Solid Freeform Fabrication 2021: Proceedings of the 32nd Annual International
Solid Freeform Fabrication Symposium – An Additive Manufacturing Conference
Reviewed Paper

# ORNL Slicer 2.0: Towards a New Slicing Paradigm

Michael Borish*, Alex Roschli*

*Manufacturing Demonstration Facility, Oak Ridge National Laboratory, Knoxville, TN 37932

## Abstract

One fundamental step of additive manufacturing is slicing. Slicing is the conversion of a 3D mesh to a set of layers containing all the necessary pathing to construct the object. The slicing process is typically viewed as one step in a sequential additive manufacturing workflow: an object is designed in CAD, sliced, and subsequent G-code is sent to the additive manufacturing system for construction. While successful, this workflow has limitations such as the utilization of sensor feedback for pathing alteration. To address limitations and better take advantage of opportunities resulting from the Industry 4.0 revolution, researchers at Oak Ridge National Laboratory developed a new slicer, ORNL Slicer 2.0. Slicer 2.0 was developed with the concept of “on-demand” slicing whereby the slicer takes a more active role in object construction. In this paper, we describe the fundamental design philosophy of this new approach as well as the Slicer 2.0 framework.

Keywords: slicing, on-demand slicing, additive manufacturing, sensor feedback, modeling feedback

## Introduction

Industry 4.0 represents the next evolution of manufacturing technology [1]. As part of the Industry 4.0 revolution, various technologies have been identified as key components of the modern manufacturing center.  Among those technologies is additive manufacturing.  Additive manufacturing is the layer wise construction of 3D objects. Additive manufacturing encompasses a wide variety of machine types, construction processes, and materials.  Below are two such examples of large-scale

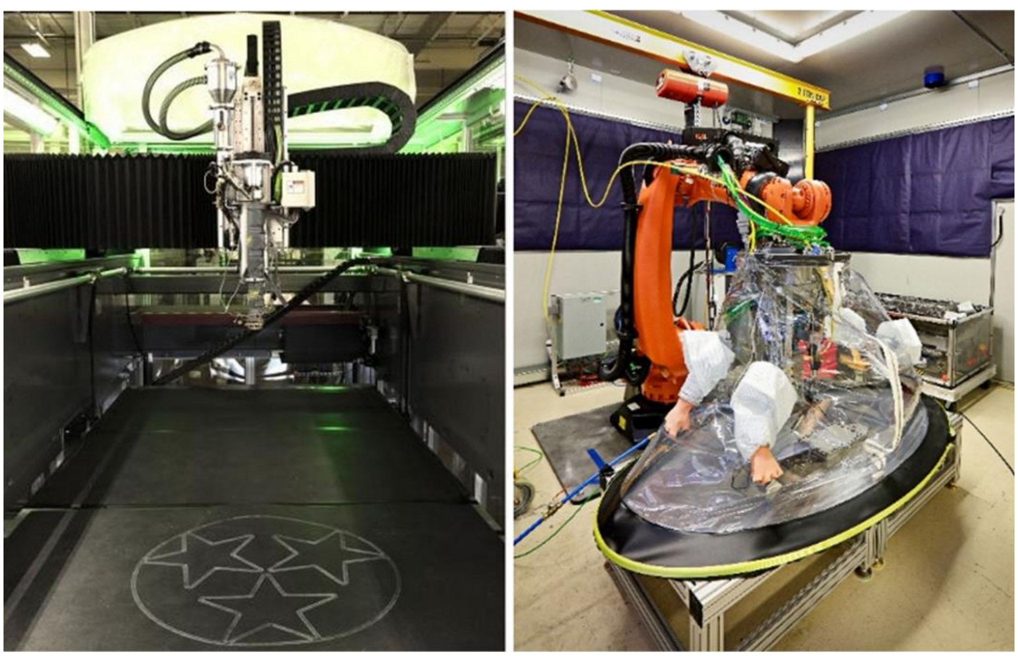

*Figure 1: Example large-scale additive manufacturing systems. Left: Large-scale gantry-based pellet fed polymer system. Right: Robotic arm-based metal filament Directed Energy Deposition (DED) system*

additive manufacturing systems showing a gantry-based polymer system and robotic-arm based metal system [2, 3].

Underlying all these systems is a slicing process. Slicing is the conversion of a 3D mesh to a set of layers with each layer containing all the necessary pathing for object construction. The general workflow is the same regardless of the system in question. First, an object is designed in CAD. This object is typically exported as an STL and imported into a slicing software package. The slicing software then cross-sections the object at appropriate layer heights and lays out pathing for each layer. This pathing is then converted to G-code, a standardized, human-readable, construction command format, that is ingested by the additive manufacturing system. Once parsed, the AM system then constructs the object based on the G-code input. So, in general, the slicing process is a sequential series of steps leading to the construction process.

