RT E JOURNAL

A new slicing workflow to effectively bridge design to process for components from Directed Energy Deposition with arc

Thomas Piendl [a, b], Alexander Bonke [b], Mathias Hartmann [a]

[a] **Technische Hochschule Deggendorf** Dieter-Görlitz-Platz 1, 94469 Deggendorf, Germany

[b] **FIT Additive Manufacturing Group** Am Grohberg 1, 92331 Lupburg, Germany

https://doi.org/10.58134/fh-aachen-rte_2026_001

Zusammenfassung Directed Energy Deposition mit Draht (DED-arc) ist eine vielversprechende Technologie zur effizienten und kosteneffektiven Herstellung metallischer Bauteile mittels additiver Fertigung, bei der der Drahtwerkstoff durch einen elektrischen Lichtbogen aufgeschmolzen wird. Im Vergleich zu alternativen Verfahren zeichnet sich DED-arc durch höhere Materialauftragungsraten bei reduzierten Kosten aus. Ein zentraler Aspekt ist der Slicing-Prozess, bei dem ein dreidimensionales CAD-Modell in Schichten unterteilt und Maschinensteuerungsparameter festgelegt werden. Während für die polymere 3D-Drucktechnik zahlreiche Slicing-Werkzeuge frei verfügbar sind, erfordert DED-arc spezialisierte Softwarelösungen, da die Materialqualität stark von den Prozessparametern abhängt. Im Rahmen dieser Arbeit wurde zunächst eine Nutzwertanalyse quelloffener Slicer-Software-Lösungen durchgeführt. Für die verwendete Systemkonfiguration, eine GEFERTEC 3DMP Maschine (arc 605) in Verbindung mit einem Fronius TransPuls Synergic 4000 CMT Schweißsystem, erwies sich Cura als optimale Basisplattform. Durch die Implementierung angepasster Skript-Module wurde die Funktionalität für DED-arc-Anwendungen realisiert und erweitert. Der neu entwickelte Slicing-Workflow wurde anhand verschiedener Testgeometrien validiert und anschließend anhand eines Demonstratorbauteils mit dem Standard-Workflow über Rhino und Grasshopper verglichen. Die daraus hervorgegangene Methodik reduzierte die G-Code-Einrichtungszeiten konsistent von mehreren Stunden auf circa fünf Minuten bei mindestens gleichwertiger, meist signifikant verbesserter Form- und Maßhaltigkeit, was die Validität und Effektivität der Methodik unterstreicht.

Abstract Directed Energy Deposition with arc (DED-arc) is a promising technology for efficient and cost-effective production of metal components, in which the wire feedstock is melted using an electric arc. Compared to other additive manufacturing methods, DED-arc offers superior material deposition rates at lower cost. A key aspect is the slicing process, in which a 3D CAD model is subdivided into layers and machine control parameters are defined. While many free slicing tools are available for polymer 3D printing, DED-arc requires specialized software solutions since material quality strongly depends on process parameters. A utility analysis of open-source slicer software was conducted in this work. For the system configuration comprising a GEFERTEC 3DMP machine (arc 605) coupled with a Fronius TransPuls Synergic 4000 CMT welding system, Cura emerged as the optimal base platform. Enhanced functionality for DED-arc applications was achieved through implementation of custom scripting modules. The newly developed slicing workflow was validated using various test geometries and then compared with the standard slicing workflow using Rhino and Grasshopper on a demonstrator component. The developed methodology was found to consistently reduce G-code setup times from hours to approximately five minutes while achieving at least equal, in most cases significantly superior dimensional accuracy, emphasizing the validity and effectiveness of the methodology.

Introduction

Additive manufacturing (AM) has established itself across diverse industrial sectors, particularly in prototype development and tooling applications. However, the technology's potential extends significantly beyond these traditional applications, with an increasing number of organizations and research institutions implementing AM for series production and end-use component manufacturing. The primary advantage of AM technology lies in its design flexibility. The layer-by-layer material deposition approach enables realization of complex geometries while simultaneously improving material efficiency [1]. Within this context, DED-arc has gained considerable importance. In contrast to subtractive manufacturing processes that remove material through milling operations, DED-arc constructs components by melting metal wire with an electric arc and depositing it in successive layers. This approach provides high deposition rates, enables cost-effective fabrication of large-scale components, and yields mechanical properties comparable to those achieved through forging or casting processes [2].

Analysis of existing slicing software solutions for DED-arc reveals significant limitations. Commercial systems including MetalXL (MX3D), SculptPrint (Lincoln Electric), and 3DMP-CAM (GEFERTEC) are specifically developed for DED-arc applications and provide high-performance capabilities. However, they require substantial capital investment and offer limited flexibility for user-specific adaptations. Research-oriented developments, such as the software introduced by Ferreira et al. [3], provide restricted functionality and have subsequently been incorporated into proprietary research projects, limiting public accessibility. The modified polymer-based Material Extrusion (MEX) slicer MOSTMetalCura by Nilsiam et al. [4] exists solely as a command-line application without graphical user interface capabilities and, having been released in 2017, may not be compatible with features and improvements implemented in subsequent Cura versions. Parametric design environments such as Rhino with Grasshopper provide flexibility for DED-arc path planning through visual programming but require manual component composition and significant expertise in parametric design methodologies [5]. Consequently, no currently available solution adequately addresses requirements for user-friendly, flexible, and freely accessible DED-arc-compatible slicing software. This research addresses this gap through development of a cost-free, DED-arc-compatible slicer featuring a graphical user interface designed to follow established polymer MEX slicer operating logic. Through the use of open-source building blocks combined with plugin-based architecture, the solution enables flexible customization and future extensions. The slicer provides comprehensive parameters for process and quality optimization, conceived as a near one-click solution requiring minimal user intervention and no programming expertise. Furthermore, it facilitates rapid estimation of manufacturing time and material consumption, thereby improving process planning efficiency. This paper is structured as follows. A comprehensive benchmark of slicing software for DED-arc applications is presented, including reference workflow definition and utility analysis of polymer MEX slicers. The innovative workflow from CAD to CAM is then described, detailing Cura modifications and post-processing scripts. Validation through initial tests, demonstrator production, workflow comparison, and multi-axis applications follows. Finally, the paper concludes with a summary of findings.

Slicing software benchmark for DED-arc applications

Various software solutions with different functionality and flexibility exist for generating G-code for DED-arc processes. This section first presents a reference workflow based on Rhino/Grasshopper, serving as a comparison baseline for subsequent evaluation of different slicing solutions.

Definition of a general reference workflow

A structured DED-arc workflow utilizing Rhino 7 with Grasshopper consists of seven main sections as illustrated in Figure 1 [5].

1. Input Parameters and Settings: Definition of the user interface with process parameters such as layer height, hatching distance, travel speed, and DED-arc-specific inputs (curve offset for strike locations, seam position, lead heights).
2. Path Construction: Decomposition of the imported mesh using the Contour component into layer contours and generation of infill patterns with defined hatching distance and overlaps.
3. Lead Points: Insertion of approach and departure points at the beginning and end of each welding operation.
4. Text Concatenation: Deconstruction of point series into coordinate data and concatenation with text commands for path file formatting.
5. Operation Separation: Organization of welding operations into separate branches for individual command assignment.
6. Movement Commands: Generation of frames with end-effector orientations, external axis positions, and velocities for linear movement commands.
7. Analysis and Validation: Simulation of machine movements for collision checking and generation of final machine-readable code.

