Algorithm Development for Generation of Powder Particles for Use in Numerical Simulations of Powder Bed Fusion Additive Manufacturing Processes

Garrett M. Kelley and Mamidala Ramulu*
Department of Mechanical Engineering, University of Washington, Seattle, WA USA

*Corresponding author email: ramulum@uw.edu

Abstract. This article presents an algorithm development methodology for incorporating the effects of powder quality into numerical simulations of powder bed fusion technologies. The framework leverages x-ray microtomography measurements of samples representative of changes in powder quality with powder reuse. Individual particles are labelled and analysed by postprocessing the three-dimensional data. Included in this data is the particle surface normal and point cloud information which can be used to reconstruct the particle using Poisson Surface Reconstruction in Open3D. This reconstruction can then be exported as a Standard Tessellation Language (STL) file that can be incorporated into discrete element method solvers such as YADE or finite element frameworks such as MOOSE. In this sense, libraries of particles can be generated and used in simulations. To demonstrate the applicability of this method, a publicly available NIST dataset is leveraged. In lieu of experimental data, a second method using Computer-Generated Imagery (CGI) software, Blender, is also presented. The methods are able to replicate powder features, such as satellites and surface irregularity, that are functions of both powder manufacturing methods as well as powder reuse cycles. These particle irregularities have important implications on part quality and manufacturing cost.

Keywords: Additive manufacturing; Powder bed fusion; Particle morphology; Procedural generation; Phase field modelling.

1. Introduction

Additive Manufacturing (AM) technologies are distinguished from their Subtractive Manufacturing (SM) counterparts in that components are typically built “layer-by-layer” as opposed to chip-formed out of stock material. Driven by their advantages over traditional approaches such as increased part complexity, decreased startup capital expenditure (e.g., tooling and fixturing), and reduced scrap rate have led to an exponential increase in AM growth and market value [1]. Electron Beam Melting (EBM) is one such process that is receiving increased attention.

EBM is classified as a powder bed fusion (PBF) process in which layers of metallic powder are selectively fused using accelerated electrons [2]. While relatively simple in theory, the process is plagued by a lack of fundamental understanding as to how powder quality affects as-built part performance. This is because the build environment occurs under vacuum conditions and, for many alloys, exceeds temperatures of 700°C which makes in situ monitoring difficult [3]. Furthermore, the length and timescales at which the process occurs are not conducive to experimentation thereby necessitating the use of numerical models [4–6]. Given that the EBM process is time-intensive, with process cycles measured in days depending on build size, as well as contains a number of process parameters such as beam power and scan speed, all of which need to be optimized for a given powder
material, experimentation is expensive and difficult [7]. Therefore, there has been a push to develop process maps using numerical simulations to cut optimization costs [8,9]. Nevertheless, many of these models are missing one crucial aspect of powder quality: Morphology. Instead, particles are treated as notionally spherical which may be an oversimplification [10,11]. For instance, Figure 1 shows the types of particles that can be present in heavily recycled Ti-6Al-4V powder as measured in our lab at the University of Washington [12].

![Figure 1. Examples of damaged particles present after reuse: (A) Fractured; (B) Agglomerated; (C) Partially melted/fused. (D) The particle size distribution (PSD) over 30 (‘b’) build cycles [12].](image)

Given that the effects of powder morphology on as-built part quality is an open question (e.g. Gap PM3 [10]), this article presents and discusses initial developments of a framework for incorporating the effects of powder morphology into phase-field simulations. While there have been other developments to incorporate the effects of morphology in studying the constitutive response of granular media, most notably the “level-set” discrete element method or LS-DEM technique developed by Kawamoto, et al. [13], the methods have not yet been applied to phase-field simulations in AM. Furthermore, although the multisphere approach [14,15] is popular in incorporating the effects of morphology into DEM studies on PBF spreading dynamics, the accuracy of the approach is questionable [16] and the particle representations have not been extracted to understand downstream process effects on part quality. To the authors’ knowledge, there have been no other attempts at incorporating this information in the context of AM into phase-field simulations. Therefore, the primary contribution of this work is to present initial developments on two separate algorithms capable of generating irregular particles as initial conditions to phase-field simulations: EbPF (Experimentally-based Phase-Field) and PGPF (Procedurally Generated Phase-Field). Each of these algorithms will be discussed next in Chapter 2 while the conclusion in Chapter 3 will discuss the relative advantages of each algorithm as well as practical applications of the framework.