This type of slicing workflow generally lays out pathing based purely on the geometry of the object that will be constructed. Additionally, such a slicing process is based on the underlying assumption that the construction process will be perfect. That is, the construction will exactly mirror the G-code commands provided by the slicer. Obviously, this is not always the case, and a significant body of work has investigated ways to augment the build process through the addition of sensors, modeling, or machine learning for example. While these investigations have been successful, they are generally open loop feedback. This type of feedback has limitations.

To address these limitations and take advantage of the full breadth of capabilities provided by advancement of the Industry 4.0 evolution, researchers at Oak Ridge National Laboratory developed a new slicing package. This slicer, called ORNL Slicer 2.0, was designed with the concept of "on-demand" slicing. On-demand slicing is the generation of select portions of G-code so that pathing can be informed by additional capabilities. In this way, the slicer becomes an active participant in the construction, leading to closed loop feedback for greater flexibility in the construction process. This new workflow opens the door for new investigations into the appropriate role of slicing in the construction process and stands in contrast to the commonly accepted methodology of sequential steps. In this paper, we will outline this new workflow, the impact to the design of the slicer software, and examples of the capabilities that the design allows.

## Related Work

As mentioned, slicing has typically been based purely on geometry. This geometry is the cross-sectional area of the mesh at any given height. However, a significant amount of work has investigated approaches that includes additional information. Feature-based slicing is one example [4]. There are multiple ways to approach feature-based slicing, but one of the most common involves additional consideration of the surface of the slice.

Relatedly, work has sought to include modeling and machine learning as part of the slicing process as well. In Peng et al, a model for predictions of stress and distortion based on thermal properties [5]. Liu, Yan, and Yu investigated topology optimization based on stress fields [6]. Similarly, work by Li et al. also used topology optimization based on stress fields to drive their construction processes [7]. There has also been effort in understanding the impact of process parameters and pathing on mechanical properties. For example, Koepf et al. investigated impacts of the construction process on grain size in metal powder systems [8]. Data can also be used to construct a digital twin of the object for modeling purposes such as in work done by Liu et al. [9].

Entirely new slicing approaches are also being investigated. Recently, collaborative slicing has become popular with the rise of systems with more than one build point. In work by Manoharan et al, volume decomposition was investigated to segment the slicing process for unique build volumes of multiple robotic arms [10]. Similar work was conducted by Djuric and Urbanic and Michel et al. that investigated modular approaches to slicing for multiple agents [11, 12]. McPherson and Zhou also investigated attempts at segmentation by breaking the slicing process into chunks that were later assigned to multiple printing agents [13]. Collaborative systems also include gantry-based systems with multiple independent gantries such as in work by Roach et al [14].

In addition to entirely new approaches, work has also investigated methodologies to speed up the slicing process through the use of GPU-based solutions. For example, in work done by Mao, Chen, & Wang a procedure for utilizing the GPU to calculate adaptive slicing parameters was utilized [15]. In work by Zhang et al., the process of calculating cross-sections was implemented entirely as a GPU-based algorithm as opposed to the traditional CPU-based algorithms [16]. In this work, the segments that comprised the triangular faces of the mesh were pre-processed by sorting and assignment to buckets. Each of these buckets could then be computed on the GPU. A similar approach to Zhang et al. was used by Rebaioli et al. to speed-up cross-sectioning for their system as well [17]. These approaches realized a significant speed-up.

Work has also investigated non-planar slicing techniques. In work by Ogayar-Anguita et al., a GPU approach was utilized to convert objects into voxels [18]. These voxels were later used for machine learning training. In work by Yigit and Lazoglu, non-planar slicing approaches for robotic arm-based systems was investigated to take advantage of the additional degrees of freedom such a system offers [19]. In work by Lefebvre, the entire slicing process was reimagined on the GPU using an A-buffer lookup [20]. Though technically planar, in the A-buffer approach, cross-sectioning and hull generation can be seen as steps of the graphical rasterization process.

This work builds upon all of this in a complementary fashion. Despite all of these different approaches and investigations, the slicing process is still fundamentally viewed as one step in a sequential process. Slicing is conducted, G-code is produced, and then the object is constructed. In the next section, we discuss the difference in fundamental design with ORNL Slicer 2.0 keeping in mind that any of these approaches could be utilized in combination with the proposed workflow.

