Simulation of Polymeric Composites Additive Manufacturing using Abaqus

Anthony J. Favaloro, Bastian Brenken, Eduardo Barocio, and R. Byron Pipes

School of Aeronautics and Astronautics, Purdue University, West Lafayette, IN, US

***Abstract**: Additive manufacturing, specifically the extrusion deposition process, involves the progressive addition of material at elevated temperatures following a prescribed machine path at prescribed speed. In order to properly simulate the thermal history and resulting mechanical deformations and stresses, this transient addition of material must be captured. Utilizing the newly available features of Abaqus 2017, specifically element activation and event series, progressive element activation is performed by implanting the user subroutine UEPActivationVol. In this work, the system at Purdue University for extrusion deposition of highly filled, high temperature thermoplastics is modeled through a user subroutine suite which coordinates element activation according to machine instructions, assigns appropriate local coordinate systems for using anisotropic material properties, and sets relevant initial state variables for user material models. Simple structures are modeled and printed to calibrate convection and radiation properties. A case study of printing an air inlet duct tool is presented showing the full part simulations using a voxel mesh predicting both in process deformations and stresses as well as deformations and stresses following a springback analysis representing part removal from a build plate.*

***Keywords**: Abaqus, Abaqus/Standard, additive manufacturing, 3d printing, composites, composite materials, UMATHT, SDVINI, ORIENT, UEXTERNALDB, UEPActivationVol, UEXPAND.*

1. Introduction

Additive manufacturing (AM) is promising technology for producing complex shapes quickly with minimal waste. AM with polymeric composite materials provide the potential for increased stiffness and strength versus neat polymers. The addition of carbon fibers enables scaling the technology to the production of large parts due to the stiffness increase and thermal expansion decrease reducing warping issues often encountered in AM (Love, 2014). Extrusion deposition (EDAM) is one AM method which can directly utilize polymeric composites as feedstock materials. However, the addition of carbon fibers also complicates analysis as the printed material is anisotropic due to flow alignment of fibers in the extrusion process (Heller, 2016). Unfortunately, through thickness strength of EDAM parts is a remaining issue due to the layer by layer process. Therefore, initial applications for this technology are stiffness critical only such as prototype tooling for other processes both in low temperature applications [3] (Hassen, 2016) (Sudbury, 2016) and elevated temperature applications [5](Hassen, 2016) (Kunc, 2016). Simulations predicting residual stresses as well a final warped shape are important to aid the tool design process and ensure tolerances.

A bulk of previous work in simulating the EDAM process sought to understand the process on simple geometries essentially attempting to decouple the process from the realistic printed geometries. For example, Zhang and Chou investigated the warping effects of varying raster pattern to produce simple cuboids (Zhang, 2006) as well as the residual stress effects of varying scan speed, layer thickness, and deposition width (Zhang, 2008). Costa et al. attempted to determine which heat transfer phenomena are most impactful and necessary to capture (Costa, 2015). While these studies provide some insight and assist in establishing an intuition on the process. Similar to welding, the physical phenomena cannot, in general, be decoupled from the specific geometry considered.

In a combined effort, AlphaStar and Oak Ridge National Labs present a full simulation of the printing process of a Strati, 3D printed car (Talagani, 2015). The simulation was used to identify potential areas of in-process delamination between layers due to stress accumulation resulting in process adjustments to minimize these issues. While the simulation is relatively comprehensive, no attempt is made to develop a general framework in which many parts could be simulated quickly.

In this work, we present a framework developed in Abaqus for simulating the EDAM process that utilizes UEPActivationVol, a new user subroutine available in Abaqus 2017. Simulation of a new part using the same printer and material properties in this framework requires minimal modification of the input deck to point towards a new machine code file, in the form of an event series, as well as a new mesh file which is automatically generated from the machine code using a Python script. Element activation timing and element orientations are automatically determined through a single execution of the subroutine ORIENT per element greatly increasing computational speed compared to implementations which check for activation every time increment.

