The 8th European Congress on Computational Methods in Applied Sciences and Engineering
ECCOMAS Congress 2022
5-9 June 2022, Oslo, Norway

PHYSICS-DRIVEN DIGITAL TWIN FOR LASER POWDER BED FUSION ON GPUS

Stephanie Ferreira[1,2], Benjamin Klein[2] and André Stork[1,2], Dieter Fellner[1,2,3]

[1] Fraunhofer Institute for Computer Graphics Research IGD
Fraunhoferstraße 5, 64283 Darmstadt, Germany
e-mail: info@igd.fraunhofer.de, www.igd.fraunhofer.de

[2] Technical University of Darmstadt
Karolinenplatz 5, 64289 Darmstadt, Germany
www.tu-darmstadt.de

[3] Graz University of Technology
Inffeldgasse 16c, 8010 Graz, Austria
www.tugraz.at

Key words: physics-based digital twin, additive manufacturing, GPGPU computing, LPBF

Abstract. Metal Additive Manufacturing (AM) processes such as Laser Powder Bed Fusion (LPBF) suffer from part distortion due to the localized melting and resolidification of the metal powder, which introduces stresses and strains. Despite becoming more and more important as a manufacturing process, options for simulating the printing process to predict the distortions are limited, especially because existing solutions often require very long computation times. In this work, we present the results of an implementation of the inherent strain method on graphics processing units (GPUs) that exploits the massive parallelism of the many GPU cores to speed up the simulations considerably compared to CPU-based implementations.

1 INTRODUCTION

The field of Additive Manufacturing (AM) is ever-growing. The market size for 3D printing products and services in 2020 was estimated at 12.7 billion USD, having doubled in size from 2016 [1]. Wohlers Associates [1] expects this value to double again by 2024. There are several reasons for this success. AM allows for the production of geometrically complex parts, which would be laborious or near impossible to create using conventional manufacturing processes such as cutting or casting. Having the ability to create more complex parts allows for weight-saving designs, making the technique interesting, e.g. to the aerospace industry. Further, it enables quick prototyping and highly individualized one-off products without the associated tooling costs.

Over the last decades, numerous AM processes have been developed which can operate on a wide range of materials. ISO 52900 lists seven distinct processes, one of which is the so-called Powder Bed Fusion (PBF). We focus on Laser Powder Bed Fusion (LPBF) for its ability to produce metal parts. Parts made of metal are of interest to the industry for their high strength, heat-resistance and durability. A major challenge in the employment of LPBF is part distortion.

In LPBF, a focused laser is used to fuse together a thin layer of fine metal powder in precisely the spots where the cross-section of the part is supposed to materialize. The build platform then moves downward by the height of one layer and a new layer of metal powder is dispensed on top. Again, the laser melts this new layer, welding it to the layer below. This procedure is repeated until the part has been fully "printed".
Rapid heating, melting and subsequent resolidification lead to large thermal strain and residual stress in the material. These stresses induce a deformation in its shape, potentially making it unfit for its intended purpose.
Being able to consider or even mitigate these distortions already in the design phase, before the lengthy build process, obviates expensive experiments.
For this reason, computational simulation is employed. In available commercial software the process is often simulated using thermomechanical Finite Element Analysis (FEA), which can yield quite accurate results, but quickly becomes infeasible for larger structures due to very long computation times. Rapid simulation is however crucial to fully leverage the benefits of simulation in the design phase. A further area where fast simulation times are important are digital twins. Simulations conducted ideally in real-time are required as input for digital twins. We employ the inherent strain method and present a GPU implementation, which is able to achieve significant speed-ups compared to a CPU-based implementation.

2 RELATED WORK

Much research into the simulation of LPBF has been undertaken. Closest to the actual physical processes taking place are numerical models on the particle scale, which take into account the interaction between laser and metal powder and melt pool. Qui et al. [2] developed such a model, where they carry out a fluid dynamic calculation and simulate a multitude of effects including surface tension, buoyancy and heat loss due to evaporation, conduction, convection and radiation, but on only 60 particles of the powder. Clearly, this method alone is not usable to predict distortion within the complete part. Ways must be found to reduce computational complexity.