## Paradigm Explanation and Software Design

ORNL Slicer 2.0 was designed with the concept of “on-demand” slicing. This on-demand model allows the slicer to be a more significant participant during the construction process as well as harness additional capabilities from other sources. Essentially, “on-demand” slicing is the capability of the slicing package to produce G-code on a layer-by-layer basis rather than a monolithic block of G-code. By providing G-code one layer at a time, the ability to close and incorporate feedback loops from the construction process becomes possible. The difference in slicing workflows is shown in

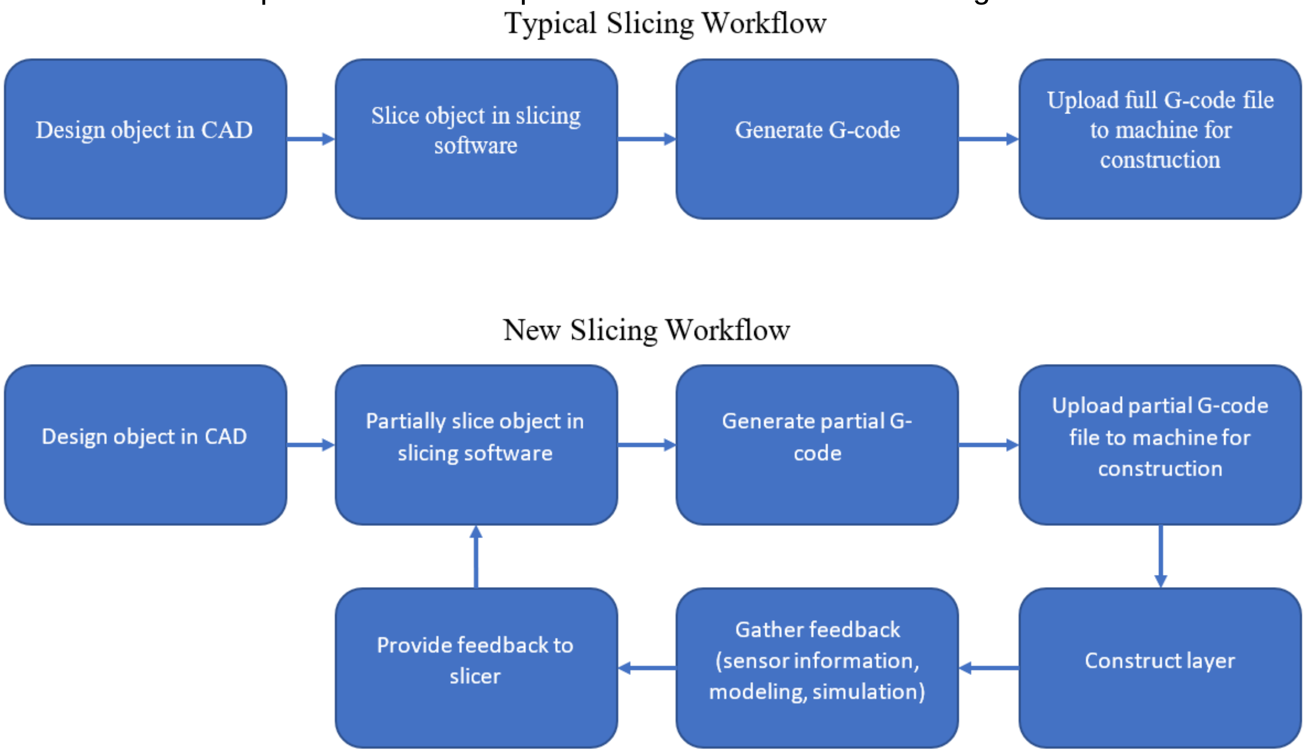


Figure 2.

*Figure 2: Top: Typical slicing workflow – a sequential process that moves from object design to construction. Bottom: Proposed slicing workflow – once object design is complete, the slicing software provides G-code on demand while incorporating feedback during layer construction*

As mentioned previously, the typical slicing workflow is one of sequential steps. Slicing is a clearly defined step that produces the necessary G-code for construction and provides no additional construction support. These approaches have typically been based solely on an evaluation of the geometry to be constructed. While successful, this approach does have its limitations. Primarily, this workflow has the underlying assumption that construction will exactly mirror the pathing previously laid out by the slicing software.

This assumption has been shown incorrect, and a great deal of work has been spent on developing open-loop feedback concepts. These concepts are varied and include items such as: sensor feedback, simulation and modeling, and machine learning. The goal of this body of work is the recognition that the construction process will not be ideal. Feedback is incorporated during the construction process to help with object defects or process parameters with the ultimate goal being successful object construction. By incorporating this feedback, pathing moves beyond an analysis of the geometry for slicing.

