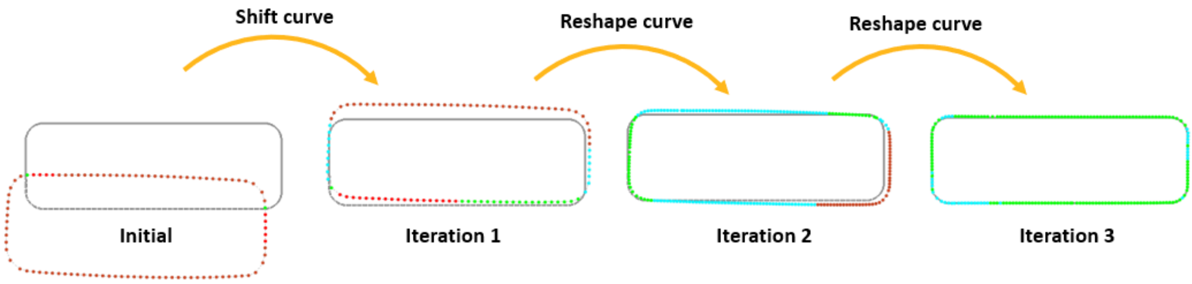


*Figure 3 Example of auto shift trim line development. The gray curves show the target contour. The colored curves represent the contour of the part after the last forming operation. Red: out of tolerance. Blue and green: within tolerance*

The automatic shifting functionality is particularly useful to adjust the position of holes created during a trimming operation. Figure 4 shows an example of a part with a trimming operation that generates 28 holes on the part requiring both a trimming and a drawing operation. These holes need to be accurately positioned and their shape needs to be adjusted with a tolerance of ±0.075 mm. By using the automatic shifting option, the first iteration of the solver generates a set of trimming curves that are positioned centered with respect to the target contour geometry. In the next iteration, the solver adjusts the shape of the trimming curves so that the part meets the geometrical requirements.

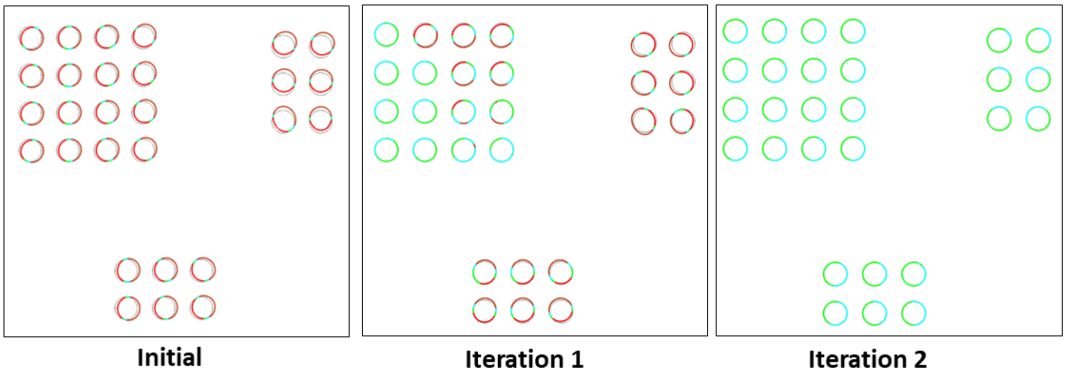


*Figure 4 Trim line development of several holes on a part with a tolerance of ±0.075 mm. Red> out of tolerance. Green and blue: within tolerance. The solution converges within 2 iterations*

## 4 Segmented trim line development

In trimming line development, the solver needs to connect univocally points of the contour of the final part to points of the target contour. In some cases, the target contour geometry matches well the contour of the part linked to the corresponding trimming operation after simulating all forming stages. This is the case depicted in the left side of Figure 5. For this case, establishing a univocal relationship between the two contours is straightforward, evaluating the error and performing the correction of the trimming curve.

In other cases, the user supplies a target curve that only partially covers part of the contour of the part associated to the trimming curve, as shown in the right side of Figure 5. The intent of the user might be to exclusively develop part of the curve, while leaving the rest of the trimming curve unaltered. However, without additional information the solver develops the full trimming curve, which might lead to undesired results.

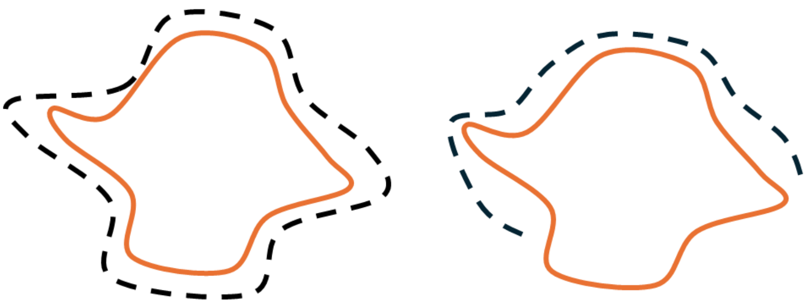

*Figure 5 Left: The contour of the part corresponding to the trimming operation (orange) wraps around the target contour (dash line). Right: The target line wraps around partially the contour of the part of the part corresponding to the trimming operation*

One solution to this problem could be dividing the curve into two sections, namely splitting the part that needs to be developed and matches the target contour from the rest of the curve. However, this approach is inconvenient for the user as it requires additional operations in the CAD editor. In addition to that, the resulting curves might not be smooth in the transition between the section that is developed and the section that remains unaltered.