2. Algorithms

2.1. Experimentally-based Phase-Field (EbPF) Particles

In general, EbPF was developed using publicly available x-ray microtomography (XRCT) datasets (p35h8v1c300; p35h8v2c300 [17]). Each dataset can be represented as a three-dimensional matrix composed of voxels with 0.95 μm edge length. To demonstrate the algorithm, we will track a single particle (17459) identified in the three-dimensional slices shown in Figure 2. In general, particles are extracted via contrast correction and image segmentation. Since XRCT specimens are usually embedded within a material matrix, the MATLAB `imadjustn` function is used to adjust the contrast between the matrix and the particles so that the particles are correctly identified as separate entities using k-means clustering. The `imsegkmeans3` function takes as an input both the intensity-adjusted subvolume data as well as the number of particle material clusters desired. The segmented data is then denoised using `bwmorph3` and the ‘majority’, ‘clean’, and ‘fill’ operations. Figure 2 (B) shows the volumetric representation of the extracted particles.

After the particles are extracted from the matrix, they’re counted and numbered using a percolation algorithm similar to those found in [17,18]. Once the particles are labeled, they’re separated from the volume and saved individually into a binary file. Note that particles that intersect with the edges of the subvolume are filtered out and not considered. One such particle is identified in Figure 2 (C).
Next, particle properties are calculated using \texttt{regionprop3}. Statistics of the principal axis lengths are provided in Figure 3 (A) which describe the particle aspect ratio. This quantity has been linked to preferential orientation of the powder bed and can lead to anisotropy in as-built part properties [15]. Therefore, there’s an increasing desire to relate these characteristics to resulting process performance and build quality.

Finally, Open3D is used to generate a mesh of the particle using point cloud and surface normal data extracted using the MATLAB Computer Vision Toolbox (Figure 2 (D)). Although several meshing algorithms are implemented within Open3D, the one chosen for this framework is the Poisson Surface Reconstruction (PSR) due to its generality as it is not limited to convex particles [19]. Figure 3 (B) shows the particle mesh result for several mesh densities (e.g., “octree depth”) which can be used to adjust numerical expense. This particle can now be implemented in downstream numerical simulations.

2.2. Procedurally Generated Phase-Field (PGPF) Particles

The PGPF algorithm leverages the Blender 2.82a application programming interface (API) [20] and was developed for situations in which experimental data is not available. Most of the variability in powder morphology is attributed to Blender Mesh Modifiers that can generate several different powder morphologies that, at least qualitatively, match those found in practice. These methods operate only in Blender’s ‘Object’ mode which is used to generate and interact with mesh primitives at a global scale. The Subdivision Surface Modifier was used to increase mesh density and further refine particle detail as necessary. The Boolean Modifier and the \texttt{BoolTool} addon were used to combine primitive and edited meshes to generate more complex morphologies such as spherical particles with satellites and fractured particles. Finally, the Displace Modifier was used to generate non-spherical particles using displacement mapping [21]. Although not yet implemented in the algorithm, further flexibility can be realized by incorporating procedures that leverage Blender’s ‘Edit’ mode which provides access to mesh vertices, edges, and faces so that they can be manipulated. Figure 4 shows a selection of particles [12] generated using both ‘Object’ and ‘Edit’ modes compared to the particles presented in Figure 1. Additional details will be provided in a follow-up article.
Figure 4. Particles generated using Blender ‘Edit’ methodology