However, this open-loop feedback has limitations. Primarily, the issue is that the slicer is no longer involved in the pathing process. For example, if sensor feedback was to be collected after every layer and such feedback showed pathing deviations from the ideal, the best that can be done is modification of subsequent G-code. Ideally, this sensor information would be fed back into the slicer in order to inform the pathing for subsequent layers and the slicer would replan accordingly potentially producing an entirely new pathing solution. By feeding this information back to the slicer, the feedback loop can be closed. This closed feedback loop can then be used to enhance the already existing capabilities of the slicer instead of attempting to make adjustments in an additional software package as is often the case with open loop solutions.

Closing this feedback loop and expanding beyond simple geometric evaluations is at the heart of the new slicing workflow. The new workflow shown in Figure 2 is a simple example of the power of "on-demand" slicing. In this new workflow example, the slicer provides G-code one layer at a time to the AM system. After each layer, the system utilizes sensor capabilities to inspect the object. This information is then fed back into the slicer so that subsequent layers can be informed by the deviations from the ideal pathing. With this additional information, the slicer software can then perform path planning as appropriate.

This on-demand capability was also paired with simple caching techniques to reduce computation time. The caching techniques allow Slicer 2.0 to recompute only those layers that should be altered. For example, if feedback indicates that the next layer needs to have pathing adjusted, there's no reason to recompute other layers. A secondary benefit of the on-demand slicing and caching is scalability.

Consider the case when an object is sufficiently path dense. That is, there are potentially thousands of layers with tens of millions of vectors to construct the object. This could either be due to high object complexity, sheer size of the object, or an additive manufacturing system that has very high resolution. In addition, as previously mentioned, there is significant interest in moving beyond simple geometric analysis for path planning in additive manufacturing. When these approaches are considered wholistically, there is a significant computational burden.

In this situation and moving forward with the ever-increasing collection of data spurred by the Industry 4.0 evolution, even GPU-based algorithms will require a reasonable amount of time to compute the appropriate pathing. Over the course of hundreds or thousands of layers, this process can easily take hours to complete and potentially represent a significant computational cost. However, this can be partially addressed with the proposed workflow. Because the slicing process is now an active participant in the build process, pathing requirements are reduced to a single layer at a time. The pathing process can now be interleaved with the construction process allowing a significant overlap in time that hides the computational cost.

To be clear, the entire computation cost cannot be hidden. If the pathing from a layer is dependent upon sensor collection during the previous layer's construction, this processing must take place after construction. However, in many cases, this evaluation results in only minor modifications if the process generally stays nominal. As a result, the typical geometric evaluation to lay out the pathing can be done in parallel during object construction and then modified by a lightweight post-processing step that incorporates feedback.

## Examples

Closing the feedback loop through the inspection of the construction process as it happens was one of the key motivators in the design of ORNL Slicer 2.0. With a supportive workflow in place, multiple processes can be augmented with slicer support. In this section, a few examples of processes that could provide additional pathing input will be described. The first example will concern the use of sensors in the form of thermal cameras and laser profilometers. Previous work has been conducted using both types of sensors [21, 22, 23].

In the first work, a laser profilometer was used to ensure appropriate layer heights in a large-scale metal DED system [21]. To achieve these results, the laser profilometer was passed over the object after each layer. The pathing was a simple raster pattern shown in Figure 3 and was automatically generated as part of the slicing process. In addition, “ideal” representations were generated as part of the slicing process. This ideal information was constructed as part of the cross-sectioning step in the slicing process and provided a 2D polygon overlay used as part of the image analysis. This ideal representation defined boundaries and expected heights of the object for a specific layer.

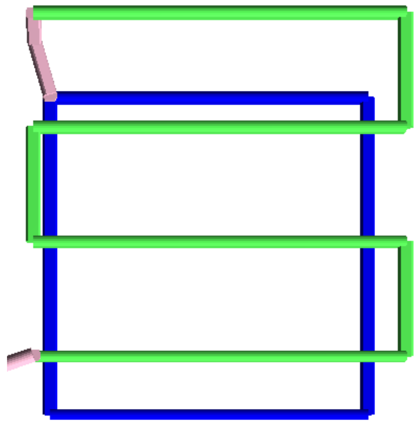

*Figure 3:Example of automated rasterization pathing for laser profilometer support. The pink line represents a travel to begin the path, followed by the green lines representing the scan path, and finally the blue lines representing construction.*

This pathing and ideal representations were combined with data collected by the laser profilometer after each layer construction. By comparing the data collected by the laser profilometer to the ideal representations, a height map describing positive/negative offsets was constructed. This height map was then used in the construction of the subsequent step via interpolation. An example of a height map with pathing overlay is shown in Figure 4.