2. Extrusion Deposition Simulation Requirements

The EDAM process is also known as material extrusion, fused deposition modeling (FDM), and fused filament fabrication (FFF). However, FDM and FFF are commonly associated with processes that begin with a filamentized material. Extrusion deposition is a more general term encompassing processes that begin with a filamentized material or pelletized material.

We will be considering the custom designed and build Composites Additive Manufacturing Research Instrument (CAMRI) at Purdue University (see Figure 1). In this machine, pelletized material is fed through a single screw extruder with a 5 to 10 lb/hr output rate. The material is then extruded through a converging nozzle and finally onto a 3 axis motion table which carries out the appropriate motions to build a 3D shape through bead by bead, layer by layer deposition of material in a build volume of approximately 1.5 ft x 1.5 ft x 1.5 ft. Figure 2 shows a simple example of this process showing a newly printed bead or road being deposited onto a previously existing arrangement of beads.

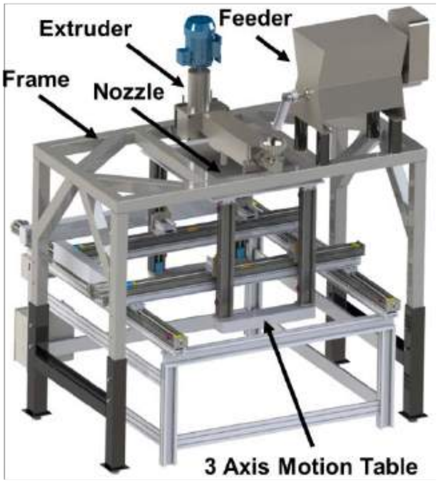


Figure 1. Composites Additive Manufacturing Research Instrument

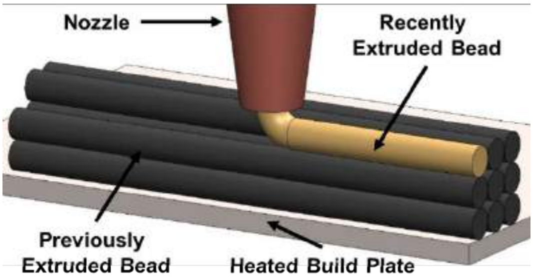


Figure 2. Simple Extrusion Deposition Diagram

Our material of interest is 50 weight percent carbon fiber reinforced polyphenylene sulfide, or 43vol% CF-PPS. This PPS reaches a maximum relative crystallinity of 84% in the polymer, approximately 48vol% of the composite. As a fiber reinforced material, the thermal and mechanical response is anisotropic dependent on the orientation state of the microstructure. In general, flowing the material through the converging zone of the nozzle causes a large degree of alignment along the bead length (Heller, 2016). Additionally, compressive forces from the nozzle or external consolidation mechanisms results in beads with cross sections which are wider than tall resembling ellipses cut off by a bounding rectangle (see Figure 3). In order to consider simulations of full printed parts rather than small sections, we must consider homogenized properties. In the bead coordinate system with x_1 representing the length direction, x_3 representing the stacking direction, and x_2 representing the transverse in plane direction, the printed material behaves as an orthotropic material (Favaloro, 2016).

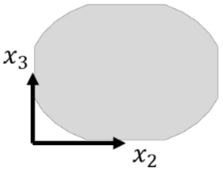


Figure 3. Example Bead Cross Sectional Geometry

PPS is a semi-crystalline polymer. This means that when cooling from the melt, PPS undergoes crystallization through the processes of crystal nucleation and crystal growth. This phase change is exothermic and causes the material to shrink. Specifics of the user material subroutines developed to determine crystallinity as well as apply crystallization eigenstrains are discussed in a companion paper by Brenken (Brenken, 2017).

With this machine and material in mind, we have the following requirements for a simulation of the extrusion deposition process:

- Progressive activation of elements in a model based on machine path information.
- Automatic determination of local coordinate systems based on machine path information.
- A material model which can account for crystallization development and shrinkage.