This seven-step Rhino/Grasshopper-based process defines a typical structure for DED-arc slicing in parametric environments and serves as a reference for the subsequent comparison of alternative software solutions. A flow chart of the general structure of the reference workflow is depicted in the appendix showing the Rhino/Grasshopper reference workflow structure.

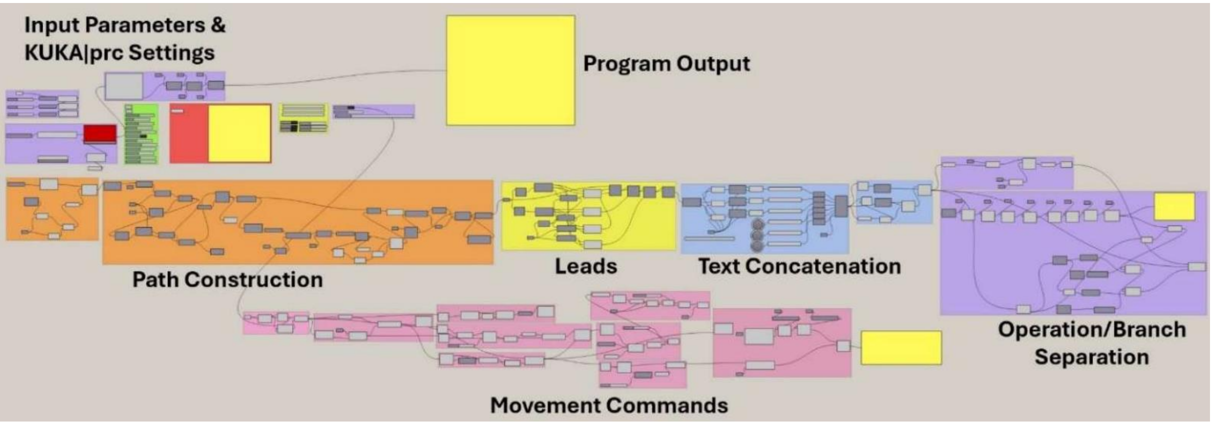


Figure 1: Diagram of general Grasshopper script structure [5]

Analysis and comparison of polymer MEX slicers based on a utility analysis

Five free-of-charge polymer MEX slicers - Bambu Lab Studio, Orca Slicer, PrusaSlicer, Cura, and Slic3r - were examined with respect to their infill patterns and functionalities. All programs were downloaded and systematically analyzed to identify similarities and differences [6-10]. Based on this analysis, specific evaluation criteria were developed, weighted, and subsequently combined in a utility value analysis (UVA). The results provide a decision-making basis for selecting the software that best meets the requirements of this work. Slicer functionalities were evaluated with a particular focus on features relevant to DED-arc applications. Notable differences were observed in structure and naming of categories, broadly divided into five main areas comprising printer settings, material settings, print settings, pre-slicing settings, and post-slicing settings. Additionally, aspects of open-source availability and plugin support were considered. Especially relevant for DED-arc are functions such as spiral vase mode, alternating extra wall, shrinkage compensation, and overhang adjustment. These features enable smooth outer surfaces, increased stiffness, compensation of shrinkage effects, and printable overhangs without support structures. At the same time, they reduce process time and improve process control. For slicer evaluation, a utility value analysis (UVA) was conducted, converting qualitative factors into numerical values (utility scores) [11]. The analysis consisted of four steps encompassing identification of relevant criteria, weighting, assessment of fulfillment degree, and calculation of overall score. Criteria were derived from categories including printer, material, and print settings along with pre- and post-slicing settings, open-source availability and plugins. To determine the relative importance of these criteria, a pairwise comparison was applied. In this method, all criteria are arranged in both rows and columns of a comparison matrix. The diagonal cells are excluded as they represent self-comparison. For each cell, the row criterion is compared against the column criterion. If the row criterion is considered more important, the cell receives a value of 2. If the column criterion is more important, the cell receives a value of 0. If both criteria are considered equally important, a value of 1 is assigned. The absolute and relative row sums then yield the weighting for each criterion [12]. These weights are subsequently used in the utility value analysis to calculate the overall score for each slicer. The results indicated that open-source availability represents the most important factor, followed by plugins and print settings, as these significantly affect quality, stability, and print time. Printer and material settings showed minor differences, while pre- and post-slicing settings were considered useful additional options. The resulting weighting is summarized in Table 1.

Table 1: Weighting individual criteria (pairwise comparison)

	OS	PRS	MA	PS	PR	PO	INF	PL	Row Sum (abs.)	Row Sum (rel.) [%]
OS: Open Source	---	2	2	2	2	2	2	2	14	25,0
PRS: Printer Settings	0	---	2	0	2	2	2	0	8	14,3
MA: Material Settings	0	0	---	0	2	2	2	0	6	10,7
PS: Print Settings	0	2	2	---	2	2	2	0	10	17,9
PR: Pre-Slicing Settings	0	0	0	0	---	1	0	0	1	1,8
PO: Post-Slicing Settings	0	0	0	0	1	---	0	0	1	1,8
INF: Infill Patterns	0	0	0	0	2	2	---	0	4	7,1
PL: Plugins	0	2	2	2	2	2	2	---	12	21,4
Result									56	100

The fulfillment degree for each criterion was rated on a six-point scale, where 6 represents very good performance and 1 indicates insufficient performance [11]. For utility value analysis, B denotes the unweighted score assigned to each slicer based on this scale, while P represents weighted points, calculated as unweighted score multiplied by criterion weight. Results (Table 2) demonstrate that Bambu Lab Studio, due to lack of open-source availability, and Slic3r, due to limited functionality, perform significantly worse. Orca Slicer and PrusaSlicer achieve comparable results but provide fewer print settings and lack plugin support. Cura features the highest utility value with 589.3 points, exceeding PrusaSlicer by 132.2 points, and has therefore been selected for further development in this study (version 5.8.1).

Table 2: Utility value analysis for slicer comparison

Crite-rion	Weight [%]	Bambu Lab Studio		Orca Slicer		Prusa Slicer		Cura		Slic3r	
		B	P	B	P	B	P	B	P	B	P
OS	25,0	1	25,0	6	150,0	6	150,0	6	150,0	6	150,0
PRS	14,3	6	85,7	6	85,7	6	85,7	6	85,7	3	42,9
MA	10,7	5	53,6	5	53,6	6	64,3	6	64,3	4	42,9
PS	17,9	4	71,4	4	71,4	4	71,4	6	107,1	3	53,6
PR	1,8	6	10,7	6	10,7	6	10,7	5	8,9	3	5,4
PO	1,8	6	10,7	6	10,7	6	10,7	5	8,9	3	5,4
INF	7,1	6	42,9	6	42,9	6	42,9	5	35,7	4	28,6
PL	21,4	1	21,4	1	21,4	1	21,4	6	128,6	1	21,4
Utility Value		321,4		446,4		457,1		589,3		350,0	

Innovative workflow from CAD to CAM

The workflow, which is developed in this chapter, encompasses both modifications within Cura and the creation of corresponding post-processing scripts that directly manipulate the G-code after slicing. It should be noted that post-processing scripts can either be integrated directly within Cura or executed externally. In both cases, the scripts operate on the generated G-code after the slicing process has been completed. For the work at hand, the scripts were developed externally as it allows for the integration of downstream functionalities, such as time and material estimation and toolpath visualization. The complete workflow architecture, from CAD import through G-code generation to machine-ready output, is visualized in the appendices, with the modified Cura-based workflow appendix showing the overall process flow and the post-processing scripts appendix detailing the script structure.

