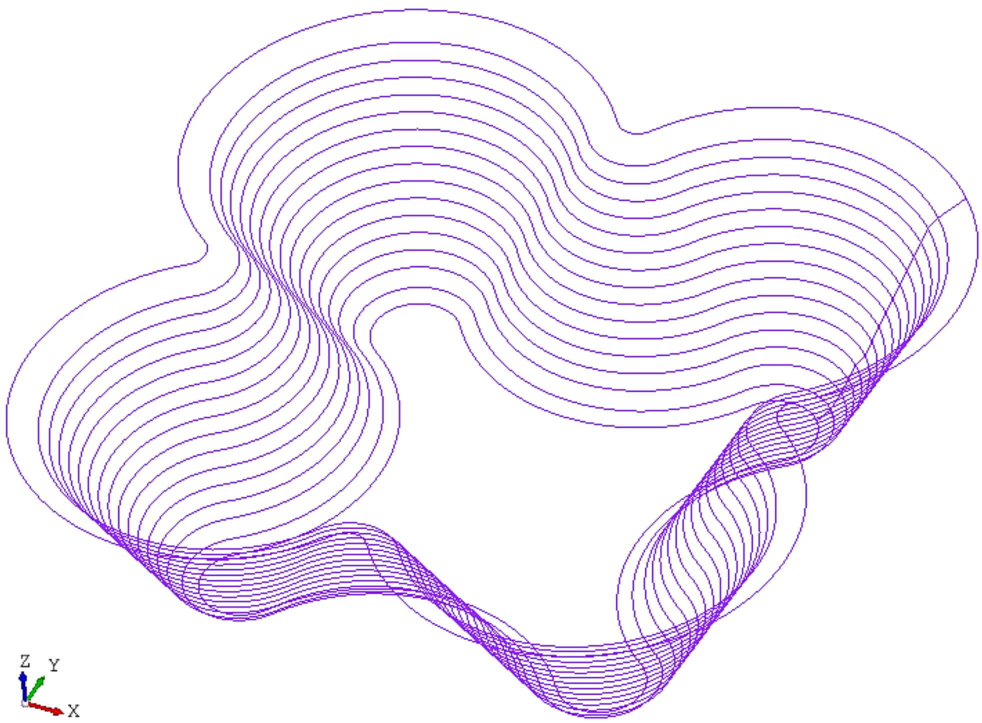
AMPL Toolpaths – User Guide



Software Creator:

Newell Moser, PhD Candidate
Advisers: Prof. Jian Cao and Prof. Kornel Ehmann
Advanced Manufacturing Processes Laboratory (AMPL)
Mechanical Engineering, Northwestern University

Contents

Disclaimer

IN NO EVENT, UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING, SHALL *AMPL TOOLPATHS*, OR ANY PERSON BE LIABLE FOR ANY LOSS, EXPENSE OR DAMAGE, OF ANY TYPE OR NATURE ARISING OUT OF THE USE OF, OR INABILITY TO USE THIS SOFTWARE OR PROGRAM, INCLUDING, BUT NOT LIMITED TO, CLAIMS, SUITS OR CAUSES OF ACTION INVOLVING ALLEGED INFRINGEMENT OF COPYRIGHTS, PATENTS, TRADEMARKS, TRADE SECRETS, OR UNFAIR COMPETITION. BY USING *AMPL TOOLPATHS*, YOU AGREE TO THESE TERMS OF USE.

Overview

AMPL Toolpaths is a minimalist software product designed for the purpose of creating and visualizing incremental sheet forming (ISF) toolpaths. ISF is a flexible process in which a sheet of metal is formed by a progression of localized deformation. It is flexible because specialized tooling is not required; a simple tool moves over the surface of the sheet such that a highly localized plastic deformation is caused. Hence a wide range of 3D shapes can be formed by moving the tool along a correctly designed path. The principle goal which motivates the development of ISF is the possibility of forming sheet metal without the need to manufacture specialized dies. This is particularly advantageous for small batch or customized production. ISF is still in active research and improved forming strategies are continuously being proposed in the field. For more information, please refer to Cao et al.[1,2], Allwood et al.[3], and van den Boogaard[4].

The generation of toolpaths in ISF presents rather unique challenges compared to conventional toolpath design, as is commonly found in CNC machining. In CNC machining, it is safe to assume that material will be removed/cut in and around the tool's current location, which in CAD terms, can be captured using a simple negative boolean operation between the tool and solid work piece. On the other hand, the sheet of metal in ISF bends, locally thins, stretches, and springs-back during the forming process which are all challenging to predict and account for during toolpath generation. *AMPL Toolpaths* uses various modern procedures, many of which are still being actively researched (e.g. AMPL website), to overcome some of the aforementioned challenges and produce reliable toolpaths specific to the ISF process.

The CAD engine in *AMPL Toolpaths* was coded on top of Open CASCADE Technology 7.1.0 (LGPL V2.1) while the front-end, or graphical user-interface, was designed using Qt 5.8.0 (LGPL V3.0). These C++ libraries were chosen in an effort to keep *AMPL Toolpaths* "free" in terms of licensing while simultaneously providing a modern, intuitive implementation of today's popular CAD technology. *AMPL Toolpaths*, however, is not a substitute for modern Computer-Aided Engineering software (e.g. Dassault Systèmes SolidWorks, Siemens NX, PTC Pro/Engineer, etc.). The design and creation of general CAD surfaces, which represent the user's desired part to be formed, must still be made using external CAD products and then imported into *AMPL Toolpaths*.

[1] N. Moser, D. Pritchet, H. Ren, K.F. Ehmann, J. Cao, An Efficient and General Finite Element Model for Double-Sided Incremental Forming, ASME. J. Manuf. Sci. Eng., Volume 138, Issue 9, 2016, Pages 091007-10

[2] R. Malhotra, J. Cao, M. Beltran, D. Xu, J. Magargee, V. Kiridena, Z.C. Xia, Accumulative-DSIF strategy for enhancing process capabilities in incremental forming, CIRP Annals - Manufacturing Technology, Volume 61, Issue 1, 2012, Pages 251-254

[3] Kathryn Jackson, Julian Allwood, The mechanics of incremental sheet forming, Journal of Materials Processing Technology, Volume 209, Issue 3, 1 February 2009, Pages 1158-1174

[4] W.C. Emmens, G. Sebastiani, A.H. van den Boogaard, The technology of Incremental Sheet Forming—A brief review of the history, Journal of Materials Processing Technology, Volume 210, Issue 8, 1 June 2010, Pages 981-997

Installation

System Requirements
OS: *Windows 7 or newer, (x64)*
Graphics: *OpenGL 3.3 or newer*
Min. Memory: *512 MB, 1 GB recommended*
Min. Free Disk Space: *256 MB, 512 MB recommended*

AMPL Toolpaths is a self-containing software package in that no independent installation process is required to run the software. For the latest version, please contact/email the Advanced Manufacturing Processes Laboratory (AMPL) at Northwestern University for a download link (AMPL website). The software package is nothing more than a compressed file containing all of the required executables and libraries to run the software; the user must simply double-click the windows batch (.bat) file to begin.

User Interface

Upon starting *AMPL Toolpaths*, a top-level menu, dockable view toolbar, graphics rendering region, and text-based dialog is presented, as illustrated by Figure 1.