3. Implementation Details

We coordinate the requirements of the previous section using a user subroutine suite which includes UEXTERNALDB to create global arrays, ORIENT to assign local material coordinate systems, SDVINI to assign necessary state variables for crystallization calculations, UEPActivationVol for progressive element activation, UMATHT for orthotropic heat transfer as well as crystallization calculations in thermal analyses, and a UEXPAND for crystallization calculations in static analysis (using input temperature histories from a thermal analysis). Machine path information is communicated to the user subroutines through the use of the new event series functionality which allows for fast determination of segments of machine path nearby to a query point (e.g. element centroid).

The primary algorithm we are concerned with is using the machine code to determine element activation times as well as local orientations. This algorithm will be executed for every element in a potentially printed domain. We begin with knowledge of the elements centroid location, x_i^C, as well as the full machine path information, and a globally defined stacking direction, x_i^S. From this centroid location, we search outward in a sphere for event series segments which are either fully contained within the sphere or intersect the sphere. This search is performed using the event series API, getEventSeriesLGLocationPath which quickly performs this search. The radius of the sphere we use for this search is defined in Equation 1 where r is the search radius, r_f is a small positive adjustment factor on this radius, w_{max} is the maximum possible bead width, and h_{max} is the maximum possible bead height. The extra factor of $5/4$ accounts for the possibility of activation beyond the end of an event series segment as will be discussed.

$$r = \left(1 + r_f\right)\sqrt{(5/4)w_{max}^2 + h_{max}^2} \quad [1]$$

For each segment found by this search in which the extrusion flag is on at the start of the segment, we let t^A be the start time, t^B be the end time, x_i^A be the starting location, and x_i^B be the ending location. With this information, we can define the location event series coordinate system defined by the principal directions X_i^1, X_i^2, and X_i^3 as shown in Equation 2. The length of the event series is then $L^{AB} = \|x_i^B - x_i^A\|$.

$$\begin{aligned} X_i^1 &= \frac{x_i^B - x_i^A}{\|x_i^B - x_i^A\|} \\ X_i^2 &= \frac{\varepsilon_{ijk} x_j^S X_k^1}{\left\|\varepsilon_{ijk} x_j^S X_k^1\right\|} \\ X_i^3 &= \varepsilon_{ijk} X_j^1 X_k^2 \end{aligned} \quad [2]$$

Next, we determine the elements location based on this coordinate system with origin at x_i^A giving segment coordinate system based coordinates x^1, x^2, and x^3.

$$\begin{aligned} x^1 &= (x_i^C - x_i^A)X_i^1 \\ x^2 &= (x_i^C - x_i^A)X_i^2 \\ x^3 &= (x_i^C - x_i^A)X_i^3 \end{aligned} \quad [3]$$

If x^1 is negative, then the elements centroid is behind this event series segment, and we can immediately dismiss it. We determine a bead width for this segment assuming a constant height throughout the entire model of h using Equation 4 where w^m and V^m are the nominal or mean bead width and machine linear speed. An alternate functional form could be used provided more information of machine dynamics is known. With this expression we can account for variability in machine linear speed (i.e. the machine moving slower prints wider beads and faster prints thinner beads assuming constant). Of course, more complicated expressions could be used to account for specific machine dynamics considering extrusion rate control.

$$w^{AB} = w^m \frac{V^m}{L^{AB}}(t^B - t^A) \quad [4]$$

If x^1 is larger than $L^{AB} + w^{AB}/2$, then the element's centroid is too far in front of the event series segment to be relevant. If x^1 is less than L^{AB}, we have two final comparisons to make. If

$|x^2| \leq w^{AB}/2$ then the element centroid is sufficiently within the width of the bead, and if $0 \leq x^3 \leq h$ then the element centroid is sufficiently within the height of the bead. Of course, now that we have determined local event series coordinates any shape can be used as the beads cross section. Here, we choose to use the simple case of a rectangle.