To address the cases where a partial trim line development is required, a new functionality is introduced allowing the development of segments of the original trim line. The user selects which sections of the trimming curves need to be developed using a convenient interface. Figure 6 shows an example of the interface used for segmented trimming line development. The user can split the trim line by using the button *Edit Segment*. All the different segments defined by the user are displayed in a table. The user can include or exclude each of the trimming curves from the development by clicking or unclicking each of them. If the trimming curve is included, the trimming curve is displayed with a solid line in the main window. If the trimming curve is excluded, it is displayed with a dotted line.

Once the user defines all required segments, the solver develops the selected segments and connects the different parts of the curve ensuring its smoothness. The user can control the transition zone between two segments using the parameter *end smoothness*, which can range between 0 to 1. If the *end smoothness* is equal to 0, there is an abrupt transition between the developed and undeveloped segment. If the *end smoothness* is set to 1, the solver provides a longer transition zone leading to a smooth change in the transition between the developed and undeveloped segment.

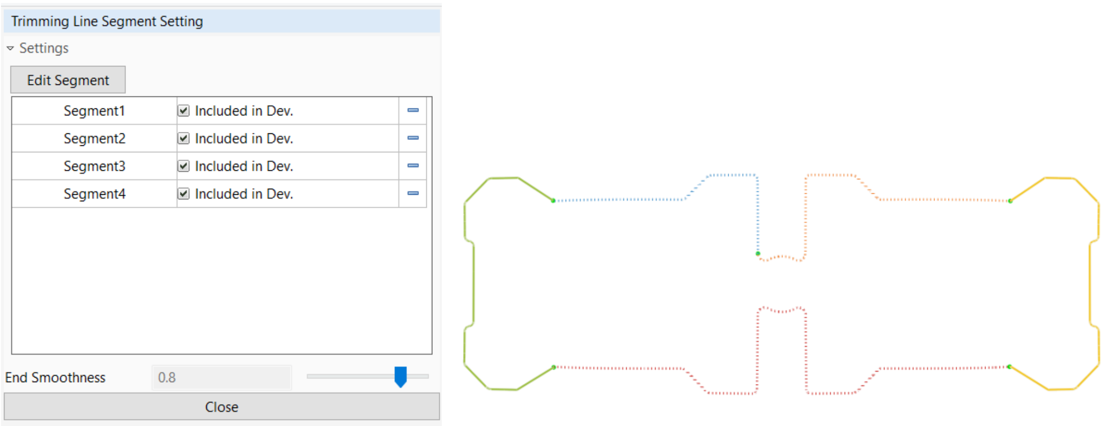


*Figure 6 Segmented trim line development. The segments of a single trimming curve are represented by different colors. The solid lines show the segments that need to be developed. The dotted lines represent the segements that need to remain unaltered*

The case study in Figure 7 showcases a practical use of the segmented trimline development. The example illustrates a multi-stage forming process for a part with an undercut. Several trimming curves define the contour of the final part, as shown in the left side of Figure 7. As the user selects to develop all trimming curves, some undesired results emerge. In the corner section, there is an overlap of several trimming curves. When the solver aims at developing all of them independently, the resulting part contacts itself during the last forming operation leading an unwanted wrinkling effect.

*Figure 7 Left: Trimming curves (red) and target contours (blue). Middle: Final part after all forming operations with the initial trimming curves. Right: Wrinkling of the material at the corner of the part after the first iteration without using segmented trim line development. Mesh and thickness contour plot*

The segmented trimming line development can be used to avoid these issues. The user can avoid the development of the segments that overlap with other trimming curves. In this case, there is a unique trimming curve for each section of the target geometry, ensuring a univocal correspondence with the target and the part contour. Figure 8 shows that the segmented trim line development succeeds in a convergent solution within one iteration for a tolerance of ±0.5 mm. Figure 8 also depicts the contour plot of the thickness of the part in the corner, leading to the desired final geometry of the part.

