Solid Freeform Fabrication 2022: Proceedings of the 33rd Annual International Solid Freeform Fabrication Symposium – An Additive Manufacturing Conference
Reviewed Paper

EXAMINING THE GPU ACCELERATION SPEED-UP FOR FINITE ELEMENT MODELING OF ADDITIVE MANUFACTURING

J. Logan Betts[1,2], Will Downs[1,2], Matthew J. Dantin[3] and Matthew W. Priddy[1,2]

[1]Department of Mechanical Engineering, Mississippi State University, MS, 39762
[2]Center for Advanced Vehicular Systems, Starkville, MS, 39759
[3]Naval Surface Warfare Center Carderock Division, West Bethesda, MD, 20817

Keywords: GPU Acceleration, Additive Manufacturing, Finite Element Analysis, Directed Energy Deposition

Abstract

Using a graphics processing unit (GPU) in addition to a central processing unit (CPU) has demonstrated promise for the acceleration of processing-intensive operations such as finite element (FE) simulations. Commercial FE solvers have begun to utilize GPU acceleration for classical multi-physics applications, but the speed-up for additive manufacturing (AM) simulations is not well understood. There is a significant need for GPU acceleration for metal-based AM FE simulations, which are computationally expensive because of the high mesh densities and the large number of time increments employed. This study examines the efficacy of GPU acceleration for Abaqus AM simulations, where benchmark simulations using a sequentially coupled FE thermo-mechanical model are run both with and without GPU acceleration. The speed-up is compared across the AM process for the thermal and mechanical analysis. In this study, GPU acceleration provided the ability to decrease simulation runtime by two-to-four times on 4-8 CPU cores, and one-to-two times on 16-32 CPU cores.

Introduction

Graphical processing units (GPU) have been used in addition to a central processing unit (CPU) to accelerate highly parallelizable operations for over a decade [1]. The first paper detailing methods of using graphics hardware for matrix multiplication was published in 2001, but the first GPU designed for computing was not released until the release of the Nvidia G80 GPU in 2006 [2]. Alongside the G80, Nvidia released compute unified device architecture (CUDA), which allowed CUDA C/C++/FORTRAN language to be parallelized on GPUs [3]. The integration of CUDA allowed graphics hardware to be used in deep learning and supplied a path for GPU acceleration to multi-physics simulations. Limited support for GPU acceleration to commercial finite element analysis (FEA) solvers was introduced in Abaqus/Standard 6.11 in 2011, showing promise to decrease simulation runtime [4]. Early studies on using GPU acceleration focused on linear-static analysis, looking at decreasing the total of number of CPUs, thus requiring fewer acceleration tokens [4], [5].

Metal-based additive manufacturing (AM) is a process for creating near-net shape parts layer-by-layer, using a moving heat source and material deposition. AM offers a reduction in processing and tooling steps when compared to traditional manufacturing, but often requires post-processing to reduce defects and residual stresses. Directed energy deposition (DED) is a metal-based AM process that uses a high-power coaxial laser with wire fed or blown powder deposition, but due to the localized heating and large thermal gradients experienced in the workpiece, the thermal history is hard to control and predict. Process parameters such as laser power, scanning speed, hatch spacing, and melt pool morphology drive the thermal history and affect the resultant microstructure and defects such as distortion, porosity, and residual stress formation. FEA is commonly used to model and simulate AM processes to predict distortion and residual stress formation, as well as conducting parametric studies of printing parameters [6]–[9].

Previous research on applying GPU acceleration to static mechanical analysis reported seeing up to a 2x speed up when using 1 GPU vs just 4 or 12 CPU cores [10], and a 5.2x speed up using 32 cores + 2x Nvidia K80 GPUs compared to 16 CPU cores using Abaqus. At present, no data could be found for using GPU acceleration with AM simulations in any commercial code, but custom codes have

been used and showed speedups of 3-30x through a modified solution strategy or by relaxing constraints [11], [12].