Modifications in Cura

Initially, start and end G-code sequences were adapted for general machine initialization and shutdown procedures, including coordinate system definition, homing sequences, and safety protocols. Additionally, DED-arc-specific process parameters were defined for stainless steel 1.4430. Parameters such as line width and overlap were configured directly within Cura's interface, while others including wire feed speed, welding current, and cooling pauses were integrated through the post-processing scripts, as these parameters directly affect the quality of the weld beads and the dimensional accuracy of the components. Table 3 provides an overview of the essential Cura settings for producing high-quality prints. Cura offers numerous additional options including number of perimeters, infill patterns, and print sequence, adjustable according to application requirements. To avoid repetitive modifications, profile creation is recommended. For specific cases, such as stainless steel 1.4430 in bulk components, dedicated profiles for printer, material, and print settings should be created and clearly labeled, ensuring all relevant parameters are preconfigured, thus requiring only minor adjustments for each print.

Table 3: Basic Cura settings

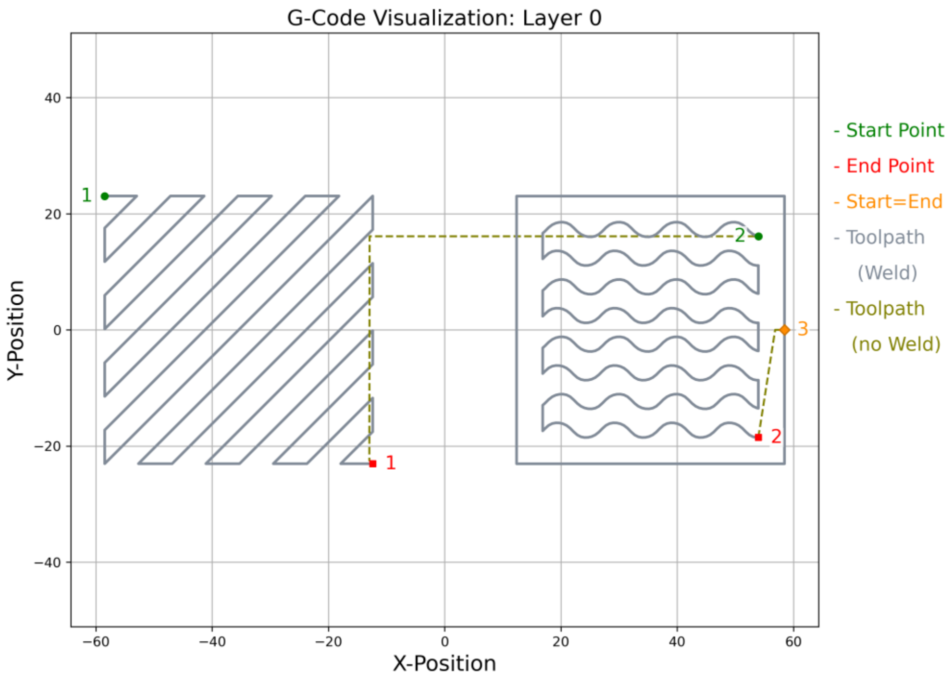
Setting Type	Parameters
Printer Settings	Printer size, coordinate system origin: center, heated bed: no, nozzle size, compatible material diameter
Material Settings	Density, designation, description, and cost of the respective material
Print Settings	General layer height, line width, line spacing (≙hatching distance), infill, randomize start of infill: yes, welding speed, travel speed

Post-processing scripts

Following G-code export from Cura, further processing has been implemented with a series of post-processing scripts adapting originally polymer MEX-generated code for DED-arc processes. Scripts are implemented in Python within the open-source development environment Spyder and form a central component of the developed workflow. The process initiates with G-code preprocessing. The script determines the heights of parts to be printed, outputting them with part names to the console. For multiple parts, display is provided individually. Additionally, the script verifies correct Cura slicing, as Cura calculates layer count based on constant layer height. Deviations between calculated and actual layer count are detected by the function *correct_slicing*, though processing continues despite warnings, as minor differences can be compensated for during subsequent machining operations. Since DED-arc components are typically manufactured with deliberate oversize allowances for post-processing through milling, missing or additional layers often remain within acceptable tolerances. Superfluous lines including thumbnails and polymer MEX initializations are removed through the function *process_gcode_file* to reduce memory requirements and optimize G-code for DED-arc. Subsequently, the central Metal Printing script converts polymer MEX G-code into DED-arc-compatible code. The first Z-value, corresponding to layer height, is extracted. For DED-arc, an additional Z-offset is calculated, accounting for substrate thickness and stick-out distance to ensure correct vertical positioning. All speeds are recorded, linking each line to corresponding velocity.

The subsequent *parse* function, while initially inspired by a 3D metal printing slicer plugin developed in 2015 by Mike Vennard for an open-source metal 3D printer [13], has been extensively modified and enhanced for DED-arc-specific requirements. The significantly expanded function now processes layer changes, movements, and extrusions while removing polymer MEX-specific commands including fan control and inserting DED-arc-relevant process commands, safety measures, pre-welding temperature checks, and post-welding dwell times. The main function *process_gcode_metal* calls parsing routines and controls complete G-code processing. During post-processing, all remaining unnecessary E-commands are deleted. The function *compare_m42_m43_counts* ensures matching numbers of weld starts and stops. Discrepancies are corrected with additional weld stops and flagged with warnings. Each weld bead is consecutively numbered to allow resumption after process interruptions. Optionally, temperature controls between parts or layers can be inserted, with the function *check_and_insert_temp_ctr* ensuring execution under specific conditions. All numerical values are rounded to three decimal places, and counter numbers are inserted to improve readability and traceability. The final step includes simulation of completed G-code to visualize toolpaths, check for Z-direction collisions, and estimate time and material consumption. All relevant X- and Y-coordinates, movement types, layer information, and visualization points are extracted and connected. Time estimation accounts for travel paths, welding times, dwell times, and wire feed, including tolerances for feed fluctuations. For each layer, time and material consumption are calculated, enabling precise production planning and improved material preparation. Simulation also enables visualization of start and end points of each weld and toolpath analysis, crucial for quality control of final G-code. To automate execution of all four post-processing scripts, a master script ("Postprocessor") was developed. This script requires only the folder path containing the G-code file and scripts, with all parameters, input data, and options preconfigured but adjustable as needed. Upon execution, the master script automatically processes the current G-code file, removes obsolete files, and creates a project folder containing all relevant files including the final Machine Program File (MPF) for the GEFERTEC system. The master script also enables configuration of fundamental parameters such as substrate thickness and stick-out distance, while special functions can be activated through switches. For instance, the measurement point for inter-layer temperature determination can be adjusted, with the option to set *use_first_point* to "False" for small components requiring uniform temperature distribution, while maintaining the default start point for larger components to optimize manufacturing time. Table 4 presents exemplary results generated by the simulation script. Since Cura designates the first layer as "Layer 0" by default, this convention is maintained to ensure consistency in representation.