Finally, if the extrusion flag is on at the end of the segment and $L^{AB} \leq x^1 \leq L^{AB} + w^{AB}/2$, then we consider the possibility that an element is missed by the piecewise linear nature of the event series data. An example of this can be seen in Equation 4 below in which the red box representing an event series segment and the blue box representing the next event series segment miss an element centroid (represented as black dots) that is captured by the green circle. To account for this we append a capping circle to end of each event series segment. We make the same width and height comparisons as before except in this case, we use $w^{CAP} = \sqrt{(w^{AB})^2 - 4(x^1 - L^{AB})^2}$ instead of the bead width.

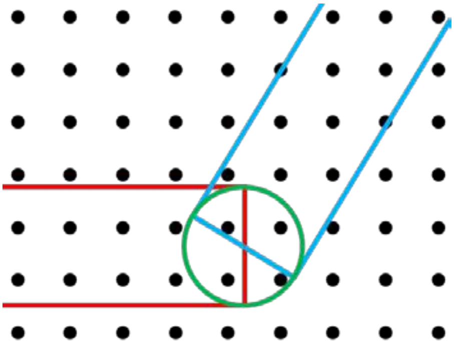
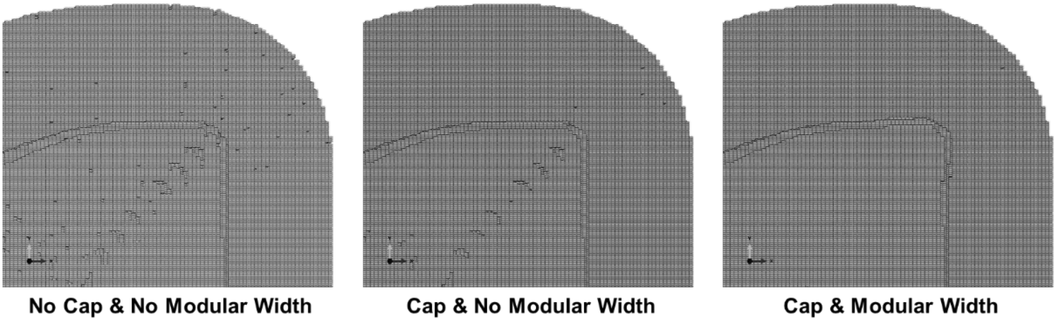


Figure 4. Activation Bounding Box with Capping Circle, Top View

Provided, the length, width, and height comparisons are satisfied, we activate the element at $t^{Act} = t^A + (t^B - t^A)x^1/L^{AB}$ with local orientations X_i^1, X_i^2, and X_i^3. We then return the information from the event series segment with the smallest t^{Act} for a given element. Figure 5 shows an example activated mesh for using this algorithm in three different cases: first without a capping circle or modular width, second with a capping circle and without modular width, and third with both a capping circle and modular width. More exact enhancements are possible such as linearly varying width between two ends of an event series segment, but any additional enhancement would require an additional set of information to be appended to the machine code.



No Cap & No Modular Width

Cap & No Modular Width

Cap & Modular Width

Figure 5. Comparison of Activated Mesh with and without Enhancements

With this algorithm developed, an efficient implementation must be devised. The crystallization behavior captured in the UMATHT or UEXPAND requires the activation time to initialize the integral expressions tracking the phase change. A naïve implementation would perform the checks in this algorithm for each subroutine which requires the information, ORIENT for local coordinates (initial time, per integration point call), SDVINI for activation time for UMATHT or UEXPAND (initial time, per integration point call), and UEPActivationVol (every time increment, per element call). This is essentially 16 (for 8 noded brick elements) calls of the algorithm per element at the initial time increment and one call of the algorithm per element every time increment. Additionally, SDVINI does not provide the algorithm element centroid information. Only the coordinates of a single integration point are made available. Therefore, adjustments to the algorithm would be necessary.