The purpose of this work is to present a first benchmark of the speedup of GPU acceleration applied to AM simulations in a commercial FEA solver. For the purposes of this paper, these simulations used Abaqus 2020 to model both an AM thermal and mechanical analysis of a 10-layer DED build. As modeling AM processes with commercial codes become more prevalent for a cost-effective prediction of distortion and processing parameters, finding methods to decrease runtime of these computationally expensive simulations is critical.

Methodology

The AM thermal and mechanical analyses were run on a workstation equipped with an AMD Threadripper 3970x 32-core processor, 256GB of DDR4 Memory, and a Samsung 980 Pro 2TB NVMe storage device. The simulations were performed in three configurations: (i) no GPU, (ii) a Nvidia GTX Titan Black, and (iii) a Nvidia RTX 3090. The simulations were run on the workstation with 4, 8, 16, and 32 CPU cores for each case. To compare the GPU acceleration performance to a traditional high performance computing cluster, the simulations were run on 20, 40, 80, and 200 CPU cores on a Cray CS300-LC cluster with two Intel Xeon Phi CPUs for a total of 20 cores per node.

The thermal-mechanical analysis is conducted in two sequentially coupled analyses. First, the AM thermal simulation is performed to generate the temperature history of the part. Both the thermal and mechanical simulations model a ten-layer buildup of the DED process, specifically Laser Engineered Net Shaping (LENS) using Ti-6Al-4V as the material [7], [9], [13]. This specimen shown in Figure 1(a), is a 5 mm x 3 mm x 6 mm test specimen using a spiral scan strategy shown in Figure 1(b). The Goldak double ellipsoidal heat sourced is utilized, since a local moving heat source is the most computationally expensive method in the Abaqus/2020 [14]. Material deposition is accounted for using element activation [8]. The ten-layer benchmark utilized a seed size of 0.08 mm in the part corresponding to 147,537 DC3D8 (heat transfer analysis) and C3D8 (mechanical analysis) linear brick elements. Both analyses used a 0.025 s time step, 0.5 mm hatch spacing, and 16.6 mm/s scan speed. Temperature dependent properties including thermal conductivity, density, and specific heat are used. Additional information on the thermal analysis and thermal properties can be found in published literature [8], [9], [13].

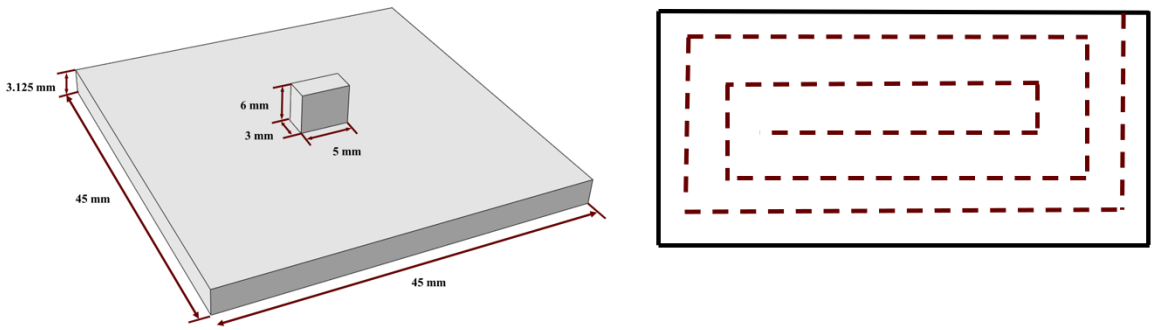

Figure 1: (a) dimensioned substrate and ten-layer benchmark specimen (b) laser scan strategy for each layer

Before the mechanical simulation starts, the temperature history is read in as shown in Figure 2. The evolving microstructural model of inelasticity (EMMI) is the material model used to predict the mechanical response. EMMI is a flow rule based internal state variable (ISV) plasticity model [12]. An example of this implementation is shown as a flowchart in Figure 2. EMMI is ideal for DED due to

temperature and rate dependence, cyclic temperature history, and recovery mechanisms that can relieve residual stress. The mechanical analysis can be used to make predictions of the residual stress or distortion within the part. More information on the calibration of material parameters and applications of EMMI for metal based AM is available in published literature [9].