Table 4: Illustration of results based on the developed simulation script

Result of the Toolpath Visualization	
Results of Time and Material Estimation	Time per layer [min]: ['0: 2.39', '1: 1.94'] Material weight per layer [kg]: ['0: 0.08', '1: 0.06'] Total processing time with welding: 0h 3min 18s Total processing time without welding: 0h 0min 54s Total dwell time: 0h 0min 16s Total processing time including pauses: 0h 4min 28s Total filament length: 21.73 +- 1.20 m Total filament mass: 0.14 +- 0.01 kg Filament cost: 2.18 +- 0.12 Euro

Validation

This section comprehensively validates the newly developed slicing workflow to ensure reliable production of functional and correctly implemented G-code for DED-arc processes. Furthermore, initial optimization is conducted to ensure adequate overlap between adjacent weld beads to prevent void formation and guarantee structural integrity. All tests were performed on a system configuration comprising a GEFERTEC 3DMP machine (arc 605) coupled with a Fronius TransPuls Synergic 4000 CMT welding system.

Generating and performing initial tests

To validate the modified slicer, six test geometries with single layers were defined, examining various infill patterns, perimeter configurations, and overlaps. The primary objective was verification of correct G-code generation under different parameters rather than print quality optimization. All tests were completed successfully, with simulations from Cura and the post-processing script used for toolpath visualization. Figure 2 presents results from Test 1.

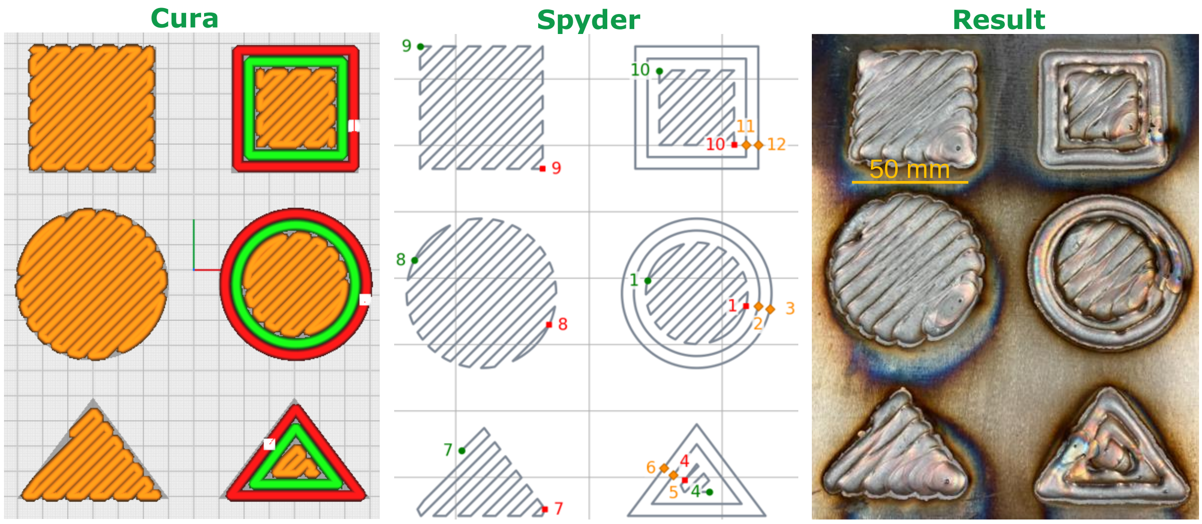
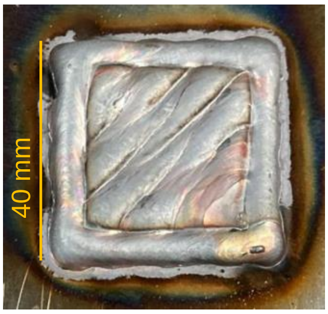


Figure 2: Result from Test 1 (Layer 0)

Visual inspection of manufactured samples revealed that using infill only produced good overlap between individual paths, while gaps could occur between infill and perimeter if the latter was set. The Zigzag pattern and Gyroid pattern proved particularly effective prints, requiring fewer welding stops and reducing internal stresses. Conversely, connected concentric, cross, and grid patterns were found unsuitable for DED-arc due to gaps or multiple overlapping welds. The printing sequence of infill and perimeter affects height and width of perimeter. However, this factor is of secondary importance in practice, as infill is generally printed first to provide a stable base. Overall, results indicate that Zigzag, Gyroid, and concentric infill strategies are most promising for DED-arc processes, while further optimization of overlap between infill and perimeter, as well as between adjacent paths, remains necessary. For structural DED-arc applications, a pore-free material morphology is essential, as voids and cavities generate stress concentrations and promote crack formation [14]. Initial tests revealed insufficient overlap between infill and perimeter as well as between individual perimeters, resulting in gaps. Since Cura provides no explicit setting for perimeter-to-perimeter overlap, line width was varied to ensure adequate overlap. This slightly increases part dimensions but is acceptable due to standard post-processing. For infill-only regions, a line spacing of 3.3 mm was applied. The "Infill Overlap" setting is used to control interaction between infill and perimeter. The optimal parameters for infill-perimeter interaction are summarized in Figure 3. For a single perimeter, the line width is 4.5 mm with an overlap of 1.6 mm, while for multiple perimeters, the line width is 3.3 mm with an overlap of 1.0 mm. These settings effectively prevent gaps and ensure the correct layer height.

Infill and One Perimeter
Line width: 4,5 mm
Infill overlap: 1,6 mm



Infill and Multiple Perimeters
Line width: 3,3 mm
Infill overlap: 1,0 mm

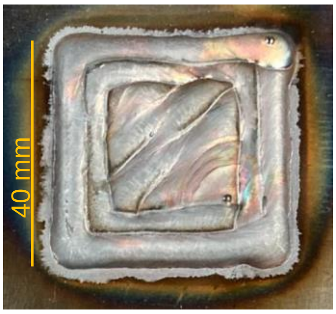


Figure 3: Results of the overlap tests with one perimeter (left) and multiple perimeter (right)

Demonstrator production

To validate both approaches, a demonstrator structure was manufactured using both the established and the optimized slicing workflow. In the established process, G-code generation was performed in Rhino/Grasshopper, with 16 layers at a height of 2.65 mm specified for the demonstrator and the substrate plate preheated accordingly. In the optimized workflow, the model was loaded into Cura, configured with the GEFERTEC machine printer profile, Zigzag infill was selected, and the resulting G-code was converted into final machine code via the master script. Both variants were manufactured on the same GEFERTEC machine.

Comparison and evaluation of the two workflows

The comparative analysis was conducted both at the process level and based on manufactured demonstrators. A stepped ring with an outer diameter of 300 mm, stepped inner diameters of 200 mm and 260 mm, and a total height of 42.4 mm, manufactured in 16 layers with a layer height of 2.65 mm, served as the test geometry.

Functional scope and flexibility

An approach similar to the reference workflow using Rhino/Grasshopper is employed. This provides 14 adjustable functionalities, including layer height, hatching distance, and infill angle. In comparison, Cura offers over 400 parameters for the polymer MEX domain. Although not all polymer MEX functionalities are directly transferable to DED-arc, Cura enables significantly higher flexibility in process optimization. The limitations in Rhino/Grasshopper can be compensated through integration of new components via Python scripts, which however requires substantial programming knowledge and development time. A significant advantage of the Rhino/Grasshopper approach lies in the implementation of multi-axis operations through rotational and pivoting movements of the machine table (A- and C-axes), while Cura is limited to Cartesian X, Y, and Z axes. However, the Grasshopper architecture presented in the reference workflow definition is not universally applicable. Peter et al. [5] note that dedicated script adaptations continue to be developed for complex geometries with holes or tapering cross-sections, causing implementation effort to vary significantly with geometrical complexity.