A more elegant approach is to perform the algorithmic check only once and store all the gained information in a global array which can be pre-allocated in UEXTERNALDB. The question remains of which user subroutine executes first and contains the necessary information (mainly element centroid). Fortunately, UEXTERNALDB executes prior to the analysis, ORIENT for the first integration point of an element executes before any other integration point, and SDVINI for an integration point executes following the execution of ORIENT. Additionally, ORIENT provides element centroid information. Therefore, we can use the call of ORIENT for an element's first integration point to execute the algorithm and simply store the activation time and local orientations in a global array. Calls of ORIENT for subsequent integration points, all calls to SDVINI, and calls to UEPActivationVol use the global array without re-execution of the searching algorithm. This form of the implementation both reduces initial setup time as well as per increment execution times. Figure 6 shows a basic flow chart of how the information is transferred between the relevant user subroutines. Additionally, to further decrease computation time, a Python script has been developed which builds a voxelized mesh using the same machine code file and algorithm so that only elements that will be activated are considered in the simulation and resulting in faster execution and smaller output databases.

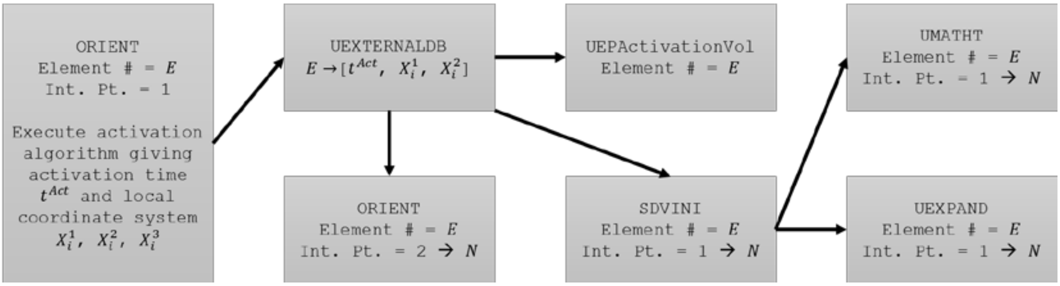


Figure 6. User Subroutine Suite Interconnectivity

4. Air Inlet Duct Tool Printing Simulation

As a case study, we consider the printing of the tool used to produce an air inlet duct as seen in Figure 7. This type of part is typically made with a continuous carbon fiber epoxy system and cured in an autoclave on a one sided tool. The 43vol% CF-PPS used in the CAMRI system has a melting temperature about 250°C which is well above the autoclave temperature of 180°C. Therefore, a printed composite tool can be used in contrast to traditional metal tools. Figure 8 shows such a printed tool with rough surfaces been machined smooth in a finishing step.

