₁ # Part-scale heat transfer modeling in LPBF with
₂ # matrix free and GPU computing

₃ Florian Dugast*, Petros Apostolou, Qian Chen, Albert To

₄ *Department of Mechanical Engineering and Materials Science, University of Pittsburgh,*
₅ *Benedum Hall, 3700 O'Hara Street, Pittsburgh, PA 15261*

₆ **Abstract**

This paper presents a matrix free algorithm implemented on GPU in order to solve the transient heat transfer problem at the part scal einvolved in the LPBF process.

₇ *Keywords:* additive manufacturing, matrix free finite element, gpu
₈ computing, heat transfer modeling

---

1. Introduction

In engineering design, additive manufacturing (AM) is intensively developed in order to create more complex geometries compared to traditional machining. In this process, new material is added in a layer by layer fashion to create the desired part and one of the most popular technique to do is the powder-bed fusion process (PBF). In fact each layer is composed by a powder (metallic in this study) and the localized heating of this powder will solidify it and create solid material. The heating and cooling induced by the laser will produce high temperature gradients during the printing and so residual stresses. As the build is composed of several thousands of layers, the accumulation of residual stresses is critical and can result in large deformation or cracking. In order to ensure the reliability of AM at an industrial scale, a lot of efforts is dedicated to the understanding or the prediction of such behaviors. Unfortunately, the numerical simulations are difficult to implement in the AM context for several reasons : an accurate modeling implies multiphysics phenomenons (heat transfer, fluid flow or solid mechanics) and multiple scales (from powder scale to part scale). A realistic powder scale simulation with all the physics introduced above require a mesh size about few microns and can take about few days to compute [1]. Therefore such detailed simulations at the part scale (few centimeters resolution) is not possible with actual computational resources and some simplifications has to be done.

First the removal of the fluid mechanics in the AM modeling enables the calculation of few layers instead of few scan lines for the more detailed models. The influence of the cooling induced by the gas flow or of the laser reflections can be taken into account by some heat transfer coefficients accounting for heat losses [2]. Such coefficient are either determined by a more detailed simulation of a smaller model or calibrated with experiments [3]. Also the powder can be treated as a continuum medium rather than being represented as many particles. About the transient heat transfer analysis, the heat source comes from the input of the laser and is usually represented by the double ellipsoid model introduced by Goldak [4]. Then this term depends on the laser properties and location and therefore is entitled as moving heat source model, following the laser path. This approach is still quite costly from a computational point of view as the laser spot dimensions is about few microns while the dimensions of a single layer are assumed to be few centimeters.

In order to get an affordable part scale simulation, the Goldak's model is then substituted by heat loads averaged over a scan line [5, 6] or more radically by an average on the entire layer [7, 8]. This latter approach called flash heating by Gouge et al [9] will be used here as it is the cheapest ones and consequently allow the calculation of a large number of layers for a representative part-scale simulation. Of course, there is a loss of accuracy compared to a more detailed models as the melt pool cannot be represented by such approach. However, it allows to get reliable results to be used for the calculations of deformation in the entire part [7, 10], which is more difficult to achieve with more demanding models. Also it can be useful to predict the preheat temperature of each layer which acts as a boundary condition is powder scale models.

Next, the size of one layer in the PBF process is about 20 to 60 microns. It represents approximately 1250 layers for a part of 5 cm height. Each layer calculation must computed within several time steps in order to take into account the heating and cooling phase. About 20 time steps are necessary to obtain a good accuracy so the entire model requires about 25000 computations of a transient heat conduction problem. One solution to reduce this number is to lump physical layers. In other words, in the mesh discretization, a layer thickness larger than 40 microns is used so that one numerical layer represents for example 10 or 20 physical ones. Li and al [10] have studied the effect of this block layer simplification on the accuracy of the results. A reasonable comparison has been achieved for residual stresses with 0.4 mm blocks but the direct effect on temperature was not investigated. Also the simulation time was still quite high (76 h) to obtain this result.

Another approach to tackle this issue is to use adaptive mesh refinement on the current activated layer [11]. In fact, one can use a fine mesh for the heated layer and a coarser one for layers below, assuming that all the physics complexity (high temperature gradients) happens in the surrounding of the activated layer. Even if it is a very efficient for simple parts with a constant cross section where the coarsening can be large, it will be limited for more complex parts which would require a minimum mesh size in order to be able to represent small features. That is why the simplified assumptions presented above are not sufficient to ensure a fast simulation time for a part-scale model.