Process flow and time efficiency

The two slicing workflows differ fundamentally in their process structure, as shown in Appendix. The Rhino/Grasshopper workflow requires manual composition of functional blocks and their linkage, while the Cura-based approach follows a linear, standardized sequence. G-code generation shows significant differences in process duration. The new Cura-based workflow consistently requires 5 minutes for the complete preprocessing phase including model import, parameter configuration, Cura slicing, and subsequent G-code conversion through the four specialized script modules (preprocessing, metal printing, post-processing, and simulation), which provide code validation, collision control verification, and processing time and material consumption estimates. With standard settings, slicing is even possible in under 2 minutes. The Rhino/Grasshopper process varies considerably. With existing components and standard geometries such as the stepped ring, the duration is approximately 10 minutes. However, this assumes that all required Grasshopper components have already been developed and the architecture is established. For more complex geometries or missing strategy components, the process can extend over several hours up to weeks, as new components may initially need to be programmed using Python scripts. In addition, this workflow requires a higher level of technical expertise in working with parametric design environments.

Dimensional accuracy, material efficiency and surface appearance

Both workflows employed Zigzag infill patterns but differ in implementation: The Cura-based workflow maintains a constant hatching distance, while in the Rhino/Grasshopper-based one it is varied towards the center, inherent to the slicing strategy of that platform. A relative performance comparison was therefore carried out with regard to dimensional accuracy, as a strict 1:1 comparison is not possible for the configuration at hand. Dimensional accuracy was evaluated using calipers and GOM scan (ATOS TRIPEL, 28 μm resolution). The Cura-based workflow showed a mean dimensional deviation of -0.18 ± 0.87 mm compared to 0.37 ± 1.04 mm for the Rhino/Grasshopper approach. Histogram analysis revealed an approximately normally distributed deviation for the new workflow, while the reference workflow exhibited a second peak at 2 mm positive deviation, cf. Figure 4. The Rhino/Grasshopper approach led to systematic oversizing at the inner diameter (195.66 mm instead of 200 mm, deviation -2.17%), directly attributable to the varying hatching distance implementation. These geometric deviations correlated directly with weight deviations. The actual component weight deviated by -4.56% (9.20 kg) from the calculated CAD weight (9.64 kg) for the Cura workflow, while the reference workflow showed a positive deviation of +11.51% (10.75 kg). This demonstrates that the slicing strategy employed in the Cura workflow can replicate the target geometry more accurately overall. Regarding surface appearance, more pronounced notches are observed in the proposed workflow. These are attributable to both the selected slicing strategy and the layer offset caused by altered orientation. While the reference slicing workflow maintains a constant infill pattern orientation with layers only translationally offset, the proposed workflow implements a 90° rotation of path orientation between layers. Consequently, gaps cannot be compensated as effectively by subsequent layers, leading to more pronounced surface texture. A potential optimization approach would involve reducing the rotation angle between consecutive layers, for instance to 45° instead of 90°, which may result in a more uniform material distribution and thus reduced notch formation. However, the present investigation prioritizes internal material quality and dimensional accuracy over surface finish, as DED-arc components are typically subjected to post-processing through machining operations as standard practice. Surface irregularities within the machining allowance are therefore of secondary importance for most industrial applications. Figure 4 shows the comparative result images and GOM scan analyses with histograms of both workflows.

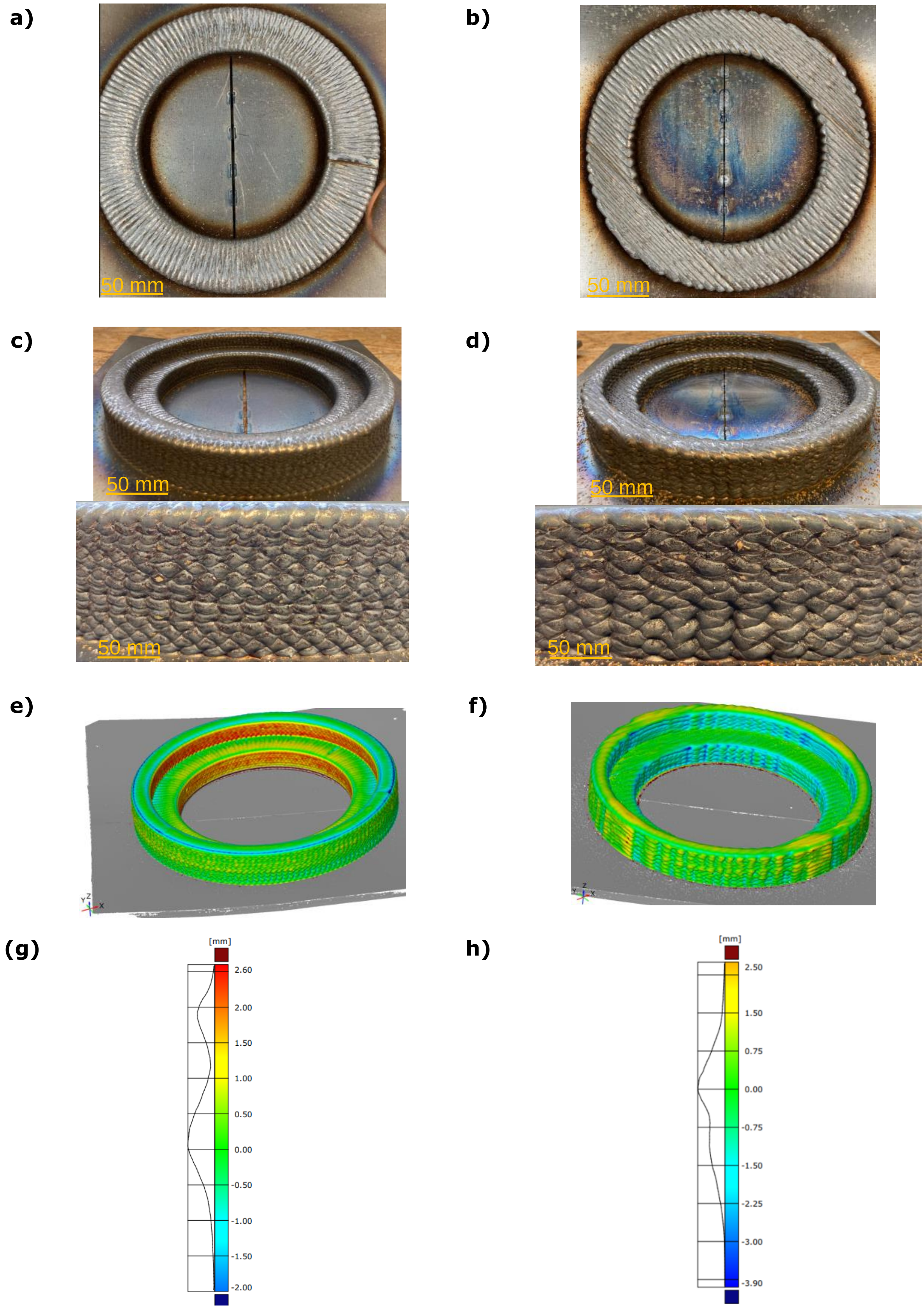


Figure 4: Demonstrator manufacturing trial results comparing reference (left) and proposed slicing workflow (right): (a, b) first layer surface appearance, (c, d) complete part overview, (e, f) GOM scan deviation analysis, (g, h) deviation histograms