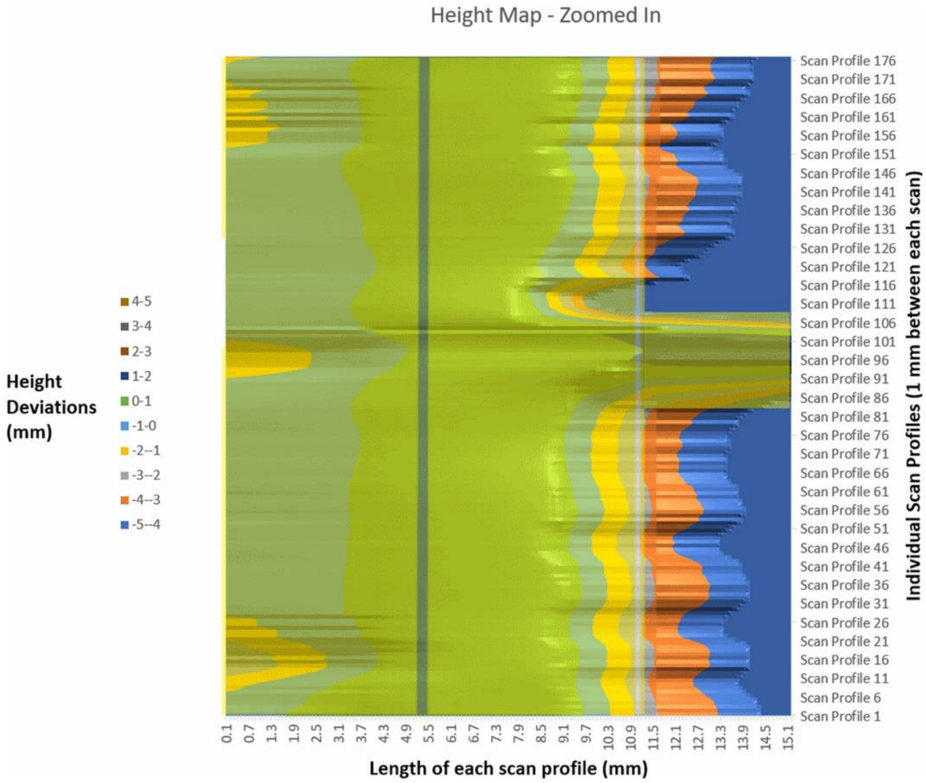


*Figure 4: Example height map with highlighted overlay showing the pathing for the next layer along with height deviations as gathered by the laser profilometer. (Borish et al. [21])*

Based on this example, the pathing for the next layer would require modification. The path on the left is, on average, lower than expected. As a result, the system would need to increase material deposition to balanced out the deviation. An in-depth analysis can be found in Borish et al. [21].

The process achieved good results and an example object was constructed to showcase dynamic height control. The example object is shown in Figure 5.

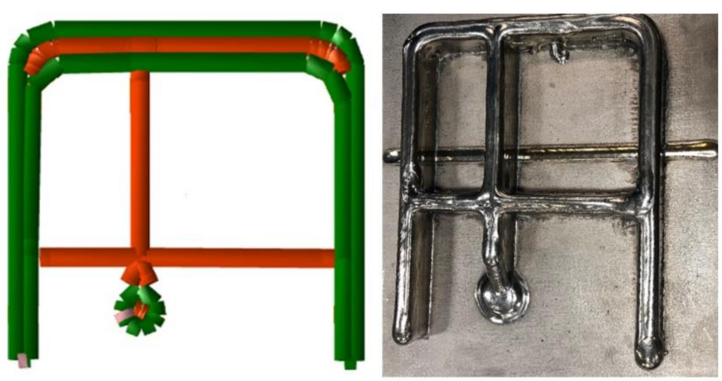

*Figure 5: Example object constructed with automated profilometer support for height control. The system generated a height map after each layer construction. This height map was then used to adjust material flow for the subsequent layer. Left: Pathing from slicer. Right: Final*

Similar work was conducted using a large-scale polymer system as well [22, 23]. In this work, the setup and procedure were identical to the previous metal-based case. The slicing software again automatically generated the appropriate ideal information and raster patterns. However, rather than simply maintaining appropriate layer heights, the system dynamically identified and mitigated defects.

In this example, the system adjusted for both underfill and overfill defects. A simple example of underfill and the resulting mitigation is shown in Figure 6.