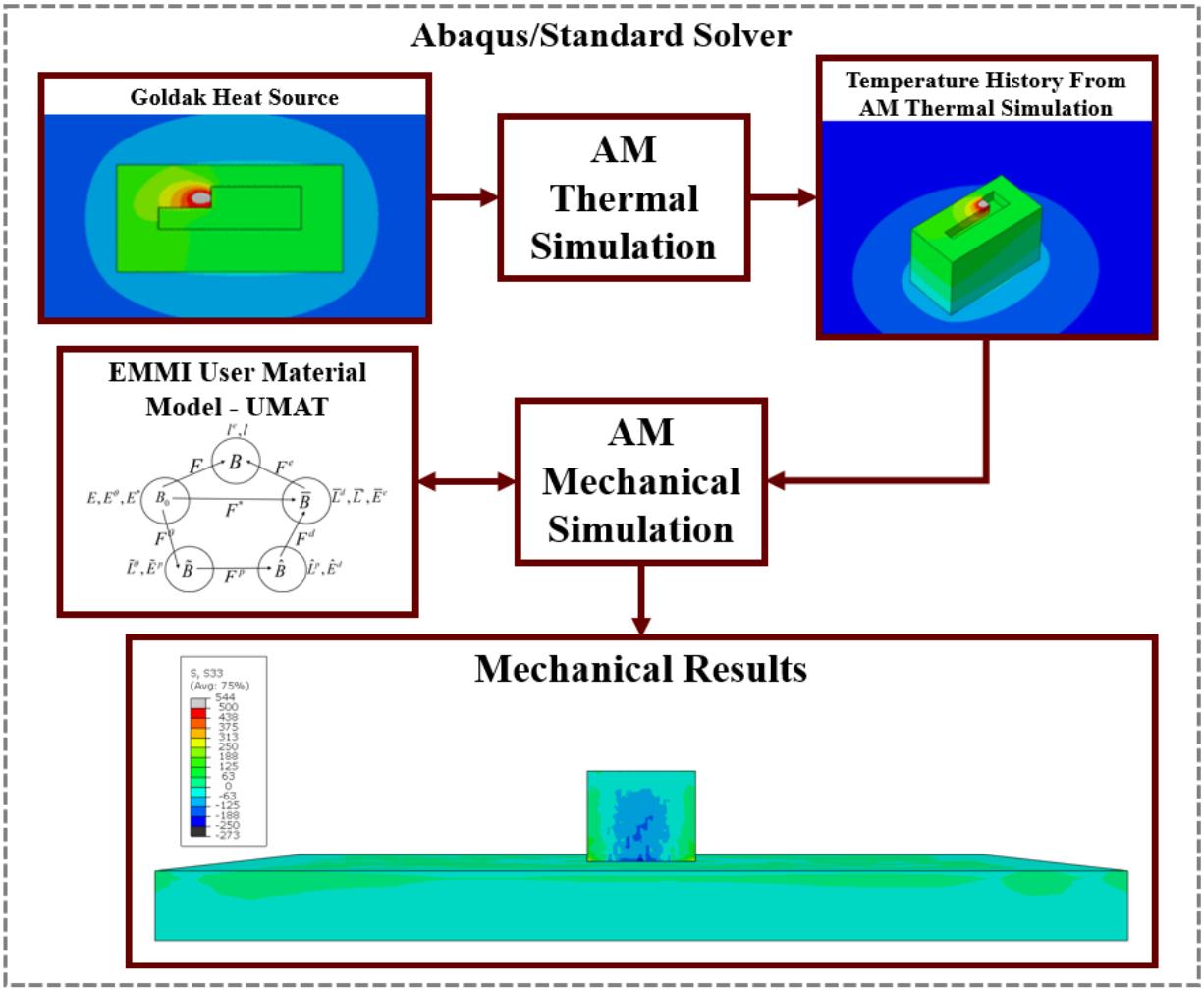

Figure 2: An overview of the AM thermal-mechanical analysis framework using the commercial FEA solver Abaqus