After the particles are generated, they’re distributed throughout the domain as shown in Figure 5 (A). Initial inputs to the algorithm include the PSD, scan speed, simulation time, and beam diameter which are used to size the simulation domain. This information was used to construct a rigid container from plane mesh primitives with passive, stationary rigid body properties. Once the particles are distributed, they’re relaxed using Blender’s rigid body solver and assigned order parameters using the Welsh-Powell algorithm similar to that discussed in [5]. The centroids of each of the particle mesh faces as well as their outward normal unit vectors are then exported as a CSV file. A custom class was generated by extending the InitialCondition class of the MOOSE framework [22]. During projection of the initial condition onto the finite element mesh, the contents of the CSV file are imported as a vector. In general, the position vector from the current MOOSE point to the face centroid is calculated and compared with the Blender unit face normal vectors. If the inner product between these two vectors is negative, the vectors are pointing in the opposite direction and the current MOOSE point is considered outside of the particle and assigned an “outer” value. Otherwise, the point is considered inside of the particle and assigned an “inner” value. Finally, to account for the diffuse interface that’s predominant in phase-field simulations, the magnitude of the position vector was compared with the user-defined interface width. Depending on the current position within the interface, the values were interpolated accordingly. An example of this is shown in Figure 5 (D). The final simulation initial condition is shown in Figure 5 (E) and is ready for follow-up analyses.

Figure 5. (A) Particles generated in a PBF domain after relaxation. (B) Construction of adjacency matrix. (C) Order parameter assignment. (D,E) Initial condition projected on finite element mesh.

3. Conclusion

The purpose of this article was to demonstrate initial developments on a framework for generating representative particle morphologies as initial conditions for numerical simulations of powder bed fusion AM processes. The framework is composed of two separate algorithms shown in Figure 6. This information was projected onto a finite element mesh in MOOSE by extending the InitialCondition base class. While both methods can be used to generate initial conditions to phase-field simulations, each has associated tradeoffs: EbPF is based on experimental particle data and therefore has a sound physical basis. However, the method assumes that the researcher has access to XRCT equipment which can be cost prohibitive. This requirement can be circumvented by generating the particles procedurally using PGPF. Nevertheless, generating particles procedurally is often subjective which does not guarantee that a particle’s representation is accurate. With that being said, both techniques can be used to capture the effects of powder morphology on process and as-built performance of PBF components. This is accomplished by generating process maps which describe the relationship between the process variables and product quality. By capturing this information, we can understand
the effects of powder morphology on powder bed fusion as-built part quality and begin to generate quality assurance and process risk mitigation strategies. Follow-up work will include refining workflow by implementing the algorithms as standalone addons within the Blender environment to improve accessibility to the academic community.

Figure 6. Particle generation framework. (Left) EbPF workflow. (Right) PGPF workflow.

Acknowledgement
This work was partially accomplished using facilities funded by the Joint Center for Deployment and Research in Earth Abundant Materials (JCDREAM) in Washington State. We also sincerely acknowledge the support of Boeing-Pennell Professorship funds, the Purvis Family Endowed Fellowship, and the Mark and Lisa Tuttle Endowed Fellowship in Mechanical Engineering at the University of Washington awarded to Mr. G. Kelley.

References
[1] Bourell, D. L., 2016, “Perspectives on Additive Manufacturing,” Annu. Rev. Mater. Res., 46(1), pp. 1–18.
[2] F42 Committee, 2015, Standard Terminology for Additive Manufacturing Technologies, ASTM F2792-12a, ASTM International, Conshohocken, PA.
[3] J. Raplee, A. Plotkowski, M. M. Kirka, R. Dinwiddie, A. Okello, R. R. Dehoff, and S. S. Babu, 2017, “Thermographic Microstructure Monitoring in Electron Beam Additive Manufacturing,” Scientific Reports, 7, p. 16.
[4] Gong, X., and Chou, K., 2015, “Phase-Field Modeling of Microstructure Evolution in Electron Beam Additive Manufacturing,” JOM, 67(5), pp. 1176–1182.
[5] Yang, Y., Ragnvaldsen, O., Bai, Y., Yi, M., and Xu, B.-X., 2019, “3D Non-Isothermal Phase-Field Simulation of Microstructure Evolution during Selective Laser Sintering,” npj Comput Mater, 5(1), p. 81.
[6] Cheng, B., Li, X., Tuffile, C., Ilin, A., Willeck, H., and Hartel, U., “Multi-Physics Modeling of Single Track Scanning in Selective Laser Melting: Powder Compaction Effect,” p. 16.
[7] Körner, C., 2016, “Additive Manufacturing of Metallic Components by Selective Electron Beam Melting — a Review,” International Materials Reviews, 61(5), pp. 361–377.
[8] Körner, C., Bauereiß, A., and Attar, E., 2013, “Fundamental Consolidation Mechanisms during Selective Beam Melting of Powders,” Modelling Simul. Mater. Sci. Eng., 21(8), p. 085011.
[9] Lu, L.-X., Sridhar, N., and Zhang, Y.-W., 2018, “Phase Field Simulation of Powder Bed-Based Additive Manufacturing,” Acta Materialia, 144, pp. 801–809.