Corresponding to Figure 1:

1. Fit all rendered items into the current view
2. Reset view to the original isometric view
3. Change view orientation to one of three side views or an isometric view
4. Enable dynamic panning, rotation, or zooming, which is then used by holding the left mouse button
5. Toggle the active rotation axis when rotating the view using the left/right arrow keys
6. View the active part as shaded, wireframe, or with a section/clipping view
7. Hide/show the active part, world origin, or active toolpaths
8. Information dialog to inform the user of helpful hot keys, which are the following:
    a. Hold the right-mouse button or use ctrl + arrow keys to pan
    b. Scroll the mouse wheel or use the up/down arrow keys to zoom
    c. Press and hold mouse wheel or use the left/right arrow keys to rotate
    d. Press X, Y, or Z to change the active rotation axis when rotating with the arrow keys
9. History dialog where relevant information is provided to the user
10. Status pane containing a small description of the function currently being hovered by the mouse

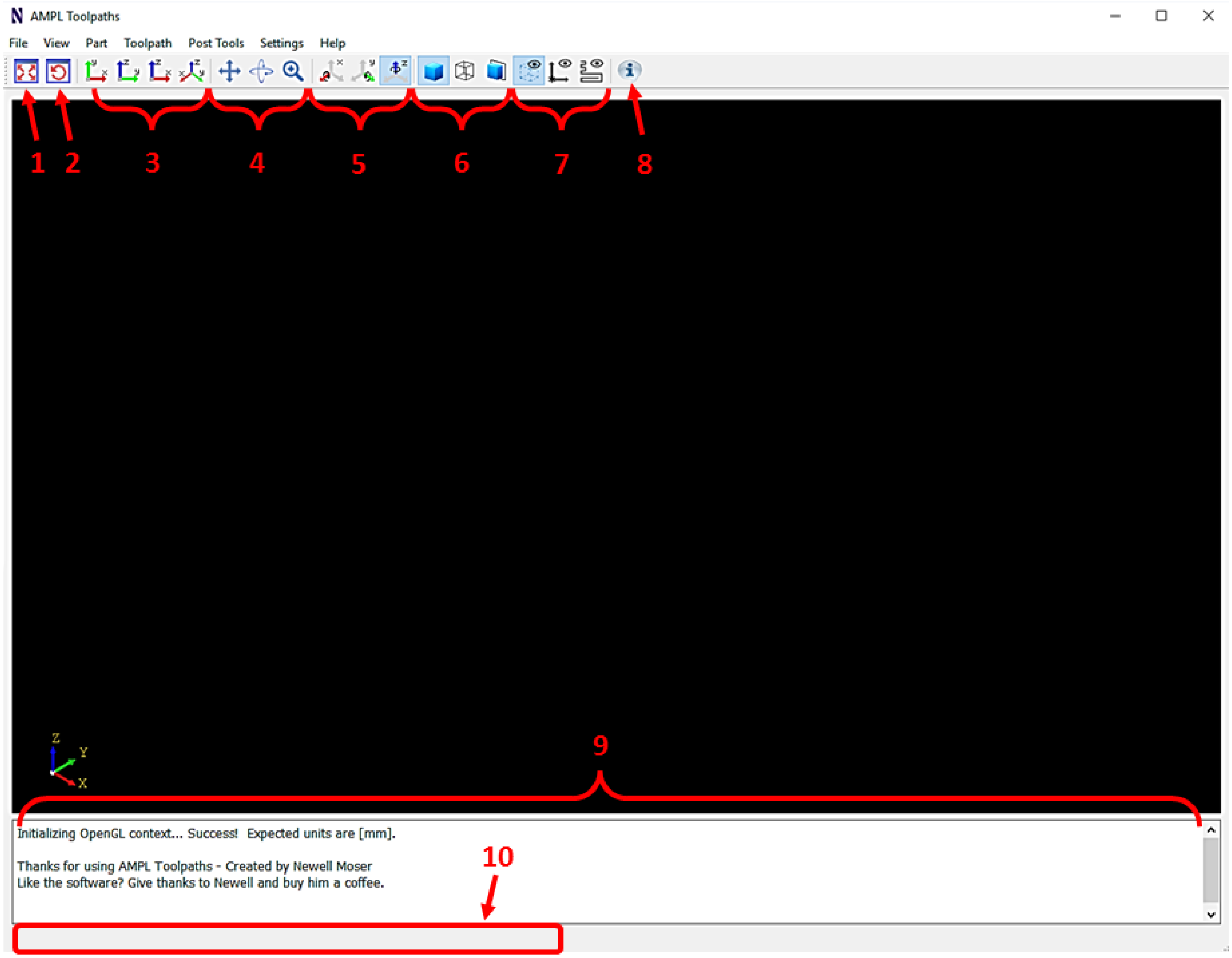


Figure 1 *Description (see below) of the various tools that are shown during the startup screen of AMPL Toolpaths. The dockable View Toolbar is shown by the buttons described in 1 – 8. The Workspace where parts and toolpaths will be visualized encompasses the large, central black box. Progress Information is provided to the user in 9 (the History Dialog), while small user-tips are provided by hovering over each button and simultaneously given by 10 (the Status Pane).*

The top-level menu categorizes various sets of features as follows:

File – Import/Export/Load/Save parts and native toolpaths
View – Top-level access to the View Toolbar
Part – Create common part shapes (i.e. cone, funnel, and pyramid). Also, the user can rotate, scale, and translate the active part.
Toolpath – Generate a toolpath, contact points or tool-tip points, for the active part
Post Tools – Functionality to help the user measure and/or post-process ISF experiments
Settings – Change some user-preferences like background color (more options to come in later versions)
Help – Provide information on the version of *AMPL Toolpaths* that's currently be used

CAD Import, Export, and Part Creation

Computer-Aided Drafting (CAD) surfaces can be imported/exported into/from *AMPL Toolpaths*. Initial Graphics Exchange Specification (IGES) files are supported, but this is not advised since the last published standard was in 1996. Since IGES is no longer standardized, imported parts may contain continuity errors and bad surface definitions due to mistranslations from other CAD packages. For these reasons, it is suggested that STEP files are solely used, specifically of format STEP AP214 (ISO published in December 2014) or earlier. Commonly used parts in ISF can be directly constructed in *AMPL Toolpaths* under the top-level menu, Part -> ISF Part Library.

As shown in Figure 2, the Import Dialog asks the user to select a Sewing Tolerance as well as an option to perform a Shape Healing operation. *AMPL Toolpaths* imports only surfaces and faces (solid parts will be converted into shells), where the user must select a Sewing Tolerance to help the software determine if adjacent surfaces/faces should be sewn together as one continuous surface. In general, Shape Healing should be *avoided* unless needed, which is more often the case for IGES parts. If the imported surfaces appear to have holes or there exist very small edges, then Shape Healing will attempt to automatically fix these issues. To determine if the part was imported correctly, the user is recommended to visualize the part in Wireframe mode. When the part is viewed in Wireframe mode, free edges will be visualized by green lines, joined edges will be visualized in yellow, and gray edges represent support/construction lines the surfaces.

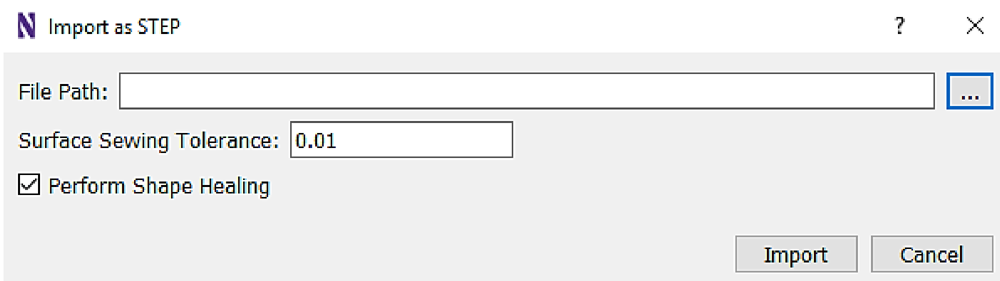


Figure 2 *The Import STEP dialog. Only the faces of a STEP or IGES part are imported by AMPL Toolpaths*

Saving, Loading, and Exporting Toolpath Files

Due to the potential requirement of storing millions of [X, Y, Z] points, *AMPL Toolpaths* uses a native AMPL file format (.ampl file extension) which is in binary in order to speed up the processing time of loading a toolpath as well as to minimize a toolpath's storage size. As will be discussed shortly, the calculation of the toolpath's contact points is a computationally expensive operation, and to help minimize performing this operation numerous times for the same part, AMPL files separate tool tip points and contact points. Because of this separation of toolpath quantities, the user can load a prior toolpath containing contact points and change the tool type/size without the need to recalculate the contact points. Furthermore, a given toolpath's information (contact points, tool tip points, geometric properties, forming setup, etc.) can be converted into a human-readable form via text files by selecting File -> Export Toolpath to CSV. This feature is particularly useful if the user desires to implement a new, custom toolpath technique that is not currently supported by *AMPL Toolpaths* such as machine compliance compensation during the calculation of tool tip points.

Creating Contact Points

The calculation of contact points for an ISF toolpath is arguably the most critical operation during toolpath generation. Upon clicking Toolpath -> Generate Contact Points -> Single Feature Part, a comprehensive dialog (Figure 3) will be presented to the user. While this guide is written to help a user understand what each of these individual parameters are responsible for, most of this information can also be found after clicking the provided Help/Information buttons.