FEA simulations run in Abaqus require analysis tokens to perform an analysis. The number of tokens required depends on the number of CPU cores specified by the user and the user's hardware. To perform an analysis on a single CPU core, five analysis tokens are needed. An overview of the required number of analysis tokens with and without GPU acceleration is depicted in Table 1. Each GPU used behaves as a co-processor, only one additional analysis token is required to run a GPU alongside a CPU. The number of tokens required to run Abaqus on a number of CPU cores follows a logarithmic curve, and beyond eight CPU cores, no additional tokens are required to run a single GPU as a co-processor. The ability to implement GPU acceleration could allow users with few analysis tokens to run computationally expensive simulations efficiently.

Table 1: Number of Abaqus tokens required for a given number of CPU cores. *For four CPU cores, one additional Abaqus token is needed for GPU acceleration. Beyond eight CPU cores, one GPU can be included without additional Abaqus tokens.

Number of CPUs	4	8	16	20	32	40	80	200
Abaqus Tokens	8*	12	16	17	21	23	31	46

Results/Discussion

The simulations were run with a range of CPU core counts, with the Nvidia RTX 3090 and GTX Titan GPUs, as well as without GPU acceleration. All other parameters were held constant, and no other simulations were run alongside any simulations, regardless of the number of CPUs used on the workstation. For the thermal analysis running the jobs with the GTX Titan yielded a speed up of 1.3x, 1.1x, 0.77x, and 0.68x for 4, 8, 16, and 32 CPUs, respectively compared to no-GPU accelerated runs. On the RTX 3090 the thermal analysis showed a speed up of 1.5x, 1.27x, 0.91x, and 0.82x for 4, 8, 16, and 32 CPUs, respectively, as depicted in Figure 3(a). For the mechanical analysis, the GTX Titan yielded a speed up of 2.98x, 1.8x, 1.8x, and 1.1x for 4, 8, 16, and 32 CPUs, respectively compared to no-GPU accelerated runs. On the RTX 3090 the mechanical analysis showed speed ups of 4.15x, 3.02x, 3.04x, and 1.82x for 4, 8, 16, and 32 CPUs as depicted below in Figure 3(b).

The thermal and mechanical simulations were then run on a Cray CS300-LC cluster at 20, 40, 60, 80, and 200 CPUs without GPU acceleration. Similar runtimes of 2.75-3 hours were observed for the thermal simulation on 20-200 cores as depicted in Figure 3(c). Compared to running the thermal