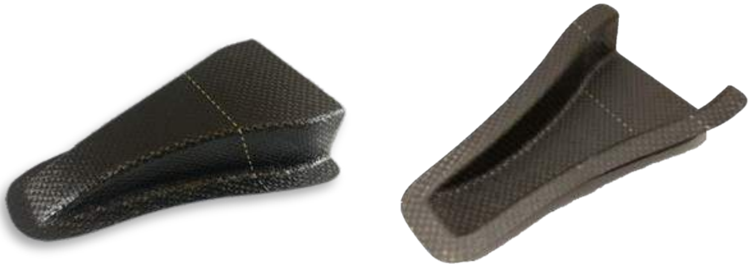


Figure 7. Air Inlet Duct

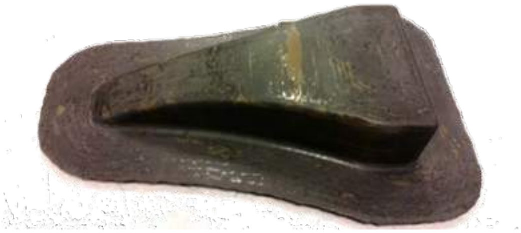


Figure 8. Printed and Finished Air Inlet Duct Tool

For an initial simulation of the printing process of this tool, we use a temperature independent, orthotropic elastic and thermal material definition. Additionally, we are using a calibrated phenomenological model for crystallization kinetics discussed in (Brenken, 2017) to determine the transient crystallization state. Our machine path information for this simulation is determined directly through sampling the microcontroller of the CAMRI system so that machine dynamics are captured. As our bead is approximately 3.4 mm wide and 2.3 mm tall, a voxel mesh is generated with elements of height 2.3 mm and in plane dimensions of 1.5 mm x 1.5 mm. Therefore, we should be able to capture multiple elements within a single bead even when making passes through the mesh at an angle. Figure 9 shows the nodal temperatures a two time points during the printing process: at 8 minutes showing the sparsely filled interior and at 12 minutes after deposition is complete. The ability to adjust infill parameters is the primary motivation for using voxelized meshes rather than conformal meshes as often the exact nature of the infill is not known in a CAD geometry. Observing the process simulation, it can be seen that initially conduction into the build plate (bottom nodes set to 180°C while material is activated at 300°C) is dominant while as the process progress and more free surface is created, convection and radiation effects become dominant. This is particularly true of a composite system in which fibers only enhance conduction in the bead direction while transverse and thickness direction conductivities are small.

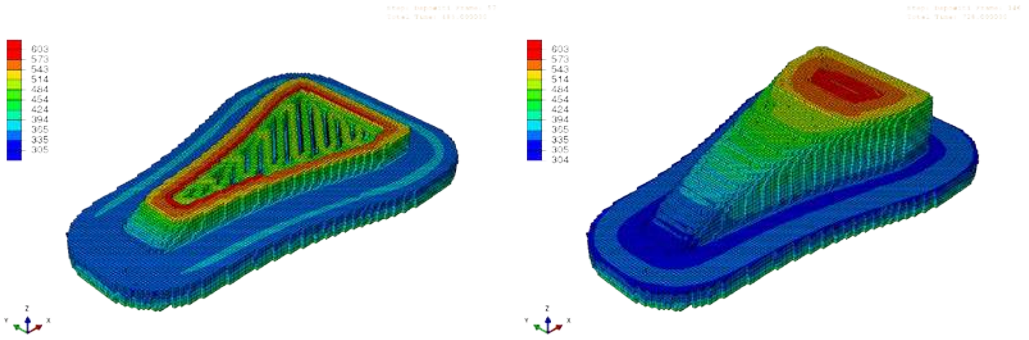


Figure 9. Nodal Temperatures in K during Process

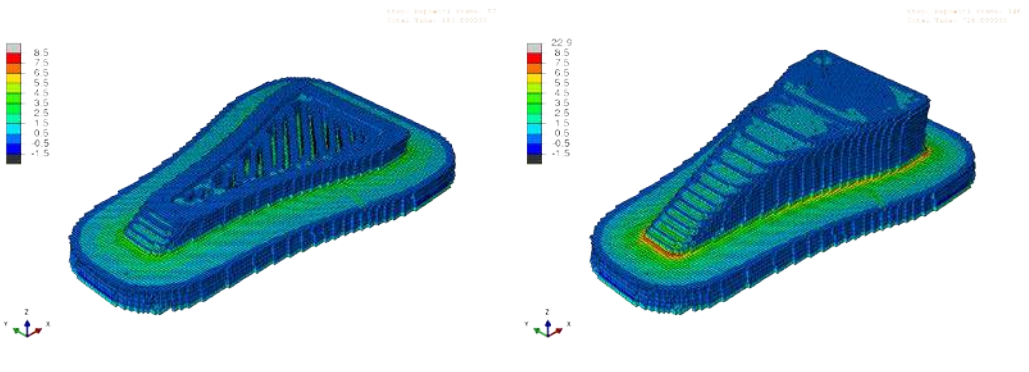


Figure 10 S22 Stresses in MPa during Process

Figure 10 shows the transverse stresses that occur during the printing process at equivalent time points to Figure 9. Significant transverse and through thickness stresses during the process can cause cracking in parts resulting in failed builds. Allowing the part to continue cooling, we determine the stresses shown in Figure 11 (left). We finally remove the mechanical boundary condition that the part is fixed to the build plate (fixed bottom nodes), and we determine the final shape and stress state of the part as shown in Figure 11 (right). During the spring back analysis, internal stresses relax leaving the part in a deformed shape with a residual stress state. An interesting observation which can be seen clearly in Figure 12 is that the resulting deformed shape as asymmetric. This is primarily due to the asymmetric infill pattern. This resulting asymmetry has been validated versus a printed part using the same machine path.