Li et al. [3] present a multiscale modeling approach in which they combine a particle-scale laser scan model with a meso-scale layer hatch model and a macro-scale part model. From the micro-scale model, an "equivalent heat source" is exported which is used in the meso-scale model to obtain a local residual stress field. This stress field is then mapped to the macro-scale model to predict part distortion. In other approaches, the particle-scale effects are completely neglected and only heat transfer and solid mechanics are considered to estimate the mechanical response. An example is the work by Hodge et al. [4] where they describe a simulation based on a thermal model coupled to a solid mechanics model. Still, even for the simulation of a domain that only comprises 1mm^3, the authors note the "resource intensive nature of the algorithm". A drastic reduction of computational effort was achieved through use of the Inherent Strain Method (ISM). Originally conceived by Ueda et al. [5] to estimate residual stresses in weld joints it has since been adapted multiple times to the PBF process, see, for example, Alvarez et al. [6]. What makes it so efficient is the fact that it fully dispenses with simulating the scanning pattern of the heat source. Instead, an "inherent" strain is determined experimentally, by printing test objects using the machine and scanning strategy of interest and subsequently

measuring the magnitude of residual stress/strain. This inherent strain is then applied – in a layer-wise fashion – to the full-size FE mesh of the part, which allows performing a comparatively undemanding elastic mechanical Finite Element Analysis. Munro et al. [7] presented their own implementation of this method and demonstrated that multiprocessors can be used efficiently in simulations based on the ISM because mechanical responses to the layer-wise application of inherent strains can be computed independently and in parallel.

3 INHERENT STRAIN METHOD

3.1 Inherent strain method

If conducting a thermal-elastic simulation of the manufacturing process the strain is split into several components describing the different phenomena occurring during the heating and cooling process. These contributions to the strain inherent to the process are replaced by a constant inherent strain value approximating these values for specific configurations.

$$\varepsilon_{\text{total}} = \varepsilon_{\text{elastic}} + \varepsilon_{\text{inh}} \tag{1}$$

The inherent strain approximates in a black-box manner the effects induced by thermal changes, plasticity, phase-change and creep [8]. The value of the inherent strain needs to be calibrated beforehand. Specifically, the inherent strain is dependent on the printing settings, scanning strategies and materials. Calibration can be done by printing a reference part and measuring the occured distortion [9].
Once the inherent strain is known it can be used as the input of quasistatic structural mechanics simulation to predict the distortions occurring for a printed part. This black-box procedure works very well to predict distortions. If, however, additional information about stresses are needed a plasticity correction step has to be performed in order to retrieve acceptable accuracy. For further details we refer to Munro et al [7]. Here we focus on distortion prediction.

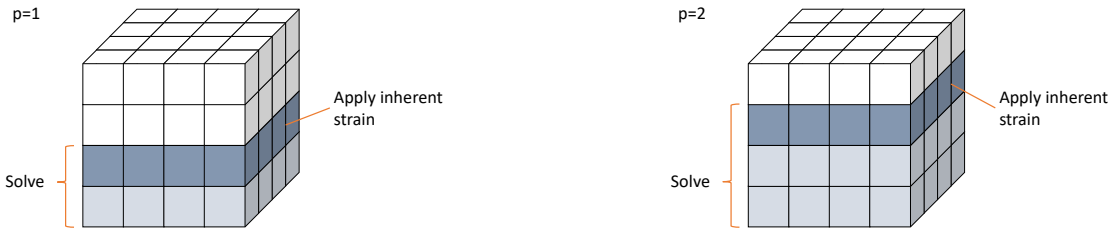
3.2 Layer-wise finite element method

To simulate the manufacturing process using the inherent strain method a quasistatic simulation is conducted following Munro et al. [7]. The simulation is divided into multiple process step. In each process step the addition of one layer is simulated. This is done by applying the inherent strain to the top layer and performing a linear static finite element simulation of all layers below. This process is depicted in Figure 1.
We base our discretization on the voxel mesh of the to be printed part as this is a natural choice in the context of additive manufacturing. Thus, the object is discretized using regular hexahedral finite elements [10]. This leads to a linear system of equations which needs to be solved in each process step p:

$$\mathbf{K}_p \tilde{\mathbf{u}}_p = \mathbf{f}_p(\varepsilon_{\text{inh}}), \tag{2}$$

where $\mathbf{K}_p$ is the stiffness matrix of all active layers, $\mathbf{u}_p$ the stacked displacement vector of all nodes involved in active elements and $\mathbf{f}_p(\varepsilon_{\text{inh}})$ is a force field function representing the applied inherent strain on the top active layer in the current step.

The total displacement is given by the linear superposition of all increments computed in



a) Process step 1. b) Process step 2.

Figure 1: Depiction of simulation process. In each process step the next layer is activated (depicted in blue). On this layer the inherent strain is applied and a static finite element simulation for all layers active layers is conducted.

each process step:

$$\mathbf{u} = \sum_{p} \mathbf{u}_p, \tag{3}$$

where $\mathbf{u}_p \in \mathbb{R}^{3n}$ with n being the total number of nodes. The entries of $\mathbf{u}_p$ are equal to the entries of $\tilde{\mathbf{u}}_p$ for all active nodes and are 0 for all inactive nodes in process step p.