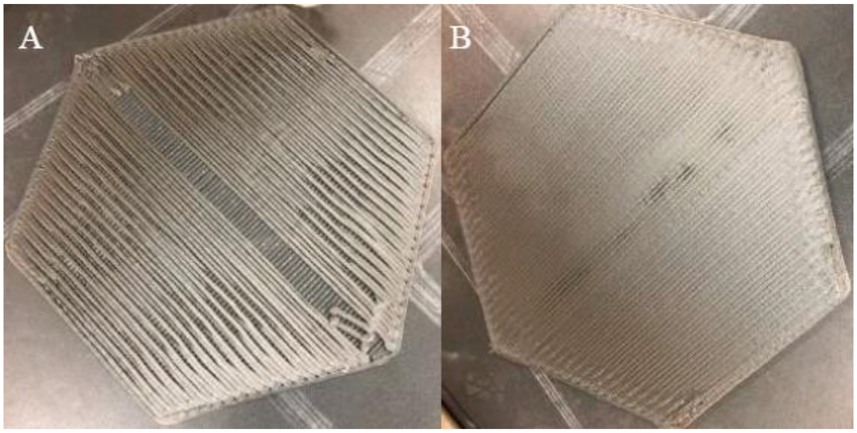


*Figure 6: Example object constructed with automated profilometer support for underfill and overfill defect correction. The system generated a height map after each layer construction. This height map was used by subsequent layers to adjust construction speed and material flow to correct for defects. Left: Layer with underfill defects. Right: Same object three layers later with defects corrected. (Borish et al. [23])*

To adjust for these defects, the system dynamically adjusted both material flow and construction speed to maintain appropriate process parameters. This capability was then combined with a thermal camera sensor to monitor layer temperatures. Just as in the case of the profilometer, the thermal camera integration was supported as part of our slicer development. For this system, simple pathing to move the gantry out of the way so as to provide the camera with an unobstructed top-down view was implemented. An example of this pathing is shown in Figure 7.

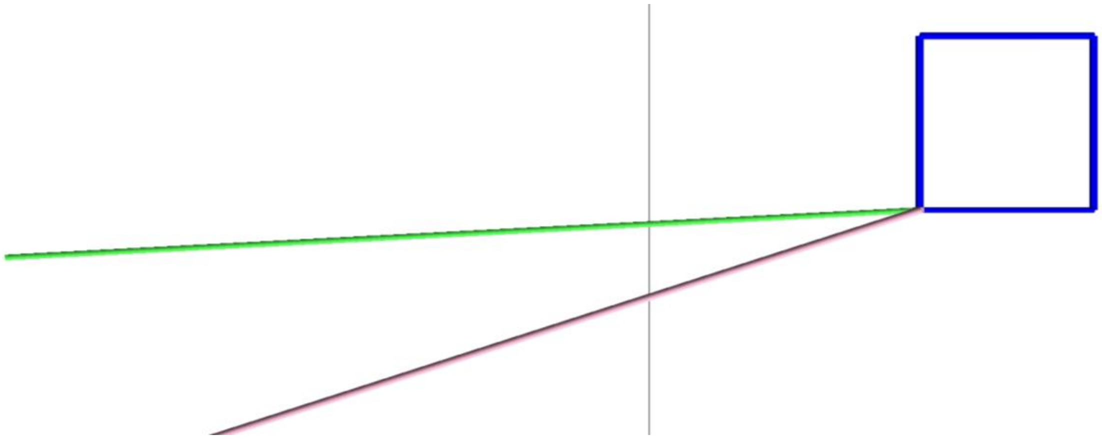

*Figure 7: Example of automated pathing for thermal camera support. The pink line indicates travel to the object, followed by blue lines for perimeter construction, and ends with the green line representing the motion to begin data collection*

This work sought to control layer times so as to provide appropriate time for layers to cool when objects had low layer times. Without this additional cooling time, the material would not be under the glass transition temperature and would eventually collapse on its own weight. A before-


*Figure 8: Example object constructed with support of thermal camera. After each layer construction, the thermal camera was extended to provide a top-down view to monitor layer temperatures. Construction speed was modulated to allow enough time for the object to reach its glass transition temperature. Top: without thermal camera support. Bottom: with thermal camera support. (Borish et al. [22])*

and-after example is shown in Figure 8.

In addition to closing the sensor feedback loop, the active design of ORNL Slicer 2.0 also allows the application of modeling information. This modeling information can be used for a variety of purposes. Increasing amounts of work are focusing on the use of information beyond simple geometry to inform path planning. One of the more straightforward examples is the use of finite element analysis (FEA). The proposed slicing paradigm fits with the use of FEA as well as sensors.