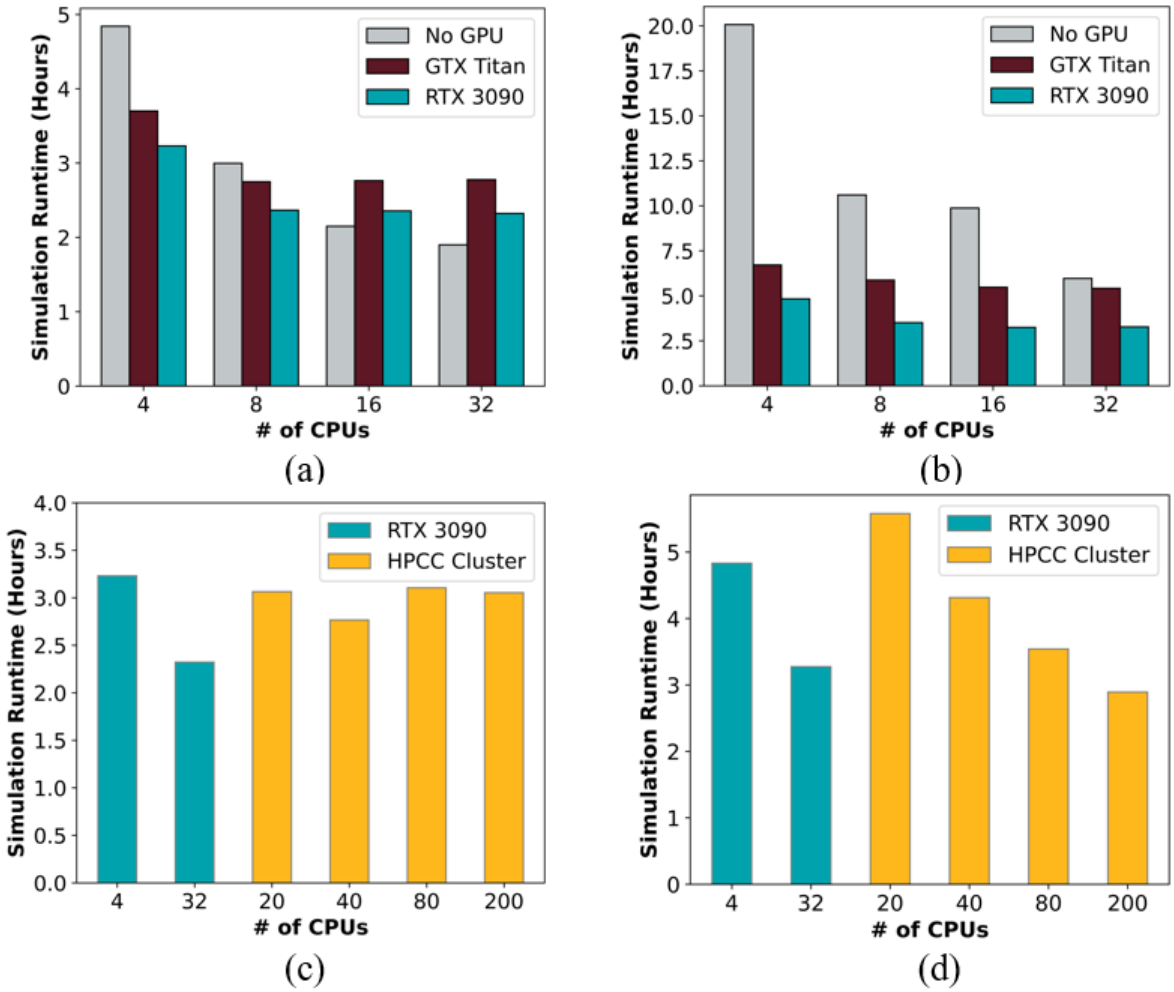

Figure 3: (a) AM thermal analysis comparison of runtime, (b) AM mechanical analysis comparison of runtime, (c) 4 & 32 CPU + 3090 GPU acceleration compared to cluster runs for thermal and (d) mechanical analysis.

simulations on the workstation with 4 Cores with the RTX 3090, a slowdown of 0.96x was observed on average compared to the cluster runs for 20 to 200 CPUs. For 32 Cores with the RTX 3090, the thermal simulations saw a speed up of 1.31x, 1.44x, 1.33x, and 1.31x for 20, 40, 80, and 200 CPUs, respectively, on the Cray cluster. Similar to the speedup seen for no GPU to GPU, the mechanical analysis yielded the greatest speedup. For the mechanical analysis, the workstation with 32 CPUs and the RTX 3090 yielded a speed up of 1.7x, 1.3x, 1.1x, and 0.88x for 20, 40, 60, and 200 CPUs, respectively compared to cray cluster as depicted in Figure 3(d). The speedup for the mechanical simulations is summarized in Table 2.