Component defects and manufacturing quality

To probe manufacturing quality, both stepped rings were cut in halves and examined macroscopically. Cross-sectional analysis showed largely void-free components for both workflows, with a single small void identified in stepped ring manufactured according to the reference slicing workflow (see Figure 5). While multiple polished samples or computed tomography would be required for a statistically sound statement on void freedom, the performed cross-sectional analysis provided an initial qualitative assessment of component integrity. Both approaches exhibited comparable substrate plate distortion of approximately 5 mm in the outer diameter region, caused by residual stresses during the non-uniform cooling process. A process-related weakness of the Cura workflow proved to be the 2 to 3 start/end points per layer compared to a single one in the reference workflow, increasing the statistical probability of local defects.

a)

b)

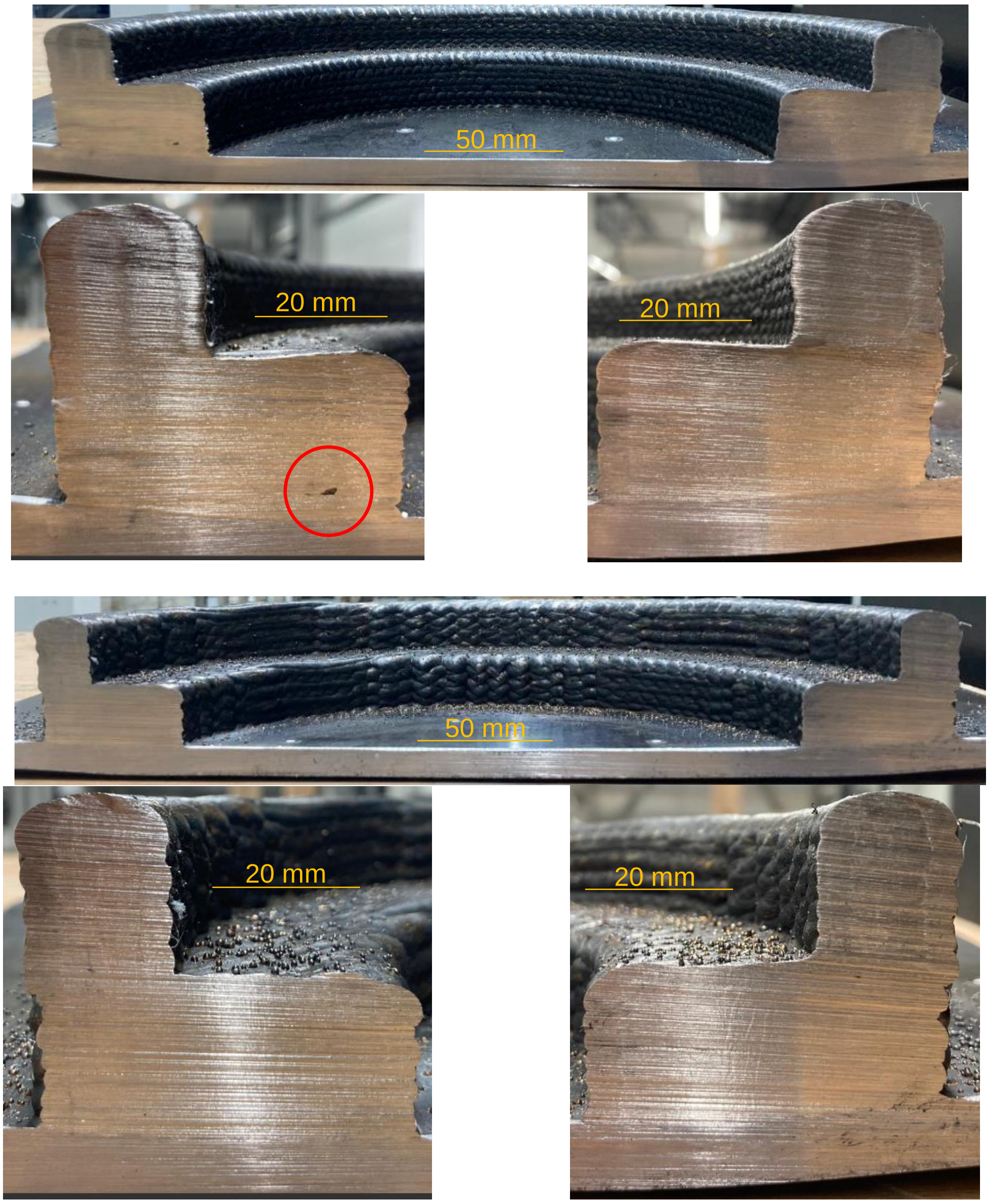


Figure 5: Cross-sectional analysis of demonstrator components showing overview and detail views: (a) reference slicing workflow, (b) proposed slicing workflow

Evaluation of time and material estimates

The integrated time and material estimation of the simulation script was evaluated based on all preliminary tests as well as the demonstrator fabrication and compared with the values obtained from Cura. The time measurements were recorded manually, taking into account cooling pauses during the process. The results indicate that the simulation script represents the actual manufacturing times significantly more accurately than Cura, particularly for single-layer prints, where no cooling pauses occur. Deviations between simulated and actual times arise from factors such as start positions, stop errors, cooling pauses, and machine control. Regarding material estimation, the calculated weight was consistently slightly below the actual consumption. An analysis of the actual wire feed revealed a deviation of 4.3% from the ideal value on average. Additionally, the wire feed exhibits fluctuations within a tolerance range of ±5.5%. After adjusting for these variations, the analysis of the demonstrator shows that the actual wire consumption falls within the predicted tolerance (see Table 5), confirming the accuracy of the estimations.

Overall, the time and material estimation provides reliable guidance for planning and cost assessment, even though exact times may deviate due to cooling pauses or due to variations related to machine internal control requirements. For future print jobs, systematic data collection is recommended to further optimize material consumption and deposition rate predictions.

Table 5: Adjusted material estimate for the demonstrator

	Weight (actual) [kg]	Calculated Weight (Tolerance 5.5%) [kg]
Demonstrator	9,20	9,46 ± 0,52

Application in multi-axis manufacturing processes

To extend the modified slicer for multi-axis DED-arc processes, a rotation of the A-axis (around the X-axis) is implemented. Standard slicers, such as Cura, typically only consider movements in the XY plane, while the Z-axis is incremented layer by layer. For overhangs or complex geometries, this can lead to issues. Components may become inaccurate, or support structures may be required, increasing manufacturing time, material consumption, and post-processing. Furthermore, certain geometries may not be reproduced optimally due to restrictions in line width. Figure 6 shows an example component in the YZ plane. The original model features a sharp corner that cannot be reproduced accurately during slicing in Cura. In particular, deviations occur in the upper corner area, yielding slicing errors at that location. It is important to note that while an overhang angle of 34.6° (measured from vertical) is generally unproblematic in polymer MEX processes, DED-arc is significantly more sensitive. In DED-arc, even small overhang angles below 20° from a vertical direction can already cause manufacturing issues, while angles exceeding 20° are particularly critical and may not be manufacturable or are produced only inadequately [15].