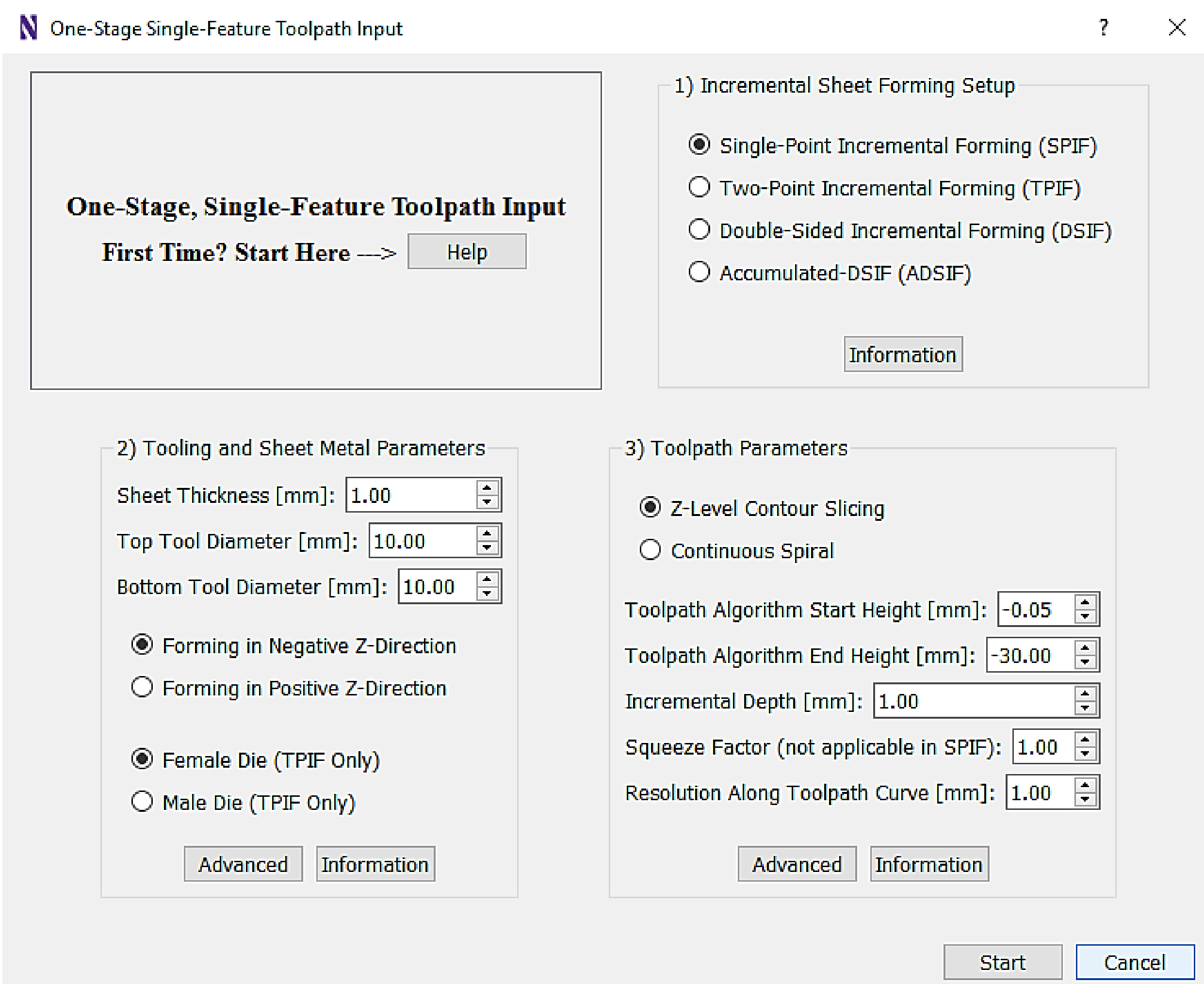


Figure 3 *Dialog used to define the necessary parameters in calculating the toolpath's desired contact points*

What is One-Stage Forming?

As the dialog's title suggests in Figure 3, the toolpath generation algorithms are limited to one-stage, single-feature parts. *One-stage forming implies that the forming tool[s] traverse in a given Z-direction just once during the forming process.* Conversely, a multi-stage forming operation corresponds to the tools forming multiple intermediate parts by reforming each previous shape, which is often required to achieve relatively high wall angles without premature fracture.

If multi-stage toolpaths are desired, then the user must manually design the intermediate shapes and run the contact point algorithm successively for each shape. Furthermore, it is sometimes desired to run the tool[s] In-to-Out during a given stage of forming rather than the default direction of Out-to-In. This can be achieved by running the standard toolpath algorithm and then exporting the contact points to a .csv file. The user can then reverse the order of the generated contact points and calculate the tool-tip points using custom external codes and scripts.

What is a Feature?

A single-feature part is defined as surface which, when successfully intersected by any plane normal to the Z-axis, contains just one, closed contour. The resultant contour need not be convex, but it must not be complex. In other words, a straight line drawn through the contour can intersect more than twice, but the contour cannot self-intersect. If the current contour, which is lying in a translated XY-plane, is not closed, then the part does not have a closed-shape and is ill-defined. *AMPL Toolpaths* will attempt to fix open contours by drawing a straight line between the end points of the contour to automatically close it.

When multiple closed contours occur upon intersecting the part with a plane parallel to the XY-plane, then the part is said to contain multiple features and should not be formed in one operation (Figure 4). If the user desires to form multi-featured parts, then each of these features must be separated from the original part and individually submitted to *AMPL Toolpaths*. Upon completion, the user can then stitch the resultant toolpaths together using whatever order that is desired. If multiple closed contours occur during toolpath generation, *AMPL Toolpaths* will attempt to combine them into one operation, which will likely result in undesirable/unexpected behavior.

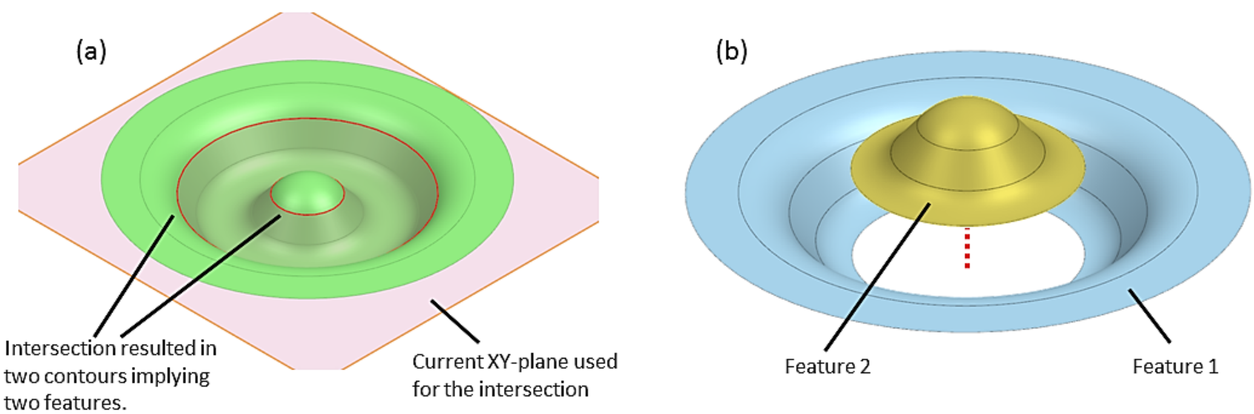


Figure 4 *(a) Example of a revolved part resulting in multiple features. (b) To form this part, the user must separate the two features manually and run the toolpath algorithms for each feature.*

Surface Definitions and Expected Input

The thin surface used to calculate the contact points can represent something entirely different depending on the ISF setup. In Single-Point Incremental Forming (SPIF), Double-Sided Incremental Forming (DSIF), and Accumulated-DSIF (ADSIF), the inputted thin surface is assumed to represent the top (i.e. positive Z-direction) surface of the desired part, which of course has both a top and bottom surface because real parts have thickness. For Two-Point Incremental Forming (TPIF), the inputted surface corresponds to the die's forming surface. To help justify why the input surface is always taken to be the top surface of the desired part in all but TPIF, refer to Figure 5 for an example of a multi-feature SPIF operation.