4 MASSIVELY PARALLEL SIMULATION OF LPBF PROCESS

Graphics processing units (GPUs) are manycore chips providing hundreds to thousands of cores. GPUs offer a "single instruction, multiple data" (SIMD) architecture, which enables massively parallel computations. These require specialized algorithms and data structures in order to leverage this potential. For suitable applications the use of GPUs can speed up computation times significantly compared to CPU-based algorithms. Coupled with iterative methods this can efficiently be used to compute the finite element simulation (see e.g.[11]).
Munro et al. [7] propose parallel computation of process steps due to them being independently computable. This can be efficiently done on multicore CPUs. For GPU-parallelization this is however not suitable. Instead, another strategy to exploit the massive parallel capabilities of GPUs needs to be found.

We propose leveraging GPU-computing capabilities to solve the linear static finite element simulation in each step. As the to be solved matrix in Equation (2) is generally large, sparse and positive-definite an adequate method to solve this system of linear equations has been chosen, for details see Shewchuk [12]. The GPU can efficiently be used for matrix-vector multiplications and dot products, these operations are very suitable to be conducted on the GPU for sparse systems.

5 RESULTS

The described method was implemented in C++ and CUDA 11.2 [13]. To assess the performance of our implementation, we compared our results with a CPU-based implementation. To this end, we recreated the experiment conducted by Munro et al. [7], where a series of meshes in the form of cubes were constructed. Each mesh increased in the number of elements as seen in Table 1. While Munro et al. used 10 to 50 cores of a computational cluster, evaluation of this work took place entirely on consumer-grade hardware, more specifically an Intel i7-3770k CPU (2012) and a NVIDIA GTX 980Ti GPU (2015). The simulation time required for each mesh is depicted in Figure 2.

Table 1: Mesh sizes used for perfomance evaluation. The meshes scale in two ways. Firstly, the degrees of freedom increase leading to larger systems of equations. Secondly, the numbers of layers increase, leading to more process steps to be solved.

Elements	Degrees of freedom
$50 \times 50 \times 50$	397,953
$60 \times 60 \times 60$	680,943
$70 \times 70 \times 70$	1,073,733
$80 \times 80 \times 80$	1,594,323
$90 \times 90 \times 90$	2,260,713
$100 \times 100 \times 100$	3,090,903

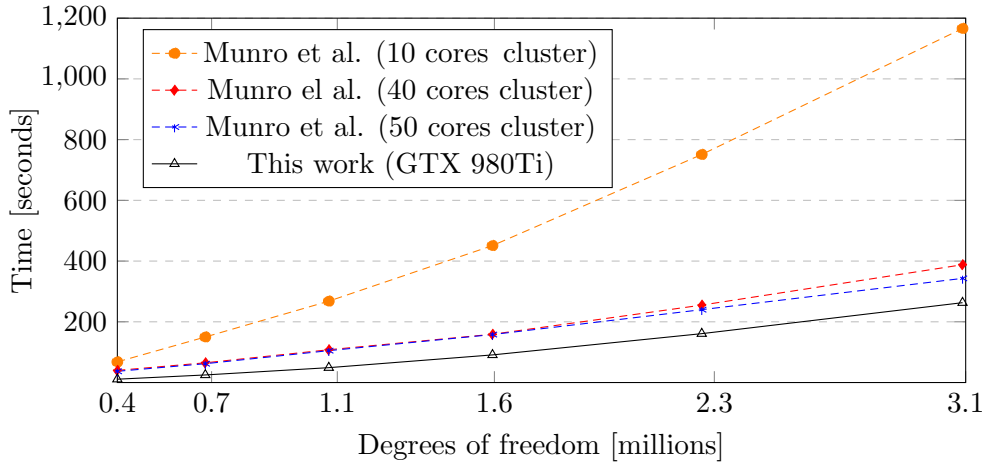


Figure 2: Processing time for the different meshes. Our GPU-based implementation running on a consumer grade NVIDIA GPU outperforms the reference computations for all tested cases.