Figure 9 shows an example of automatic application of FEA information to the pathing in question. In this example, an FEA was computed for stress fields on a simple object. The hypothetical grid from this analysis was then overlaid on top of the pathing for each layer. The stress analysis was used to modulate the density of the resulting infill pattern automatically based on user defined thresholds. So, in general, where there were high stress areas on the object, the material was denser to provide the necessary support. Conversely, if there was little stress, the area would be filled with less dense material.

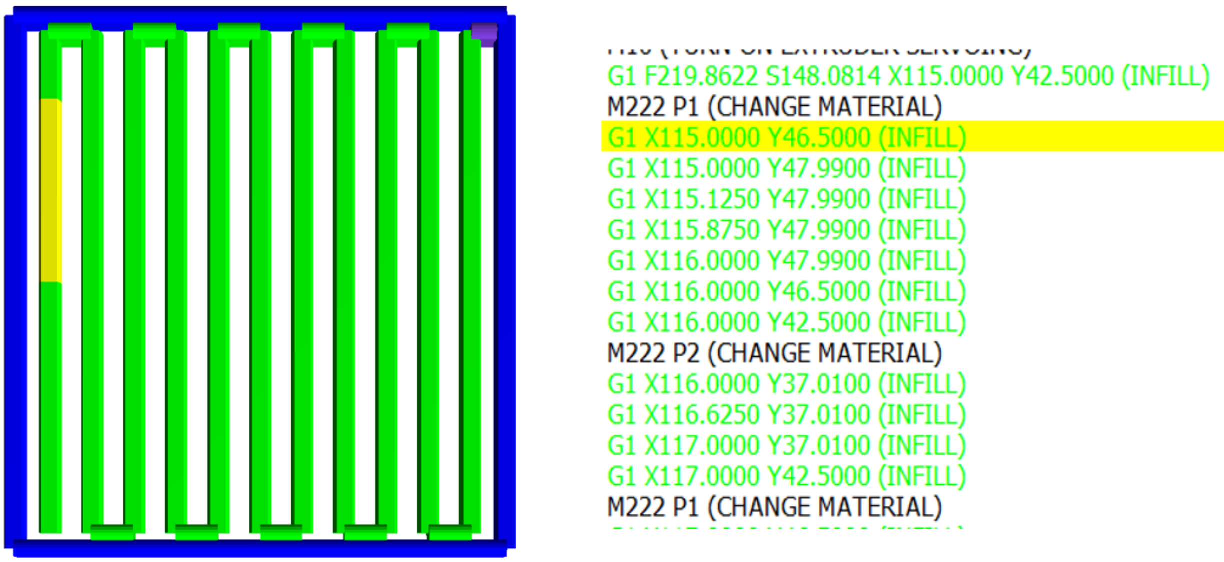


*Figure 9: Example pathing based on FEA. Slicer 2 automatically applies the virtual simulation grid from the analysis to the pathing as a post-processing step. Blend change commands (M222) happen at locations that intersect simulation grid and selects material density based on user-defined thresholds.*

The system in question for this example utilizes a specific M-code to change recipes during the construction process. A recipe is a pre-configured material blend consisting of various percentages of multiple materials. Upon interpolating into the FEA matrix, blend change commands were then inserted to modify the path as appropriate. While this approach is powerful, there is no reason that this approach could not be combined with the previous examples utilizing sensor feedback. With this new workflow that provides per-layer G-code, sensor feedback could be ingested to inform subsequent layers' FEA.

## Conclusion

In this paper, we presented a new workflow for the slicing process based on the design of ORNL Slicer 2.0. In this workflow, the slicer can provide G-code on-demand on a layer wise basis. With such a workflow, the slicer becomes an active participant in the construction process. This contrasts with the conventional utilization of a slicer where slicing is a single step in a sequential process.

There are several benefits to this new design. One of which is scalability as the slicer needs to provide pathing for only a single layer as opposed to the entire object. The computation of that pathing can be done in parallel with the construction process. This allows a significant amount of the computation cost to be hidden which will become increasingly important as pathing moves beyond simple geometric analysis. More important is the ability to close the feedback loop of the construction process with the slicer. Information from sensors and modeling can be fed into the slicer in order to replan subsequent pathing as necessary.

We intend to pursue additional work in this area as the design has only scratched the surface of what is possible with a slicer that is active during the construction process. We will be leveraging this design to expand data feedback to provide informed pathing. We also intend to allow connections between the slicer and other software packages.

## Acknowledgments

This material is based upon work supported by the U.S. Department of Energy, Office of Energy Efficiency and Renewable Energy, Office of Advanced Manufacturing, under contract number DE-AC05-00OR22725.