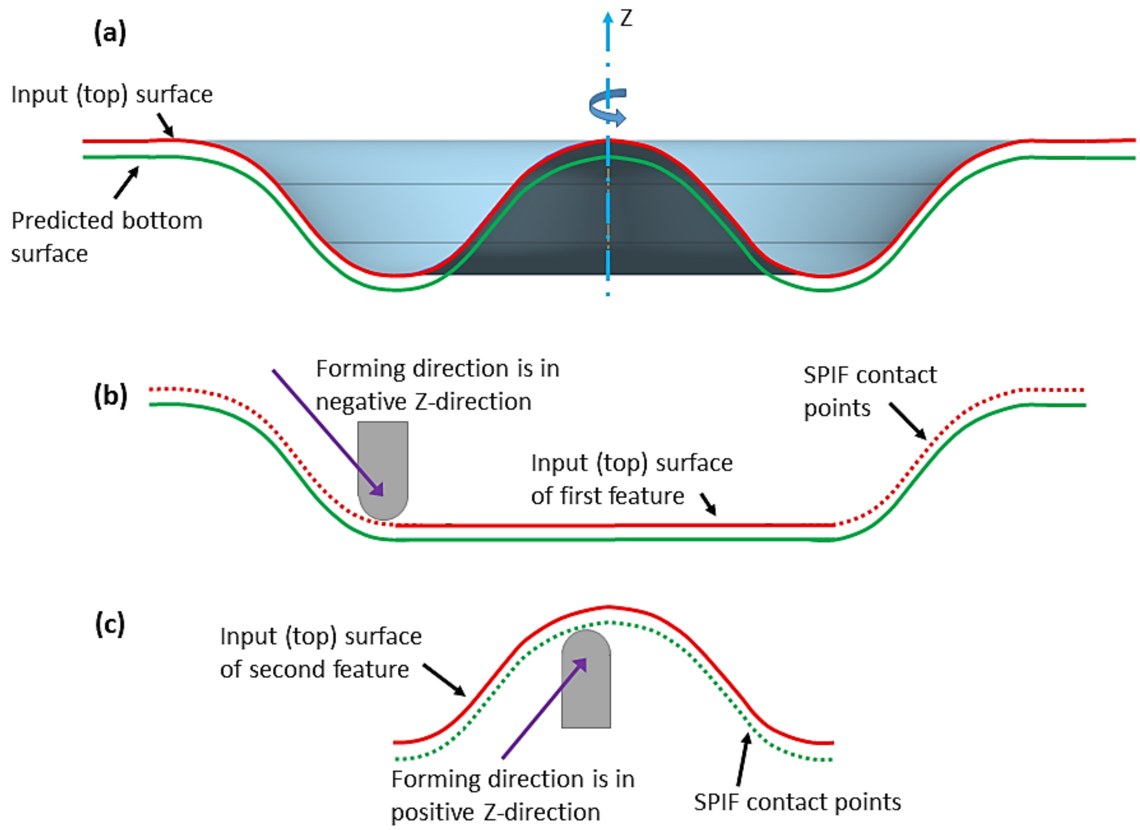


Figure 5 *(a) Example of a desired part with two features (see Figure 4) which will be formed using SPIF. In order to switch between forming in the positive and negative Z-directions, all of the features of the desired part must be defined on the same surface. To illustrate, (b) the outside feature can be formed first where the input surface and tool contact points are clearly coincident. Then, (c) the inner feature can be formed where the tool contact points will be automatically translated away from the input surface by the predicted thickness.*

To form the two-featured part in Figure 5a via SPIF, the outer feature could be formed first (Figure 5b) followed by an independent forming operation of the inner feature (Figure 5c). The contact points where the tool is expected to touch the sheet are marked by dotted lines. While forming in the negative Z-direction for the outer feature, the contact points and input surface are coincident, as expected. However, the contact points for the inner feature must then be offset by the predicted thickness in order to ensure continuity of the top and bottom surfaces of the resultant part. If *AMPL Toolpaths* did always place the contact points on the input surface, then it would be the user's responsibility to offset the input surface by the expected thickness for certain forming operations; an unnecessary chore.

To help ease the user's burden and maintain improved consistency, *AMPL Toolpaths* takes the input surface for SPIF, DSIF, and ADSIF to be the final part's top surface and automatically takes care of the necessary thickness offsets using the Sine Law (Equation 1),

$$t_f = (SF)t_0 \sin(90° - \alpha) \quad (1)$$

where t_f is the predicted sheet thickness, t_0 is the initial sheet thickness, α is the current wall angle (e.g. a flat sheet has a wall angle of 0° while a vertical wall has a wall angle of 90°), and SF is the squeeze factor (meant predominantly for TPIF, DSIF, and ADSIF) which allows the user to linearly change the predicted thickness and thus the tool gap. For example, $SF = 0.9$ will result in 10% additional squeeze relative to the predicted thickness. If an altogether different thickness prediction is desired, then the user can output the contact points into text files and perform the necessary calculations using external codes.

In TPIF, the expected input cannot be chosen arbitrarily as with SPIF, DSIF, or ADSIF since the partial die acts like a constraint for the toolpath. When a die is manufactured, it obviously cannot be easily changed and so the toolpath must deform the sheet against the die all the while maintaining the expected tool gap so as to prevent excessive squeezing and/or tool failure. For these reasons, the (female or male) die's forming surface is expected as input for all TPIF toolpath algorithms (Figure 6). *AMPL Toolpaths* will automatically offset the tool's contact points by the predicted sheet thickness for TPIF. Therefore, it is indeed the responsibility of the die maker to account for the thickness of the sheet while designing the die in order to improve the final part's geometric accuracy as well as to potentially prevent an unexpected interference between the desired forming tool and the die.

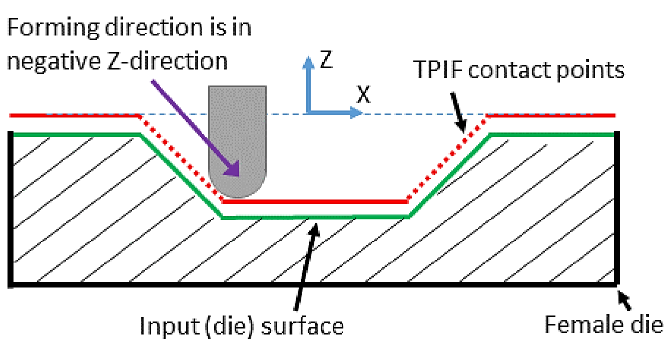


Figure 6 *Example of the expected input surface in green for TPIF (i.e. the die's surface) for a truncated cone part.*

Explanation of Selected Toolpath Input Parameters

As given by Figure 3, there are numerous features and parameters that must be defined in order to submit a toolpath job to *AMPL Toolpaths*. While most of these parameters will be mentioned to some level or another in this guide, it is the author's belief that they are explained in adequate detail within the dialog's "Information" buttons. The focus of this section will therefore be primarily for features that are not covered clearly in the program, such as the "Advanced" features.

The input parameters are separated into three general categories: 1) ISF Setup, 2) Tooling and Sheet Metal Parameters, and 3) Toolpath Parameters. Little need be said for *ISF Setup* as it should be rather apparent, especially after investigating the "Information" button and through further reading of the previously suggested journal articles.

Moving right along, *Tooling and Sheet Metal Parameters* is where the user can provide more specific information on the experimental setup. That said, all of this information will be internally saved but not necessarily used during the generation of contact points. For example, the tool diameter is not needed in generating contact points, but will be used later (or even changed if desired) for the generation of tool-tip points. Furthermore, only one of the tool diameter inputs is actually required in TPIF and SPIF, but the user can fill in both entries for the sake of simplicity. It should be said here, however, that the forming direction, either positive or negative Z-direction, is not well defined for ADSIF since this type of toolpath moves within the same XY-plane. In this case, the forming direction corresponds to the direction that the material is moving rather than the tools themselves; if the material displaces in the negative Z-direction, then the forming direction is said to be negative as well.

The final category, *Toolpath Parameters*, is where the user can control or activate various algorithms during the generation of contact points. The first decision is to decide if the toolpath should be successive contours or a continuous spiral. Though, it is advised that the user always try the Z-level Contour Slicing algorithm first to determine if there are any undesirable results in the resultant toolpath since contour slicing is fast and every other toolpath algorithm uses contour slicing as a foundation operation. By taking this advice, it will quickly become clear on whether or not the input surface is not defined well, more points are needed to capture the in-plane curvature of certain regions, or the starting location between successive contours is not continuous, etc.

While the software could automatically determine the Z-height maxima, the software asks the user to provide this input for further control. It is often desired that the user wishes to test the toolpath algorithm for specific regions of the part which could not be easily separated if the software always generated contact points for the whole part. However, here is a quick tip for those that do not know the maxima of the current part; simply go to the top-level menu and click Post Tools -> Extract Cross Section. This feature will allow you to extract position information along any cross section of the current part and save it into a text file, which then can be investigated to readily determine the highest/lowest Z-heights of the part.