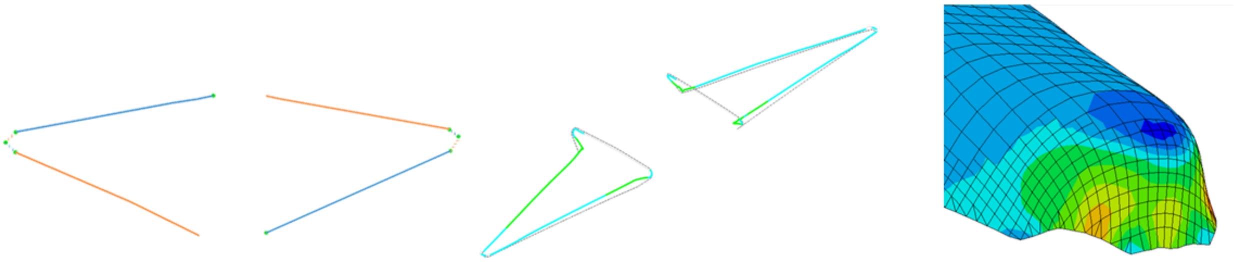

*Figure 8 Left: Definition of the segmented trim line development. Segments to be developed in solid lines and segments to remain unmodified dotted lines. Middle: Final part contour within tolerance after one iteration. Right: Detail view of the corner after trim line development. Mesh and thickness contour plot*

## 5 Overall performance improvement with additional controls

During the trim line development, the user needs to provide a geometric tolerance to assess if the developed trim line leads to a final part that matches the target contour. When the whole contour is within tolerance, the solver stops iterating. However, some cases might not require that the whole contour is within tolerance. The user might be satisfied with a certain percentage of the contour meeting the tolerance requirements to conclude the analysis. Therefore, a *convergence criterion* parameter is added, such that the iterative solution stops when the percentage of the curve indicated by the *convergence criterion* is achieved. This functionality is particularly useful for initial studies or feasibility analyses, where partial convergence suffices to achieve the desired insights on the process.

The element size along the boundaries after a trimming operation plays an important role during the trim line development. During the trim line development, the algorithm extracts the geometry of the final part from the Finite Element (FE) mesh. On one hand, the geometric features might not be accurately represented with a coarse mesh, reducing the accuracy of the resulting curves after development. On the other hand, an excessively fine mesh reduces the efficiency of the solution or might even lead to numerical instabilities of the explicit solver. The parameter *adaptive size threshold* determines the element adaptivity along the trimming curves. In previous versions, the user provides an overall value for the *adaptive size threshold* for the whole operation. As shown in the left side of Figure 9, this leads to excessive number of elements in trimming operations with curves at different scales. In the last release, the parameter *adaptive size threshold* can be modified for each of the curves. The right side of Figure 9 illustrates the resulting mesh when different values of the adaptive size thresholds are provided for each of the curves.

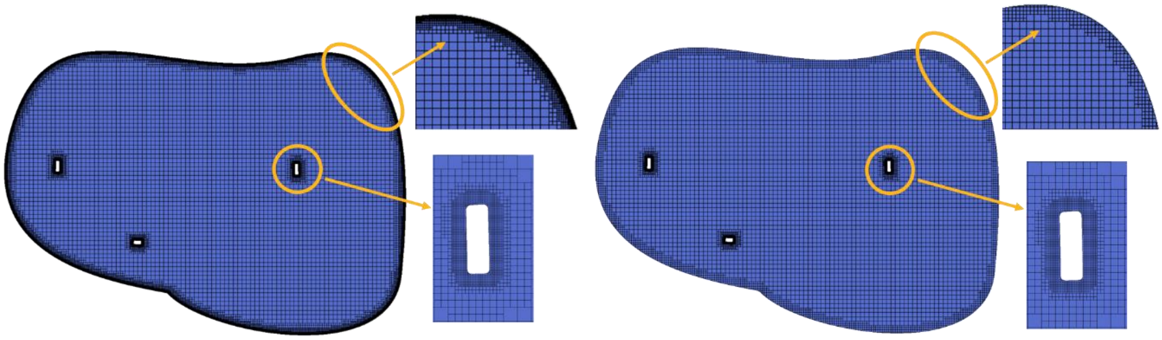

*Figure 9 Left: Resulting FE mesh after a blanking operation with an adaptive size threshold for all curves of 0.1 mm. Node number 57020. Right: Resulting FE mesh after a blanking operation with an adaptive size threshold of 0.1 mm for the inner curves and 1.0 mm for the outer curves. Node number 11424.*

The *maximum displacement* is an additional control included in trimming line development. This parameter determines the maximum correction allowed for each of the trimming curves. A larger value of the maximum displacement leads might reduce number of iterations. However, a more conservative *maximum displacement* might be advisable when the material flow has a large impact on the final geometry of the part, as a larger value might leads to oscillating, unstable results. The default value provided in the interface has proven to be a robust starting point.

## 6 Conclusion

The trimming line development is an essential tool for designing the sheet metal forming process. ANSYS FORMING provides a robust and user-friendly solution to achieve optimal trimming curves to achieve the desired final part geometry. The new functionalities and control presented in this paper unlock the solution of more complex problems, enabling the user to tackle the challenges associated to the design of multi-stage forming processes.

## 7 References

[1] P.J. Arrazola, T. Özel, D. Umbrello, M. Davies, I.S. Jawahir, Recent advances in modelling of metal machining processes, CIRP Annals, Volume 62, Issue 2, 2013, Pages 695-718, ISSN 0007-8506,
[2] https://www.ansys.com/products/structures/ansys-forming