Table 2: Simulation speed-up comparing no-GPU to GPU Acceleration for mechanical analysis

GPU	32 Cores	16 Cores	8 Cores	4 Cores
3090	1.82x	3.04x	3.02x	4.15x
Titan	1.1x	1.8x	1.8x	2.98x

The nodal temperature for all nodes at each output time is extracted from the Abaqus output database and averaged for comparing the thermal response of the GPU and non-GPU accelerated simulations. An average percent difference of 0.99% was found between the GPU accelerated average nodal temperatures compared to the non-GPU accelerated simulations. For the purposes of this initial study, this analysis serves as a verification, but further analysis is needed looking at how the parallelization of GPU acceleration will affect the results locally. It is understood that parallel computing will give slightly different results due to rounding errors when processing sums and products in different orders, even without GPU acceleration [15], [16].

For the mechanical analysis, the workstation with 4 CPUs and the RTX 3090 yielded a speedup of 1.15x, 0.76x, 0.73x, and 0.6x for 20, 40, 80, and 200 CPUs, respectively compared to the Cray cluster as depicted in Figure 3(d). Implementing GPU acceleration allowed a simulation to be run faster on 4 cores using 9 acceleration tokens, than on 20 cores using 17 acceleration tokens. Despite the simulations running 25% slower compared to 40 and 80 CPUs, the analysis used 14 and 22 fewer acceleration tokens. Without GPU acceleration, running AM simulations on 4 CPUs becomes impractical. Two implications of this work should be noted: GPU acceleration could enable AM simulations for users that (i) do not have access to high performance compute clusters or workstations and (ii) have limited access to acceleration tokens. Many universities or small research groups may not have access or funding to support compute clusters but could run these simulations on a small workstation with hardware like a gaming PC. While the reduction of 10 to 20 acceleration tokens may sound minimal, these acceleration tokens often cost thousands of dollars each, per year. For smaller institutions, this difference of required tokens could be the difference in being able to run AM simulations at all.

Between the thermal and mechanical simulations, the effect of GPU-acceleration varied. On 32-cores, the thermal simulation ran faster without GPU acceleration, while the mechanical simulation ran 1.82x faster on the RTX 3090 GPU. Our working hypothesis is that Abaqus only computes the average nodal temperature value (NT11) for 3D elements, which leads to fewer degrees of freedom (DOF) in the thermal analysis. This means that at each node in the thermal analysis, there is only one DOF, while in the mechanical analysis, there are six DOF at each node. Our current working hypothesis is that an increased number of elements will affect GPU acceleration for a thermal analysis, but further investigation is needed.

Conclusions

In this work, a benchmark of using GPU acceleration for simulating AM thermal and mechanical simulations was presented. GPU acceleration was successfully applied to a 10-layer DED build in the commercial FEA solver Abaqus and demonstrated significant speedup in runtime—especially at lower CPU core counts. This benchmark only analyzed one mesh with 147,537 elements, so further study is needed to analyze how increasing the number of elements per CPU affects speedup of GPU acceleration. One possible avenue of exploration is investigating how CPU and GPU parallelization affects rounding errors and slightly changes the local results. The ability to decrease simulation runtime by two-to-four times by using GPU computing on 4-8 CPU cores is significant because of the accessibility of this computer hardware compared to specialty workstations and clusters. Many users may not have access to large numbers of costly acceleration tokens or dedicated compute clusters, but GPU acceleration could provide a way to run AM simulations efficiently with minimal acceleration tokens and consumer grade computer hardware. Future work will focus on understanding how the number of elements run on a CPU affects speedup and building large scale parts.

Acknowledgements

Research was sponsored by the Army Research Laboratory and was accomplished under Cooperative Agreement Number W911NF-20-2-0206. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the Army Research Laboratory or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation herein.