Advanced Toolpath Parameters

Under the "Advanced" dialog, shown in Figure 7, the user is given further control on the generation of contact points, such as the in-plane direction of the toolpath: clockwise or counter-clockwise with respect to the Z-direction of forming. A rather important pair of parameters is the definition of the *Starting Construction Point* in polar coordinates. This construction point is used to determine where to start/end the toolpath for each given contour by selecting the discretized point on the contour that is located closest to this defined *Starting Construction Point* (Figure 8).

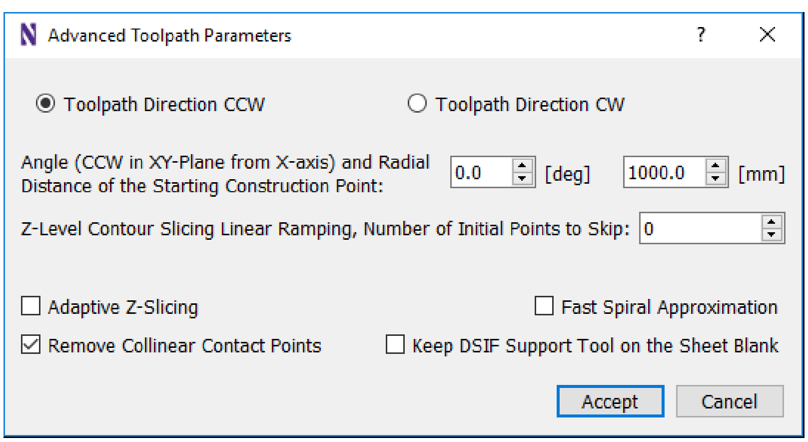


Figure 7 *Various features made available in the Advanced Toolpath Parameters dialog*

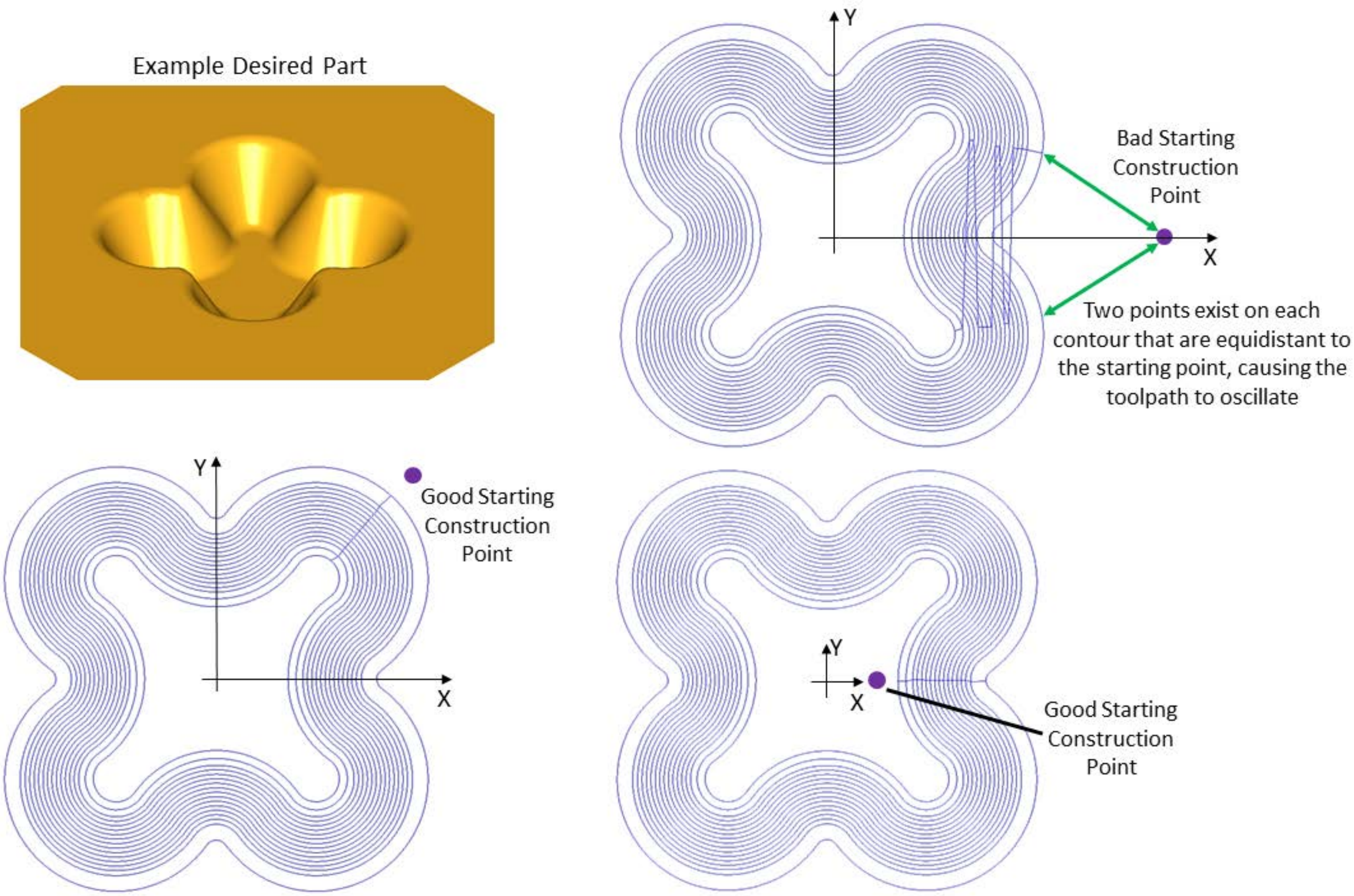


***Figure 8** The starting construction point (purple dot) is used to determine where the toolpath starts/ends for each successive contour. Bad starting points can result in unexpected toolpath behavior, especially in spiral formulations.*

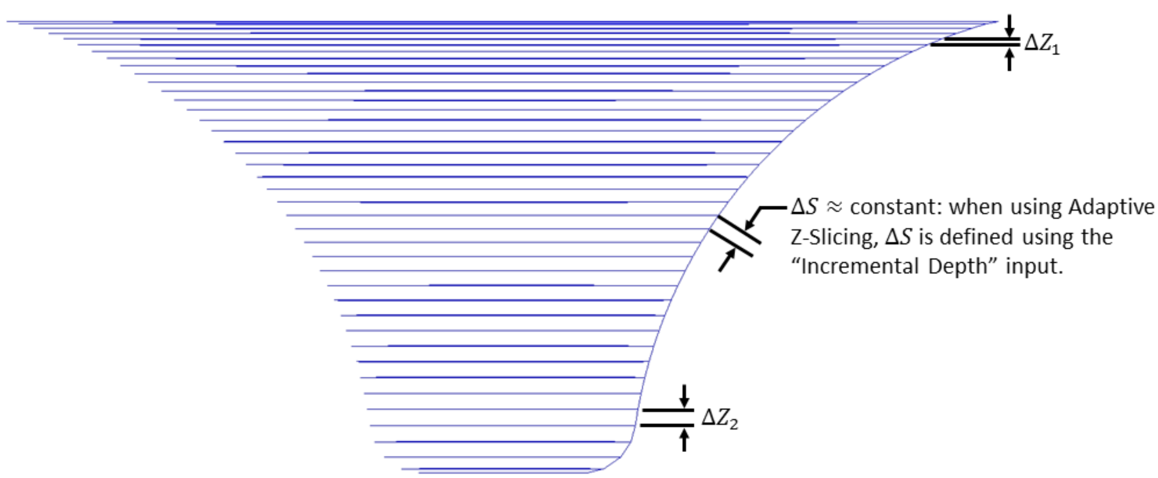


***Figure 9** Adaptive Z-slicing varies the ΔZ-increment based on the minimum wall angle for the current contour*

Another feature is Adaptive Z-Slicing, which varies the incremental depth as a function of the current wall angle (Figure 9). When Adaptive Z-Slicing is turned on, the value given by the "Incremental Depth" input under *Toolpath Parameters* (see Figure 3) will be used to define the preferred increment in arc-length, ΔS, as illustrated by Figure 9. In general, the current part will not be axisymmetric in which case the wall angle will presumably vary along any given contour. If the wall angle changes along the current contour and Adaptive Z-Slicing is activated, then the next Z-increment will be limited by the minimum wall angle of the current contour. When the minimum wall angle is used to control the ΔZ-increment, then $\Delta S \leq constant$ where again, the *constant* is defined by the "Incremental Depth" input. To add, Adaptive Z-Slicing is available for both Z-level Contour Slicing and Continuous Spiral algorithms.