[10] AMSC, 2018, Standardization Roadmap for Additive Manufacturing, Version 2.0, America Makes & ANSI Additive Manufacturing Standardization Collaborative.

[11] Vock, S., Klöden, B., Kirchner, A., Weißgärber, T., and Kieback, B., 2019, “Powders for Powder Bed Fusion: A Review,” Prog Addit Manuf, 4(4), pp. 383–397.

[12] Ghods, S., Schultz, E., Wisdom, C., Schur, R., Pahuja, R., Montelione, A., Arola, D., and Ramulu, M., 2020, “Electron Beam Additive Manufacturing of Ti6Al4V: Evolution of Powder Morphology and Part Microstructure with Powder Reuse,” Materialia, 9, p. 100631.

[13] Kawamoto, R., Andô, E., Viggiani, G., and Andrade, J. E., 2018, “All You Need Is Shape: Predicting Shear Banding in Sand with LS-DEM,” Journal of the Mechanics and Physics of Solids, 111, pp. 375–392.

[14] Parteli, E. J. R., and Pöschel, T., 2016, “Particle-Based Simulation of Powder Application in Additive Manufacturing,” Powder Technology, 288, pp. 96–102.

[15] Haeri, S., Wang, Y., Ghita, O., and Sun, J., 2017, “Discrete Element Simulation and Experimental Study of Powder Spreading Process in Additive Manufacturing,” Powder Technology, 306, pp. 45–54.

[16] Krügel-Emden, H., Rickelt, S., Wirtz, S., and Scherer, V., 2008, “A Study on the Validity of the Multi-Sphere Discrete Element Method,” Powder Technology, 188(2), pp. 153–165.

[17] Bentz, D. P., Mizell, S., Satterfield, S., Devaney, J., George, W., Ketcham, P., Graham, J., Porterfield, J., Quenard, D., Vallee, F., Sallee, H., Boller, E., and Baruchel, J., 2002, “The Visible Cement Dataset,” Journal of Research of the National Institute of Standards and Technology, 107(2), pp. 137–148.

[18] Dale Bentz, 2001, Clusterid.c, National Institute of Standards and Technology.

[19] Kazhdan, M., Bolitho, M., and Hoppe, H., 2006, “Poisson Surface Reconstruction,” Proceedings of the Fourth Eurographics Symposium on Geometry Processing, The Eurographics Association, Cagliari, Sardinia, Italy, pp. 61–70.

[20] Blender Online Community, 2020, Blender - A 3D Modelling and Rendering Package, Blender Foundation, Amsterdam.

[21] Szirmay-Kalos, L., and Umenhoffer, T., 2008, “Displacement Mapping on the GPU — State of the Art,” Computer Graphics Forum, 27, pp. 1567–1592.

[22] Permann, C. J., Gaston, D. R., Andrs, D., Carlsten, R. W., Kong, F., Lindsay, A. D., Miller, J. M., Peterson, J. W., Slaughter, A. E., Stogner, R. H., and Martineau, R. C., 2020, “MOOSE: Enabling Massively Parallel Multiphysics Simulation,” SoftwareX, 11, p. 100430.