In all szenarios our GPU-based implementation outperforms the the CPU-based parallelization by Munro et al. [7]. We further are able to provide comparable accuracy. In Figure 3 our result is shown for a 100 mm $\times$ 100 mm $\times$ 100 mm cube, for which we used the same material

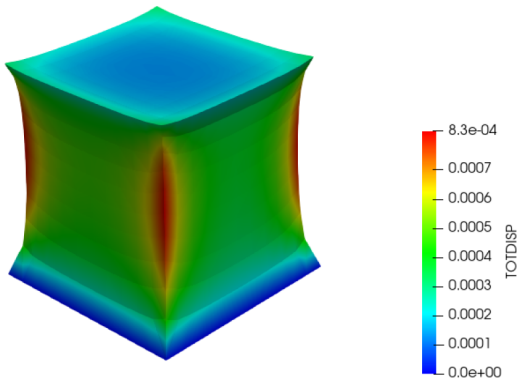


Figure 3: Predicted deformations after printing process for a cube with side length of 100 mm discretized by $10 \times 10 \times 10$ elements, visually exaggerated by a factor of 20. The colors indicate the length of the displacement vector per node.

parameters and inherent strain as Munro et al. [7]. Comparing our result to the results of Munro et al., they are visually in agreement and the maximum displacement length of 0.83 mm is equal to their reported value.

This indicates that GPU-parallelization is a promising technique to drastically speed-up the process simulation while still providing comparable results.

Table 2: Different GPUs used for performance evaluation and their performance for a process simulation of a model with 1.2M degrees of freedom and 80 layers.

Model	TFLOPS	Year	Simulation time
Consumer GPUs			
NVIDIA GeForce GTX 980Ti	5.6	2015	84.00 s
NVIDIA GeForce RTX 2080Ti	13.5	2018	14.63 s
Professional GPUs			
NVIDIA Quadro GP100	10.3	2017	27.55 s
NVIDIA A100	19.5	2020	7.58 s

The method proposed by Munro et al. scales with the number of CPU-cores used. In our case scalability can be achieved by increasing GPU performance.

To evaluate the scaling of computation time with increasing GPU performance, we ran simulations on four different GPU models out of a wide range of performance (see Table 2 and Figure 4). As memory requirements of our method are relatively low, affordable high-performance consumer grade GPUs are able to provide compatible performance compared to highly priced professional GPUs.

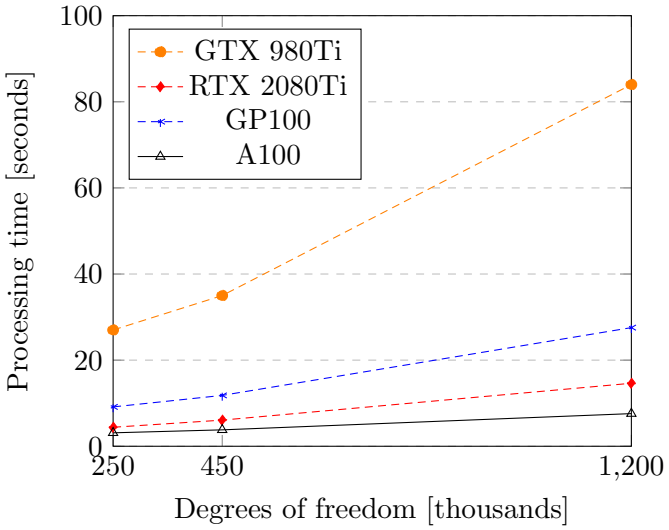


Figure 4: Simulation time for the different meshes with increasing degrees of freedom using different GPUs. The lower the line, the lower the simulation time, i.e. the better the performance. See Table 2 for a comparison of the GPUs used.

6 CONCLUSION AND FUTURE WORK

In this work, a GPU based simulation was presented that enables users to quickly predict unwanted deformations occurring during the 3D printing of metal parts in the Laser Powder Bed Fusion (LPBF) process. By applying the Inherent Strain Method (ISM), computational effort was reduced to a relatively inexpensive purely mechanical Finite Element Analysis (FEA). Our main contribution is to use GPU-based algorithms for the solution of the linear problem resulting from the Inherent Strain Method. In evaluating the simulation output, comparisons were made with results presented by Munro et al. [7]. Deformation predictions were in good agreement with the reference results. Even with just one consumer-grade GPU, the CUDA implementation of this work achieved faster simulation times compared to the implementation of Munro et al. on 50 cores of a computational cluster. The implemented software enables users to perform a fast simulation of the LPBF printing process on comparatively inexpensive computers equipped with a GPU, instead of requiring a high number of CPU cores.

In future work further analysis is needed to evaluate performance with more complex structures. Furthermore, in order to be able to retrieve information about stresses occuring in the printed part it is necessary to include the plasticity correction step as proposed by Munro et al. [7]. Lastly, in our GPU-based implementation process-step parallelizability as proposed by Munro et al. has not been exploited yet, i.e. all process steps are conducted consecutively. A straight-forward approach to address this would be to use a multi-GPU setup in which process-step parallelizability could be exploited in the same fashion as proposed for the multi-core CPU.