The Fast Spiral Approximation is a technique that attempts to construct a spiral toolpath by using a projection "short-cut" compared to the regular spiral algorithm. This feature is of course ignored if the user decides to select Z-level Contour Slicing. The increase in speed for the Fast Spiral Approximation is significant but at the cost of occasionally behaving poorly in regions where the wall angle changes relatively quickly, such as near fillets (Figure 10). Even if the contact points look fine, the user must also calculate the tool tip points to ensure that it was safe to use the Fast Spiral Approximation because this algorithm could unexpectedly affect either the contact points or the calculation of the normal vectors. For long toolpaths, it may still be advantageous to use the approximated spiral algorithm to determine if the toolpath parameters seem appropriate before committing the computational resources required of the regular spiral algorithm.

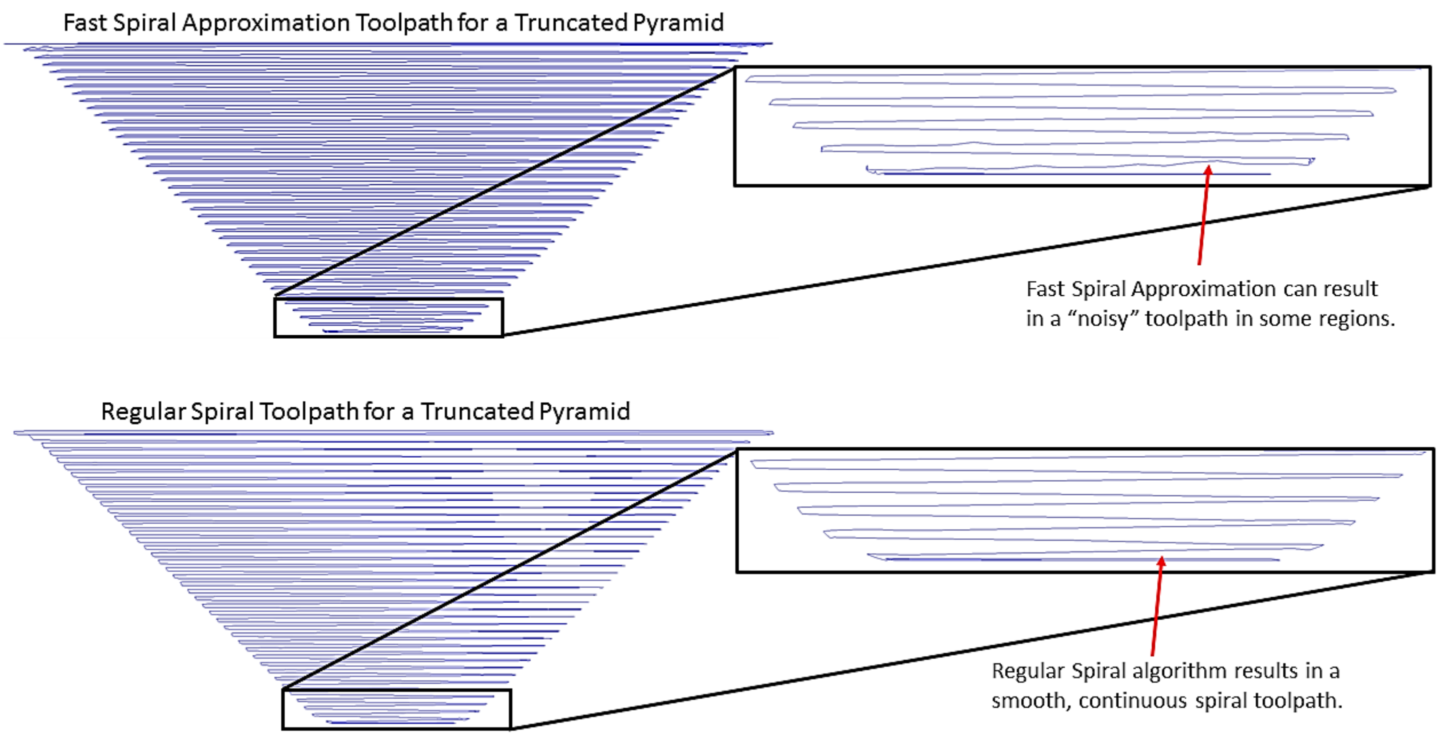


Figure 10 *Fast Spiral Approximation vs Regular Spiral toolpath for a truncated pyramid. The approximated spiral cannot guarantee that the contact points lie on the input surface due to a projection operation, which can also dramatically affect the calculation of the local normal vector and subsequently the calculation of the tool-tip points.*

Remove Collinear Contact Points is a straightforward toolpath feature that deletes select points that happen to be collinear with neighboring contact points (Figure 11). Removing these linearly dependent points will help to shrink the size of the toolpath for many types of parts. However, some features in *AMPL Toolpaths* work better if all of the contact points are kept, such as the self-interference calculation that can be performed during the calculation of the tool-tip points.

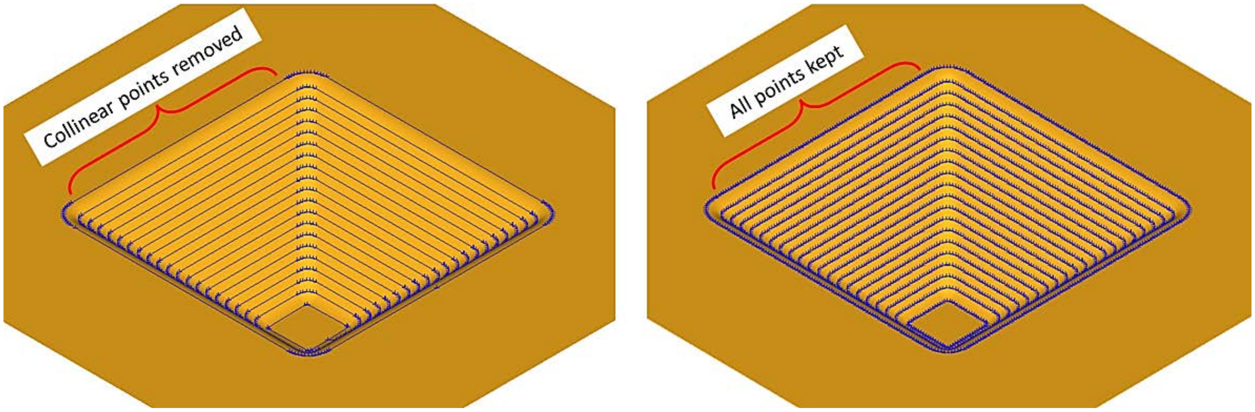


Figure 11 *If multiple contact points lie on the same line, Remove Collinear Contact Points deletes the middle points. On some parts, this can significantly reduce the file/export size of the toolpath, though the self-interference calculation performed during the generation of tool-tip points may not be as accurate.*

For some shapes, spiral projections are difficult to perform and may result in some undesirable effects. In general, spiral toolpaths are often preferred over Z-level contour slicing toolpaths due to the improved surface finish. However, *AMPL Toolpaths* incorporates an ad-hoc ramping feature for traditional Z-level contour slicing which helps to improve surface finish. When activated, a specified number of points are “skipped” at the beginning of successive contours (Figure 12).