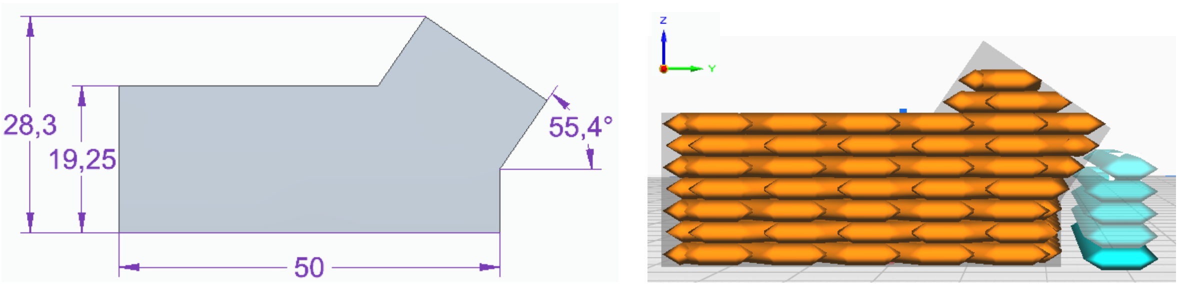


Figure 6: Overview of the example component original (left) and sliced with Cura (right) in the YZ plane

To address these challenges, the G-code must be adjusted after each segment. Two approaches are generally possible. Either the machine manufacturer is contacted to enable table rotation without generating a new coordinate system, or the coordinates of the original G-code are adjusted via an Euler transformation. In this work, the latter approach is implemented, where the Y and Z values are referenced to the rotated coordinate system while the X value remains unchanged [16]. The Euler transformation is described by the standard rotation matrix around the X-axis (Equation 1):

$$P_{new} = D_x(\Phi) \cdot P_{old} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos\Phi & -\sin\Phi \\ 0 & \sin\Phi & \cos\Phi \end{pmatrix} \cdot \begin{pmatrix} x_{old} \\ y_{old} \\ z_{old} \end{pmatrix} \tag{1}$$

The implementation is realized through the extension of the post-processing scripts. The function *parse_mehrachs* generates layer-specific information for table rotations, *adjust_z_values* corrects Z-offsets, the Euler transformation modifies the coordinates, and *extract_a_values_per_layer* verifies the A-axis values per layer. These multi-axis extensions and their integration into the post-processing script structure are visualized in the post-processing scripts appendix. The master script enables segmentation into slicing sections, including rotation angles and Z-offsets. An initial test confirms the functionality of the concept (Figure 7). The first segment is printed, the table is rotated, and printing continues. Due to the stair-step effect and Cura's slicing logic, small craters appear between segments, and inclined edges are only approximated. This approach is particularly suitable for repair operations on pre-machined surfaces, but further optimization is required to produce high-quality components.

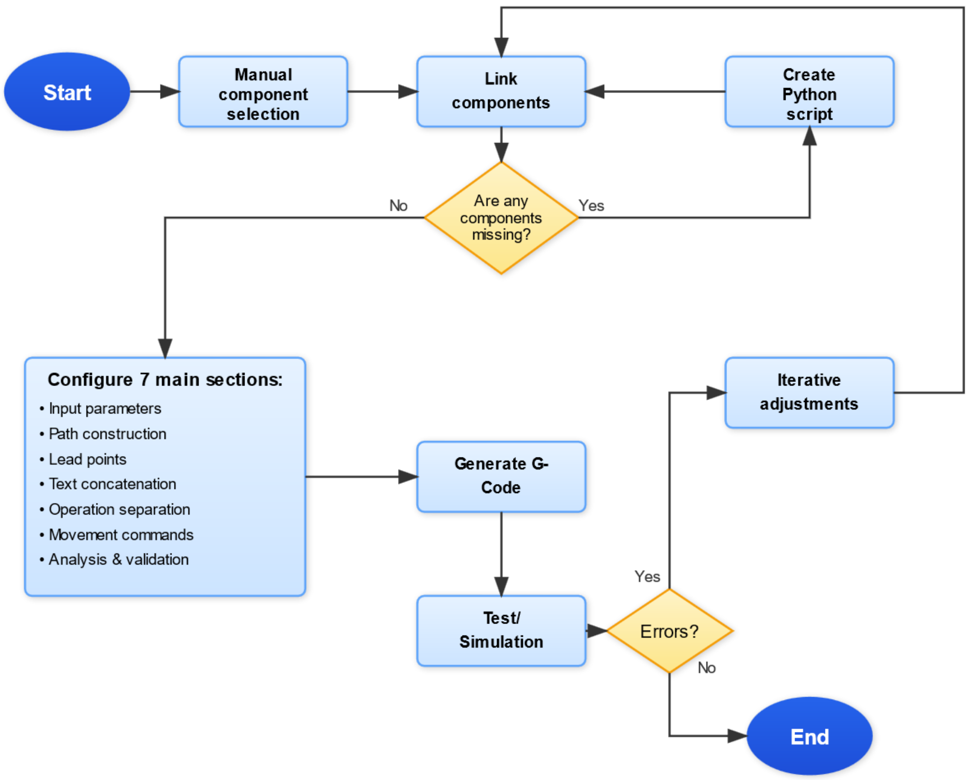
Figure 7: Test result of the multi-axis application

Summary and conclusions

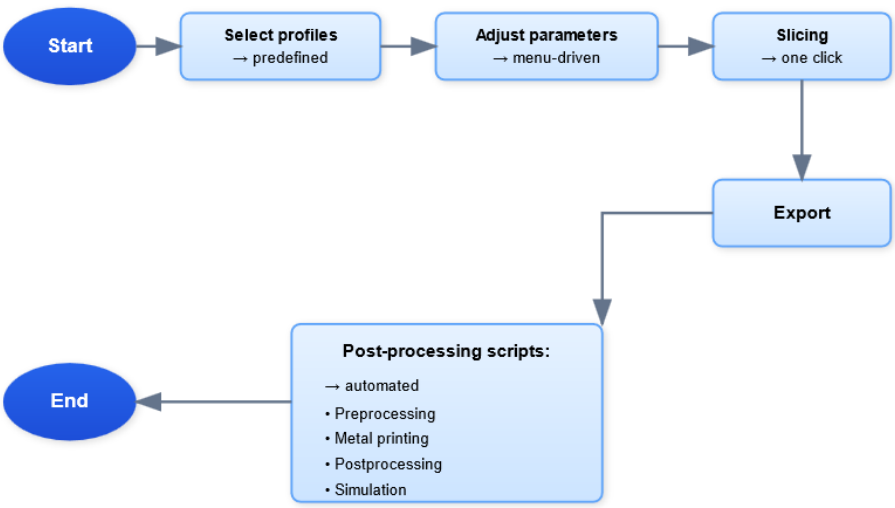
A modified polymer MEX slicer for DED-arc applications has successfully been developed by adapting open-source Cura software with specialized post-processing scripts. The utility value analysis of five polymer MEX slicers identified Cura as the optimal platform (589.3 points), primarily due to its open-source nature and extensive parameter set. The developed workflow integrates modified Cura profiles, Python-based post-processing scripts, and simulation modules for toolpath visualization. Comparative analysis with a Rhino/Grasshopper reference workflow demonstrated significant improvements: G-code generation times reduced from a variable 10 minutes to weeks down to consistent 5 minutes. Comparing design to process routes on manufacturing trials naturally involving different printing strategies, dimensional accuracy has been found to improve significantly for the stepped ring investigated in this study. In addition, enhanced material efficiency was noted reflected by a weight deviation of -4.56% in the proposed slicing workflow compared to +11.51% for standard preparation. In the validation tests, optimal overlap parameters have been established (4.5 mm/1.6 mm for single perimeters; 3.3 mm/1.0 mm for multiple perimeters), preventing gaps in mechanically loaded components. Initial cross-sectional examination showed largely void-free components for both workflows, with one small void in the reference workflow, indicating promising structural quality for industrial viability. The multi-axis extension through Euler transformation was successfully implemented, demonstrating the framework's adaptability for future developments.