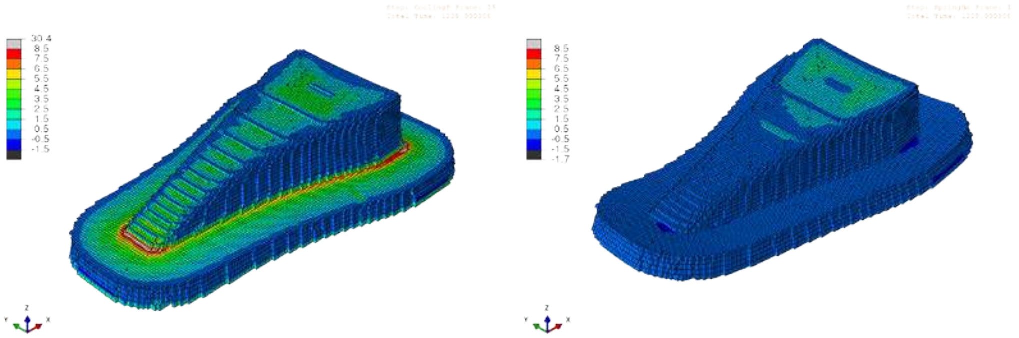


Figure 11. S22 Stress after Cooldown (left) and Springback (right)

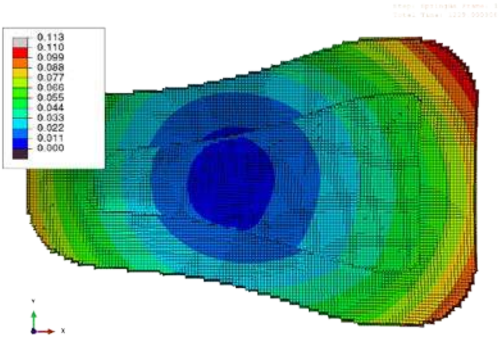


Figure 12. Top-Down View of Springback Deformation

Finally, Figure 13 shows a snapshot of the crystallization state of the part during the process. As can be seen in this image, fresh material has yet to begin crystallizing while several layers below the current layer, the material is almost fully crystallized (maximum crystallinity observed in this material is 0.84). Areas of larger surface area to volume ratios, specifically around the edge of the base of the part, did not reach full crystallinity. This implies that these regions cooled too quickly for the material to crystallize, locking the material in the amorphous state. These regions will have a lower melting temperature and lower mechanical properties than fully crystallized regions.

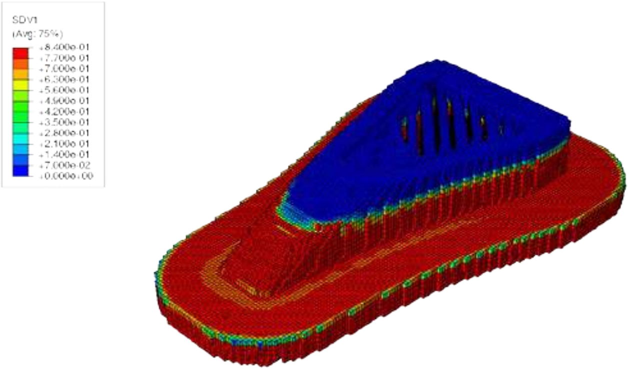


Figure 13. Crystallization Front during Process

5. Conclusions

In this work, we have presented a developed framework for simulation of the EDAM process using a custom user subroutine suite. Specifically, we have shown how the machine code is used in the form of an event series to determine element activation timing (for UEPActivationVol) and element orientations (for local orthotropic material properties) in an efficient manner minimizing redundant checks. This framework allows for the simulation of multiple parts produced by the same machine through simple interchange of machine code and automatically generated mesh files. An example simulation is presented for a duct tool geometry showing the transient thermal, deformation, stress, and crystallization states. The distortion predictions agree qualitatively in parts printed on the CAMRI. Future work involves characterizing and implementing the viscoelastic behavior of the composite material in order to study stress relaxation during the build as well as further relaxation or creep in use. Additionally, using a viscoelastic model, gravitational loading will be investigated as a major issue in printing with higher temperature polymers is sagging due to extended periods at elevated temperatures.