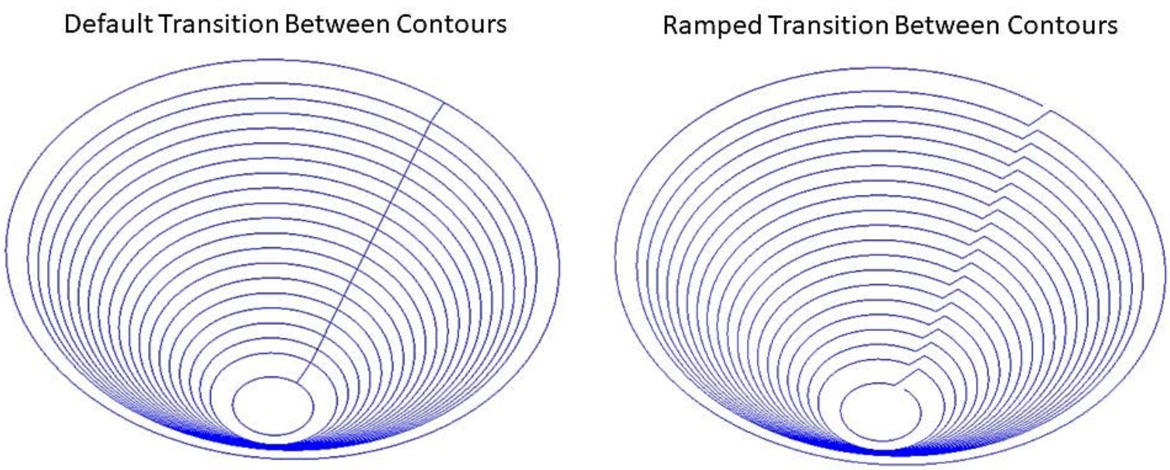


Figure 12 *Ramping between Z-Level Contours can improve surface finish as typically seen in spiral toolpaths*

The final feature to discuss is Keep DSIF Support Tool on the Sheet Blank, which as the name suggests, only applies DSIF. When activated, the supporting tool in DSIF will traverse along the first contour as the forming tool continues to form. The bottom tool is the supporting tool when forming in the negative Z-direction while the top tool is the supporting tool when forming in the positive Z-direction. The algorithm determines the correct placement of the supporting tool by using the same completion percentage of current contour. For example, if the forming tool is 65% around the current contour, then the supporting tool will also be placed 65% around the first contour. This algorithm performs best if the contours are discretized at regular intervals and thus, all contact points will be kept even if the user has chosen to activate Remove Collinear Contact Points.

Calculating Tool-Tip Points

Upon the successful generation of the contact points, the next step is to calculate the tool-tip points via Toolpath -> Calculate Tool Tip Points -> Single Feature Part. The new dialog window will then open and inform the user of the current ISF setup as well as tooling, which can be changed if desired (Figure 12). As of now, *AMPL Toolpaths* supports two tooling geometries, as shown in Figure 13. The tooling can be changed at any time because tool-tip points are dependent on the contact points and surface normal vectors, which are stored internally during the generation of contact points. Depending on the tooling geometry and/or ISF setup, some of the tool-tip input parameters may be ignored. To help summarize what types of contact points and tool-tip points will be outputted by *AMPL Toolpaths* for various ISF setups, refer to Figure 14.

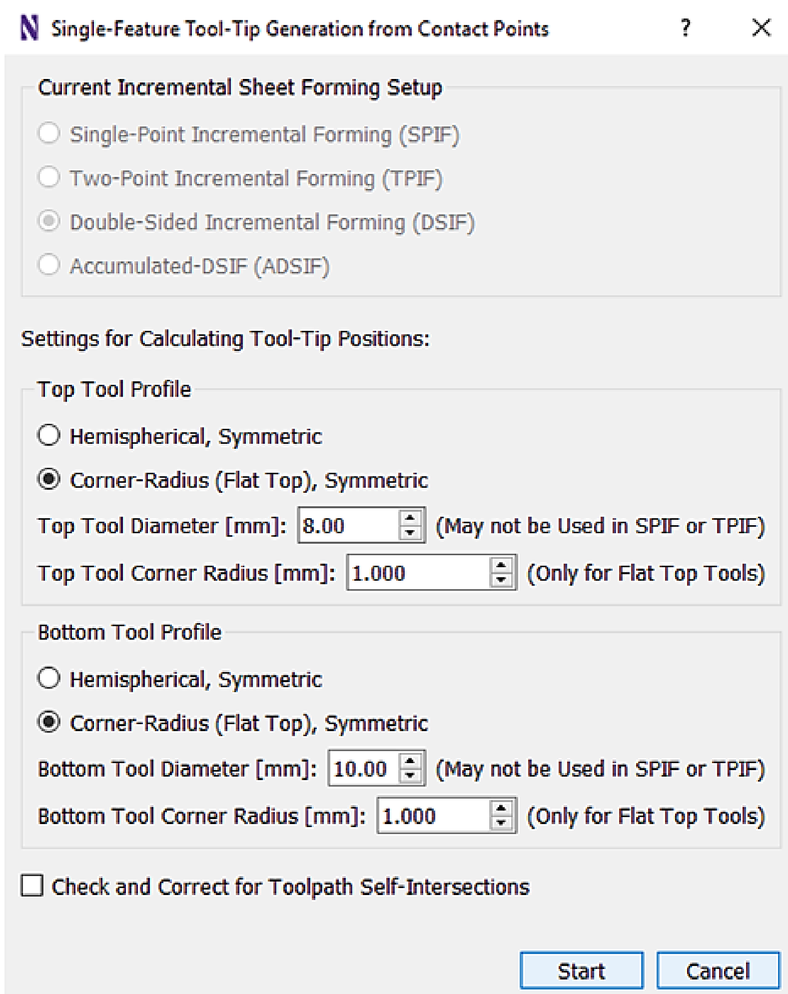


Figure 13 *Dialog window used to collect user-input and calculate the tool-tip points*

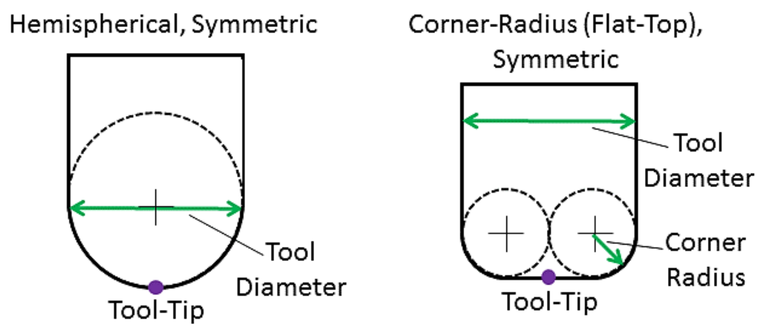


Figure 14 Section views comparing a Hemispherical tool to that of a Corner-Radius (Flat-Top) tool.