ACKNOWLEDGEMENTS

This work was supported by the European Union research projects *DIGITbrain*, which is co-funded by the Horizon 2020 Framework H2020-DT-2018-2020 of the European Commission under Grant No. 952071; *QU4LITY*, which is co-funded by the Horizon 2020 Framework H2020-DT-2018-2020 of the European Commission under Grant No. 825030; *CO-VERSATILE*, which is co-funded by the Horizon 2020 Framework H2020-DT-2018-2020 of the European Commission under Grant No. 101016070.

References

[1] Terry Wohlers, Ian Campbell, Olaf Diegel, Joseph Kowen, and Noah Mostow. *Wohlers report 2021 : 3D printing and additive manufacturing : global state of the industry*. Fort Collins, Colorado, 2021.

[2] Chunlei Qiu, Chinnapat Panwisawas, Mark Ward, Hector C. Basoalto, Jeffery W. Brooks, and Moataz M. Attallah. "On the role of melt flow into the surface structure and porosity development during selective laser melting". In: *Acta Materialia* 96 (2015), pp. 72–79. ISSN: 1359-6454. DOI: https://doi.org/10.1016/j.actamat.2015.06.004.

[3] C. Li, C.H. Fu, Y.B. Guo, and F.Z. Fang. "A multiscale modeling approach for fast prediction of part distortion in selective laser melting". In: *Journal of Materials Processing Technology* 229 (2016), pp. 703–712. ISSN: 0924-0136. DOI: https://doi.org/10.1016/j.jmatprotec.2015.10.022.

[4] N. E. Hodge, R. M. Ferencz, and J. M. Solberg. "Implementation of a thermomechanical model for the simulation of selective laser melting". In: *Computational Mechanics* 54.1 (July 2014), pp. 33–51. ISSN: 1432-0924. DOI: 10.1007/s00466-014-1024-2.

[5] Yukio Ueda, Keiji Fukuda, Keiji Nakacho, and Setsuo Endo. "A New Measuring Method of Residual Stresses with the Aid of Finite Element Method and Reliability of Estimated Values". In: *Journal of the Society of Naval Architects of Japan* 1975 (1975), pp. 499–507.

[6] Pedro Alvarez, Joseba Ecenarro, Inaki Setien, Maria San Sebastian, Alberto Echeverria, and Luka Eciolaza. "Computationally efficient distortion prediction in Powder Bed Fusion Additive Manufacturing". In: *International Journal of Engineering Research and Science* 2 (2016), pp. 39–46. ISSN: 2395-6992.

[7] Dirk Munro, Can Ayas, Matthijs Langelaar, and Fred van Keulen. "On process-step parallel computability and linear superposition of mechanical responses in additive manufacturing process simulation". In: *Additive Manufacturing* 28 (Aug. 2019), pp. 738–749. DOI: 10.1016/j.addma.2019.06.023.

[8] Nils Keller and Vasily Ploshikhin. "New Method for fast predictions of residual stress and distortion of AM parts". In: Aug. 2014.

[9] Iñaki Setien, Michele Chiumenti, Sjoerd van der Veen, Maria San Sebastian, Fermín Garciandía, and Alberto Echeverría. "Empirical methodology to determine inherent strains in additive manufacturing". In: *Computers & Mathematics with Applications* 78.7 (2019). Simulation for Additive Manufacturing, pp. 2282–2295. ISSN: 0898-1221. DOI: https://doi.org/10.1016/j.camwa.2018.05.015.

[10] O.C. Zienkiewicz, R.L. Taylor, and J.Z. Zhu. *The Finite Element Method. Its Basis And Fundamentals*. 6th ed. Elsevier Butterworth-Heinemann, 2005. ISBN: 0750663200.

[11] Daniel Weber, Johannes Mueller-Roemer, Christian Altenhofen, André Stork, and Dieter Fellner. "Deformation simulation using cubic finite elements and efficient p-multigrid methods". In: *Computers & Graphics* 53 (Dec. 2015), pp. 185–195. DOI: 10.1016/j.cag.2015.06.010.

[12] Jonathan R Shewchuk. *An Introduction to the Conjugate Gradient Method Without the Agonizing Pain*. Tech. rep. USA, 1994.

[13] NVIDIA, Péter Vingelmann, and Frank H.P. Fitzek. *CUDA 11.2*. URL: https://developer.nvidia.com/cuda-toolkit.