Appendix

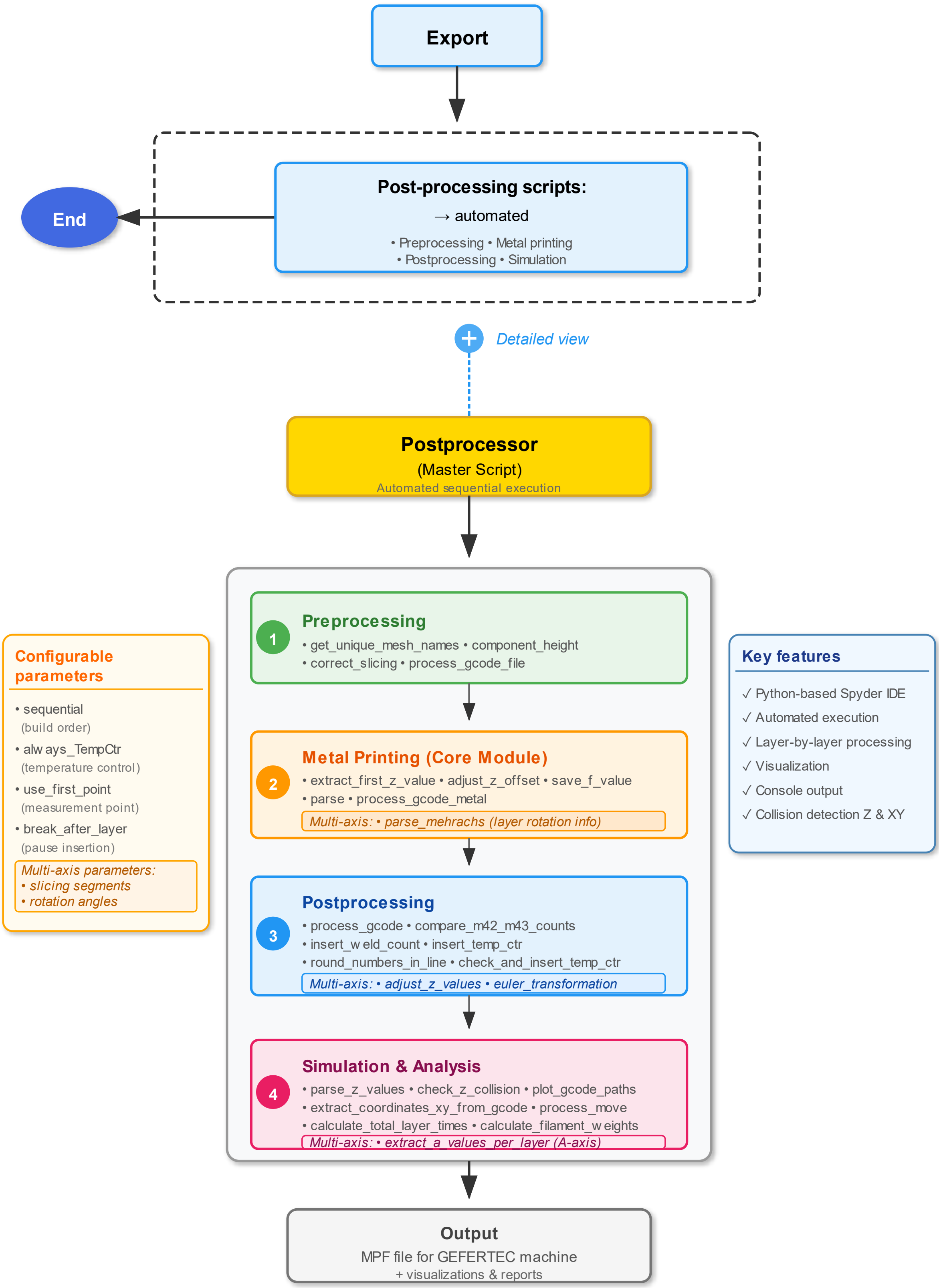
Rhino/Grasshopper-based reference slicing workflow



Modified Cura-based slicing workflow



Post-processing scripts - detailed view



E JOURNAL

References

[1] Klahn, C.; Meboldt, M.; Fontana, F.; Leutenecker-Twelsiek, B.; Jansen, J.: Entwicklung und Konstruktion für die Additive Fertigung. 1. Auflage, Vogel Business Media GmbH & Co. KG, Würzburg 2018.

[2] Raut, L. P.; Taiwade, R. V.: Wire Arc Additive Manufacturing: A Comprehensive Review and Research Directions. In: J Mater Eng Perform 30 (2021), S. 4768-4791.

[3] Ferreira, R. P.; Vilarinho, L. O.; Scotti, A.: Development and implementation of a software for wire arc additive manufacturing preprocessing planning: trajectory planning and machine code generation. In: Weld World 66 (2022), S. 455-470.

[4] Nilsiam, Y.; Sanders, P.; Pearce, J. M.: Slicer and process improvements for open-source GMAW-based metal 3-D printing. In: Addit Manuf 18 (2017), S. 110-120.

[5] Peter, H. R.; Sargent, K.; Penney, J.; Schmitz, T.: Path programming in Rhino 7 for wire arc additive manufacturing. In: Manuf Lett 44 (2025), S. 973-979.

[6] Bambu Lab. In: Bambu Lab. Internet: https://bambulab.com/en, 18.12.2024.

[7] Orca Slicer. In: Orca Slicer. Internet: https://orcaslicer.net/, 22.01.2025.

[8] Prusa. In: Prusa. Internet: https://www.prusa3d.com/de/, 22.01.2025.

[9] UltiMaker. In: UltiMaker. Internet: https://ultimaker.com/de/, 22.01.2025.

[10] Slic3r. In: Slic3r. Internet: https://slic3r.org/, 22.01.2025.

[11] Kühnapfel, J. B.: Scoring und Nutzwertanalysen. Springer Nature, Wiesbaden 2021.

[12] Projekte-leicht-gemacht: Paarweiser Vergleich. In: Projekte-leicht-gemacht. Internet: https://projekte-leicht-gemacht.de/ blog/business-wissen/paarweiser-vergleich/, 19.12.2024.

[13] APPROPEDIA: 3D Metal Printing Slicer Plugin. In: APPROPEDIA. Internet: https://www.appropedia.org/3D_Metal_Printing_Slicer_Plugin, 18.02.2025.

[14] Dindorf, C.: Ermüdung und Korrosion nach mechanischer Oberflächenbehandlung von Leichtmetallen. Dissertation, Technische Universität Darmstadt, Darmstadt 2006.

[15] Lam, T. F.; Xiong, Y.; Dharmawan, A. G.; Foong, S.; Soh, G. S.: Adaptive process control implementation of wire arc additive manufacturing for thin-walled components with overhang features. In: Int J Adv Manuf Technol 108 (2020), S. 1061-1071.

[16] Balla, J.: Lineare Algebra. Springer Spektrum, Berlin 2023.

Contact

Thomas Piendl (Corresponding author)
Am Grohberg 1, 92331 Lupburg
E-Mail: Thomas.Piendl@fit.technology