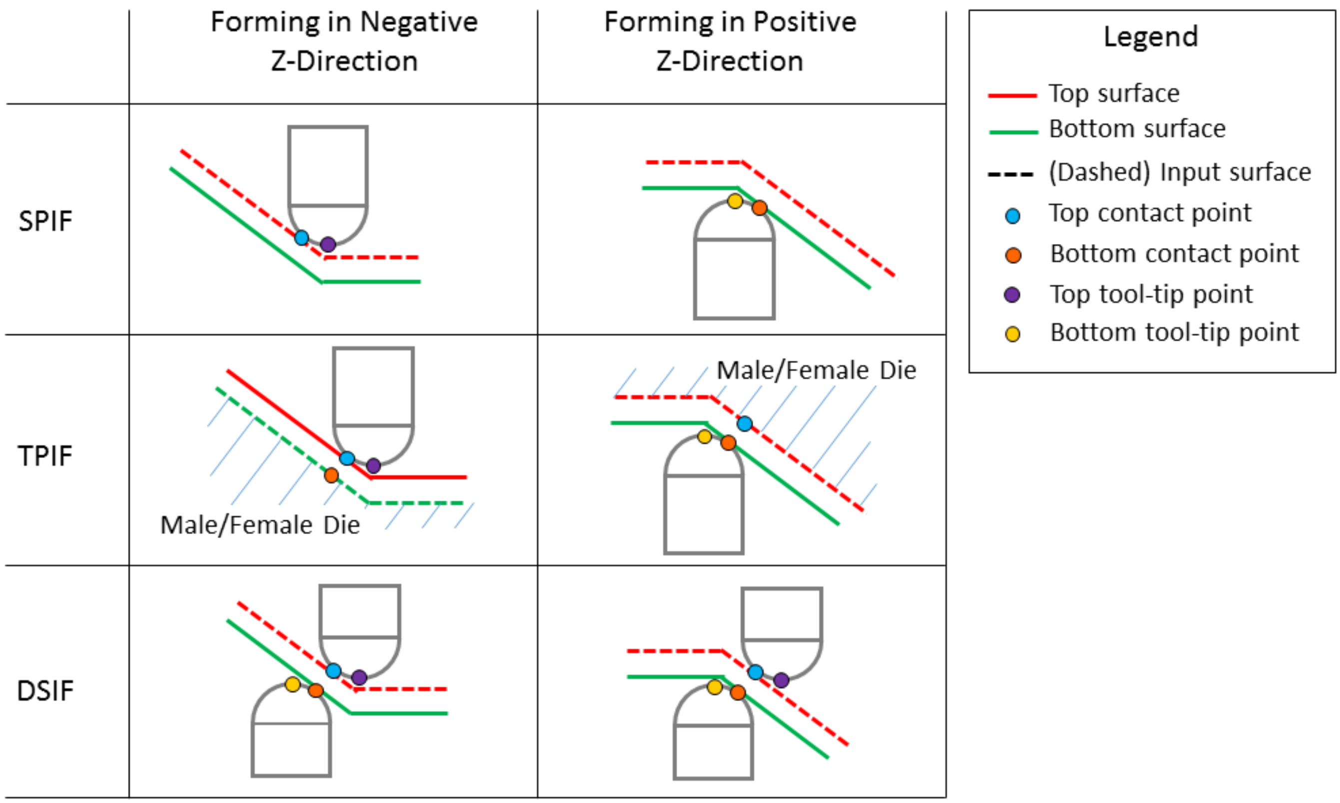


Figure 15 *Definitions of the input surface, contact points, and tool-tip points for each incremental sheet forming setup which will be computed by AMPL Toolpaths. Note that ADSIF is a variant of traditional DSIF: the contact points are simply reversed in ADSIF so that the toolpath can go In-to-Out rather than Out-to-In. To add, the tool-tip points are also projected towards the sheet blank in ADSIF.*

Check and Correct for Toolpath Self-Intersections

An additional feature that's available during the calculation of the tool-tip points is the ability to check if the resultant toolpath self-intersects with existing contact points. To be clear, this feature does not check whether the tool intersects with the input surface since this would be computationally expensive. Instead, the existing tool-tip positions, one for each contact point, are checked to ensure that only one contact point is actually touching the tool's analytical surface. If more than one contact point exists within the tool's boundary, then an interference is said to exist. A common example of an interference issue is given by Figure 15 where the tool's radius is larger than the local radius of curvature of a filleted region due to the thickness offset required in calculating the contact points.

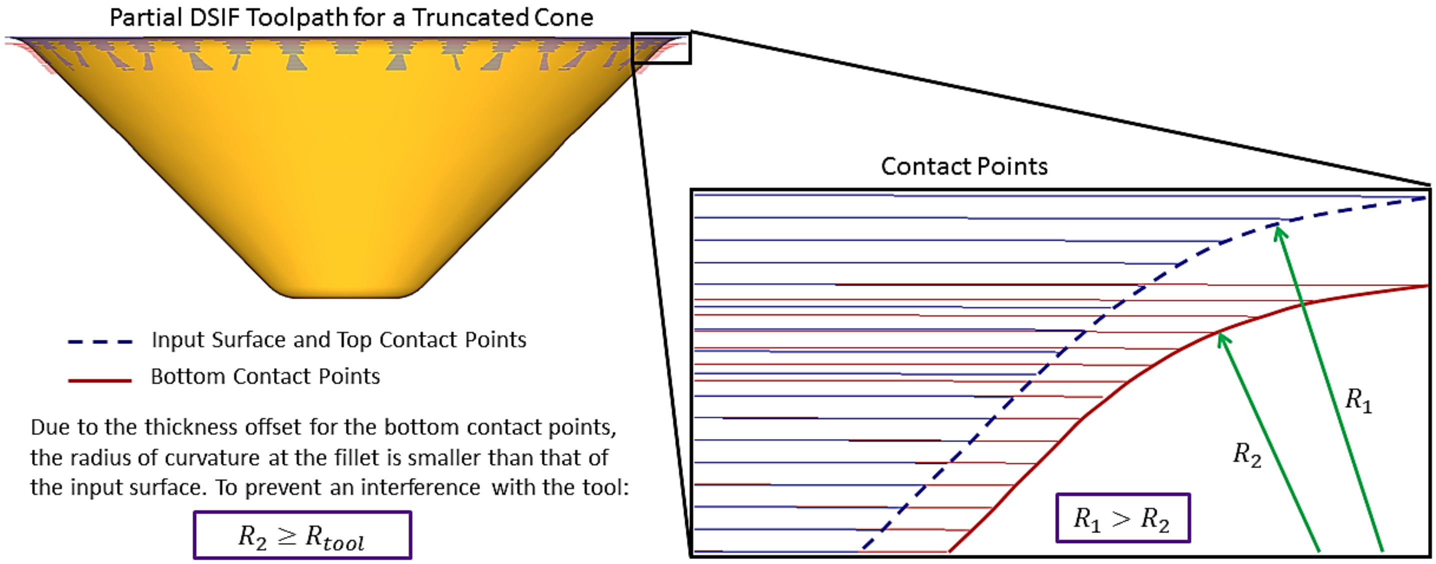


Figure 16 *Even if the radius of curvature is large enough to account for the desired tool diameter on the input surface, there still can be an interference issue due to the thickness offset often needed to calculate some contact points.*

When an interference is found, the largest interference is found and then the tool's position is translated in the direction of the local surface normal vector until there is no longer an interference. However, *this does not guarantee that the final toolpath will not interfere with the part* since only the contact points are used in the calculations. Sometimes the interference algorithm can be improved in accuracy if more contact points exist, such as is the case when retaining all of the collinear points. The key feature of this self-intersection algorithm is not actually the correction method since there are, in general, multiple solutions that exist which will eliminate the interference, and the surface normal direction may not always be the desirable/best solution. Rather, this feature can warn the user ahead of time if an interference is likely to occur for a given toolpath which is valuable information alone. Instead of trying to modify the toolpath to avoid an interference, it is advised that the user change the tooling size and/or the desired part geometry to eliminate the interference.

Convert Toolpath to Machine Code

Exporting Directly to NU's Gen. 1 DSIF Machine

AMPL Toolpaths has the capability of exporting the current tool-tip points into machine code for NU's Generation 1 DSIF Machine (Delta Tau Turbo-PMAC controller). The user can generate this type of machine code by clicking Toolpath -> Convert to Machine Code -> NU Gen. 1 DSIF Machine.

A couple of options are available when exporting to NU Gen. 1 DSIF Machine code, such as the Maximum Number of Lines per File and Desired Feed Rate. Due to the Delta Tau controller's memory capacity, it is advised that the user choose not more than 4000 lines per file. When there are more than 4000 tool-tip points in the toolpath, *AMPL Toolpaths* will split the toolpath up into multiple machine code files. Furthermore, the Delta Tau control algorithms have been tuned for a feed rate of 5 mm/sec. Unless the user has explicit experience with Delta Tau hardware and software setup, tool speeds should not be in excess of 6 mm/sec.

Post-Processing Tools

There are common operations that arise during post-processing of ISF geometry data which are available in *AMPL Toolpaths*. For one, there is often a need to compare a cross section of the desired geometry to that of the experimentally formed part. This type of information is easily extracted using Post Tools -> Extract Cross Section. The dialog window will then ask the user to define the intersecting plane, the arc-length distance between subsequent points, as well as the starting construction point which is used to order the discretized points. For more information on defining a good starting construction point, refer Figure 8. The resultant text file will contain the X-, Y-, Z-position of the generated points as well as local geometric properties including the normal vectors.

The 3D deviation map between two surfaces is also of great interest when calculating the thickness of a measured part, or when determining the 3D geometric error between the measured and desired parts. This tool can be initiated by going to Post Tools -> Two Surface Deviation Map, which then brings forward a rather comprehensive dialog menu that helps the user step through the process of defining the required input parameters. In short, two surfaces will be imported and stored in memory: Surface 1 and Surface 2. Additionally, a 2D region of interest must be defined in the XY-plane, which currently can be either a rectangle or a line. The algorithm will then proceed by projecting the XY-query points along the Z-axis towards Surface 1, resulting in a new point on Surface 1 with an associated local normal vector. Then, a new line is constructed along the direction of the local normal vector which is intersected with Surface 2, resulting in a new point on Surface 2. The distance between the Surface 1 point and Surface 2 point is the local deviation distance. To visualize the deviation distances of all of the points in a surface map, the user will have to use an external software package such as Matlab, GNU Octave, or Gnuplot.

Acknowledgements

AMPL Toolpaths is coded upon Open Source libraries:

1. Open CASCADE Technology 7.1.0 (GNU Lesser General Public License, LGPL V2.1)
2. Qt 5.8.0 (GNU Lesser General Public License, LGPL V3.0)
3. Cereal 1.2.2 (BSD License)

Thanks is gratefully given for friends, family, and advisers for all of their support.