6. Acknowledgements

The authors would like to thank Victor Oancea, Subham Sett, Rani Richardson and the rest of the additive manufacturing efforts at Dassault Systemes for the additional capabilities in Abaqus, which greatly simplify the AM simulation effort, and continued support and workshop opportunities. This material is based upon work supported by the National Science Foundation Graduate Research Fellowship under Grant No. DGE-1333468.

7. References

1. Brenken, B., E. Barocio, A. Favaloro and R. B. Pipes, "Simulation of Semi-Crystalline Composites in the Extrusion Deposition Additive Manufacturing Process," in *Science in the Age of Experience by Dassault Systemes*, Chicago, IL, 2017.
2. Costa, S., F. Duarte and J. Covas, "Thermal conditions affection heat transfer in FDM/FFE: a contribution towards the numerical modelling of the process," *Virtual and Physical Prototyping*, vol. 10, no. 1, pp. 35-46, 2015.
3. Favaloro, A., B. Brenken, E. Barocio, N. DeNardo and R. B. Pipes, "Microstructural Modeling of FIber Filled Polymers in Fused Filament Fabrication," in *SAMPE Long Beach 2016*, Long Beach, CA, 2016.
4. Hassen, A. A., R. Springfield, J. Lindahl, B. Post, L. Love, C. Duty, U. Vaidya, R. B. Pipes and V. Kunc, "The durability of large-scale additive manufacturing composite molds," in *CAMX Anaheim 2016*, Anaheim, CA, 2016.
5. Hassen, A. A., J. Lindahl, X. Chen, B. Post, L. Love and V. Kunc, "Additive Manufacturing of Composite Tooling Using High Temperature Thermoplastic Materials," in *SAMPE Long Beach 2016*, Long Beach, CA, 2016.
6. Heller, B. P., D. E. Smith and D. A. Jack, "Effects of extrudate swell and nozzle geometry on fiber orientation in Fused Filament Fabrication," *Additive Manufacturing*, vol. 12, pp. 252-264, 2016.
7. Kunc, V., J. Lindahl, R. Dinwiddie, B. Post, L. Love, C. Duty, M. Matlack, R. L. Fahrey Jr. and A. A. Hassen, "Investigation of in-autoclave additive manufacturing composte tooling," in *CAMX Anaheim 2016*, Anaheim, CA, 2016.
8. Love, L. J., V. Kunc, O. Rios, C. E. Duty, A. M. Elliott, B. K. Post, R. J. Smith and C. A. Blue, "The importance of carbon fiber to polymer additive manufacturing," *Materials Research*, vol. 29, no. 17, pp. 1893-1898, 2014.
9. Sudbury, T., V. K. R. Sprinfield and C. Duty, "An assessment of additive manufactured molds for hand-laid fiber reinforced composites," *International Journal of Advanced Manufacturing Technology*, 2016.
10. Talagani, M. R., S. DorMohammadi, R. Dutton, C. Godines, H. Baid, F. Abdi, V. Kunc, B. Compton, S. Simunovic, C. Duty, L. Love, B. Post and C. Blue, "Numerical Simulation of Big Area Additive Manufacturing (3D Printing) of a Full Size Car," *SAMPE Journal*, vol. 51, no. 4, pp. 27-36, 2015.
11. Zhang, Y. and Y. K. Chou, "3D FEA Simulations of Fused Deposition Modeling Process," in *ASME International Conference on Manufacturing Science and Engineering*, Ypsilanti, MI, 2006.
12. Zhang, Y. and K. Chou, "A paramteric study of part distortions in fused deposition modelling using three-dimensional finite element analysis," *Proc. IMechE*, vol. 222, no. Part B, 2008.