References

[1] I. Buck, “The Evolution of GPUs for General Purpose Computing,” p. 38.
[2] E. S. Larsen and D. McAllister, “Fast matrix multiplies using graphics hardware,” in *Proceedings of the 2001 ACM/IEEE conference on Supercomputing (CDROM) - Supercomputing ’01*, Denver, Colorado, 2001, pp. 55–55. doi: 10.1145/582034.582089.
[3] J. Sanders and E. Kandrot, *CUDA by example: an introduction to general-purpose GPU programming*. Addison-Wesley Professional, 2010.
[4] NVIDIA, “ACCELERATING ABAQUS COMPUTATIONS USING NVIDIA GPUS,” White Paper WP-07843-001_v03, Aug. 2016. [Online]. Available: www.nvidia.com
[5] “NVIDIA GPGPU with Abaqus 2017,” Webpage, Jul. 2017. [Online]. Available: https://www.tentechllc.com/blog/news_files/abaqus-2017-gpgpu-nvidia-gp100.php
[6] E. R. Denlinger, J. Irwin, and P. Michaleris, “Thermomechanical Modeling of Additive Manufacturing Large Parts,” *Journal of Manufacturing Science and Engineering*, vol. 136, no. 6, p. 061007, Dec. 2014, doi: 10.1115/1.4028669.
[7] S. M. Thompson, L. Bian, N. Shamsaei, and A. Yadollahi, “An overview of Direct Laser Deposition for additive manufacturing; Part I: Transport phenomena, modeling and diagnostics,” *Additive Manufacturing*, vol. 8, pp. 36–62, Oct. 2015, doi: 10.1016/j.addma.2015.07.001.
[8] M. J. Dantin, W. M. Furr, and M. W. Priddy, “Towards an Open-Source, Preprocessing Framework for Simulating Material Deposition for a Directed Energy Deposition Process,” p. 10.

[9] Matthew Joseph Dantin, “Thermomechanical modeling predictions of the directed energy deposition process using a dislocation mechanics based internal state variable model,” Theses and Dissertations, Mississippi State University, 2021. [Online]. Available: https://scholarsjunction.msstate.edu/td/5244
[10] S. Georgescu, P. Chow, and H. Okuda, “GPU Acceleration for FEM-Based Structural Analysis,” *Arch Computat Methods Eng*, vol. 20, no. 2, pp. 111–121, Jun. 2013, doi: 10.1007/s11831-013-9082-8.
[11] H. Huang, N. Ma, J. Chen, Z. Feng, and H. Murakawa, “Toward large-scale simulation of residual stress and distortion in wire and arc additive manufacturing,” *Additive Manufacturing*, vol. 34, p. 101248, Aug. 2020, doi: 10.1016/j.addma.2020.101248.
[12] E. B. Marin, D. J. Bammann, R. A. Regueiro, and G. C. Johnson, “On the Formulation, Parameter Identification and Numerical Integration of the EMMI Model : Plasticity and sotropic Damage,” p. 94.
[13] Matthew J. Dantin, William M. Furr, Matthew W. Priddy, “Toward a Physical Basis for a Predictive Finite Element Thermal Model of the LENS™ Process Leveraging Dual-Wavelength Pyrometer Datasets,” *Integrating Materials and Manufacturing Innovation*, (Accepted) doi: https://doi.org/10.1007/s40192-022-00271-6.
[14] J. Goldak, A. Chakravarti, and M. Bibby, “A new finite element model for welding heat sources,” *Metall Mater Trans B*, vol. 15, no. 2, pp. 299–305, Jun. 1984, doi: 10.1007/BF02667333.
[15] Dassault Systemes, “Abaqus 2019 User’s Manual.” 2019.
[16] S. D. Pollard and B. Norris, “A Statistical Analysis of Error in MPI Reduction Operations,” in *2020 IEEE/ACM 4th International Workshop on Software Correctness for HPC Applications (Correctness)*, GA, USA, Nov. 2020, pp. 49–57. doi: 10.1109/Correctness51934.2020.00011.