## Bibliography

[1] Dalenogare, Lucas Santos, et al. "The expected contribution of Industry 4.0 technologies for industrial performance." International Journal of Production Economics 204 (2018): 383-394.
[2] Love, Lonnie, et al. Commercialization of Big Area Additive Manufacturing. No. ORNL/TM-2020/1454. Oak Ridge National Lab.(ORNL), Oak Ridge, TN (United States), 2020.
[3] Gibson, Brian, et al. Melt pool monitoring for control and data analytics in large-scale metal additive manufacturing. Oak Ridge National Lab.(ORNL), Oak Ridge, TN (United States), 2019.
[4] Panchagnula, Jayaprakash Sharma, and Suryakumar Simhambhatla. "Feature based weld-deposition for additive manufacturing of complex shapes." Journal of The Institution of Engineers (India): Series C 99.3 (2018): 285-292.
[5] Peng, Hao, et al. "Fast prediction of thermal distortion in metal powder bed fusion additive manufacturing: Part 1, a thermal circuit network model." Additive Manufacturing 22 (2018): 852-868.
[6] Liu, Jikai, Jingjing Yan, and Huangchao Yu. "Stress-constrained topology optimization for material extrusion polymer additive manufacturing." Journal of Computational Design and Engineering 8.3 (2021): 979-993.
[7] Li, Shaoying, et al. "Multidisciplinary topology optimization incorporating process-structure-property-performance relationship of additive manufacturing." Structural and Multidisciplinary Optimization 63.5 (2021): 2141-2157.
[8] Koepf, Johannes A., et al. "3D multi-layer grain structure simulation of powder bed fusion additive manufacturing." Acta Materialia 152 (2018): 119-126.
[9] Liu, Chao, et al. “Digital twin-enabled collaborative data management for metal additive manufacturing systems.” Journal of Manufacturing Systems (2020).
[10] Manoharan, Madhanagopal, et al. "A Novel Volume Decomposition Methodology for Multi-Robots Collaborative Additive Manufacturing." 2020 IEEE 4th Conference on Information & Communication Technology (CICT). IEEE, 2020.
[11] Djuric, Ana, and R. Jill Urbanic. "Using collaborative robots to assist with travel path development for material deposition based additive manufacturing processes." Computer-Aided Design and Applications 15.4 (2018): 542-555.
[12] Jace McPherson and Wenchao Zhou, (2018). A chunk-based slicer for cooperative 3D printing", Rapid Prototyping Journal.
[13] Roach, Devin J., et al. "The m4 3D printer: A multi-material multi-method additive manufacturing platform for future 3D printed structures." Additive Manufacturing 29 (2019): 100819.
[14] Michel, Florent, et al. "A modular path planning solution for Wire+ Arc Additive Manufacturing." Robotics and Computer-Integrated Manufacturing 60 (2019): 1-11.
[15] Mao, H., Kwok, T. H., Chen, Y., & Wang, C. C. (2019). Adaptive Slicing based on efficient profile analysis. Computer-Aided Design, 107, 89-101.
[16] Zhang, Xipeng, et al. A GPU-based parallel slicer for 3D printing. 2017 13th IEEE Conference on Automation Science and Engineering (CASE). IEEE, 2017.
[17] Rebaioli, Lara, et al. Process parameters tuning and online re-slicing for robotized additive manufacturing of big plastic objects. Robotics and Computer-Integrated Manufacturing 55 (2019): 55-64.
[18] Ogayar-Anguita, C. J., Rueda-Ruiz, A. J., Segura-Sánchez, R. J., Díaz-Medina, M., & García-Fernández, Á. L. (2020). A GPU-Based Framework for Generating Implicit Datasets of

Voxelized Polygonal Models for the Training of 3D Convolutional Neural Networks. IEEE Access, 8, 12675-12687.

[19] Yigit, Ismail Enes, and I. Lazoglu. "Spherical slicing method and its application on robotic additive manufacturing." Progress in Additive Manufacturing 5.4 (2020): 387-394.

[20] Lefebvre, Sylvain. IceSL: A GPU accelerated CSG modeler and slicer. 18th European Forum on Additive Manufacturing (AEFA'13). 2013.

[21] Borish, Michael, et al. "Defect identification and mitigation via visual inspection in large-scale additive manufacturing." JOM 71.3 (2019): 893-899.

[22] Borish, Michael, et al. "In-situ thermal imaging for single layer build time alteration in large-scale polymer additive manufacturing." Procedia Manufacturing 34 (2019): 482-488.

[23] Borish, Michael, et al. "Real-Time Defect Correction in Large-Scale Polymer Additive Manufacturing via Thermal Imaging and Laser Profilometer." Procedia Manufacturing 48 (2020): 625-633.