From an hardware point of view, parallel computing can be used in order

to compute larger models at a reasonable computational time. While first limited to CPU clusters in the 2000's, it has been developed now based on the Graphics Processor Cards. Such devices have been first extensively developed for video games purpose with an architecture composed of many cores and an high memory bandwidth to calculate pixels in picture. In parallel, programming languages (CUDA by Nvidia among others) have been created to be able to use GPU cards for scientific computing. As an example, the AmgX library has been created to solve linear system on GPU based on a multigrid approach which can be applied in the finite element framework. But even if the solving time can be reduced with the use of GPUs, the assembly of the global matrix is still time and memory consuming. The assembly time is an issue for the layer by layer process in AM because new elements are activated when a new layer is created so the global matrix of the linear system needs to be modified regularly during the simulation. Next, the memory requirement of the global matrix assembly is more critical for GPU compared to CPU. In fact the GPU memory is usually limited to approximately 12 GB (for example with the Titan V which will be used in this article) compared to few hundreds GB of CPU memory for a classical desktop. It is also to be noticed that the memory transfers between CPU and GPU have to be limited for an efficiency purpose.

For all these reasons, the matrix-free FEM method is an interesting alternative to be implemented on GPUs. First developed in the late 80's for low memory computers [12], its main principle is to avoid the assembly of the global matrix K in KU=F by performing all matrix-vector products involving K in by their counterparts at the elements or nodes level. The algorithm use for the solving of the linear system is based on the conjugate gradient in this article. It implies some loops on elements or nodes which can parallelized over the large number of CUDA cores in the GPU card to provide an high efficiency. In order to simplify the implementation of this solver, a voxel based mesh with hexahedrons elements can be used as a regular mesh grid [13]. This technique avoids a complex mesh generation and is useful in the AM process where it is convenient to represent a new layer which has a constant thickness by such block elements. In [13], it has been applied for a linear elasticity problem and Prabhune and Suresh [14] used a matrix-free on GPU in the additive manufacturing framework for residual stresses prediction.

Compared to these previous matrix-free implementations, this article is

4

focused on the transient heat transfer modeling in the PLBF process. As the boundary conditions employed some heat transfer coefficients, an experimental measurement of a far field temperature will be conducted for the calibration of such parameters as it is usually done in literature for such simulations [15–17]. In [17] thermocouples have been inserted in both the base plate and the build. In order to put them inside the part, the printing has to be stopped and re-run again. Even if it gives additional measurements for the calibration, the interruption of the building process can have an effect on the temperature of the previous layers, due to this extra cooling time and it can have jeopardized the accuracy of the results. Also the thermocouple are not welded and some air holes can affect the measurements so for these reasons, only a thermocouple in the base plate has been used in this article. In addition of the part-scale heat transfer simulation and experimental campaign, the effect of lumping layers will also be investigated, similarly to what Li et al. [10] did but with a focus on the temperature results rather than the residual stresses and for a minimum block thickness corresponding to the real physical layer (40 microns) rather than 400 microns.

To sum up, the article is organized as follows. The heat transfer model and the associated experimental campaign are presented in section 2 as the matrix free implementation of the conjugate gradient algorithm on GPU is introduced in section 3. The section 4 will be dedicated to numerical results with different cases : a comparison with Ansys for a simple rectangular bar in order to validate the matrix-free approach compared to the global assembly process and the calibration of the heat transfer coefficients with the experimental campaign. Then some additional test cases will be computed in section 5 : a larger domain (150 mm^3 compared to 50 mm^3) with a more complex geometry for a more realistic simulation time and a study about the effect of the layer thickness on the accuracy of the results.

2. method

2.1. Transient heat tranfer analysis

The governing equations for the transient heat conduction problem is given as :

$$\rho C_p \frac{\mathrm{d}T}{\mathrm{d}t} = \frac{\partial}{\partial x}\left(k\frac{\partial T}{\partial x}\right) + \frac{\partial}{\partial y}\left(k\frac{\partial T}{\partial y}\right) + \frac{\partial}{\partial z}\left(k\frac{\partial T}{\partial z}\right) + Q$$

$$T(x,t) = T_p \quad \text{on} \quad \Gamma_D \qquad (1)$$

$$k\frac{\partial T}{\partial x} = h\left(T(x,t) - T_\infty\right) \quad \text{on} \quad \Gamma_{h,x}$$

with ρ, C_p and k the material properties, density, heat capacity and thermal conductivity respectively. The printing is realized with Inconel718, which has temperature-dependent properties for the temperature range involved in the simulation. Such properties have been selected as the same used in ansys workbench but they can be found also in literature. Γ_D stands for a Dirichlet boundary conditions with a prescribed temperature T_p and $\Gamma_{h,x}$ represents a Robin boundary condition for the x component of the heat flux with h a heat transfer coefficient and T_∞ the temperature of the medium. For conveniency, the boundary conditions have been expressed along the x-component but it follows the same principle for y and z. Finally, Q is the volumetric heat source in W/m^3 used to take into account the laser heating. The equation (1) will be discretized with the finite element method and linear shape functions. In matrix form, one can write this problem as :

$$M\dot{T} + KT = F \qquad (2)$$

with M, K and F, the mass matrix, the conductivity matrix and the force term respectively. For the time discretization, a implicit euler scheme will be employed such as [18]:

$$\left(\frac{1}{\Delta t}M + K\right)T^{n+1} = \frac{1}{\Delta t}MT^n + F^{n+1} \quad \Longleftrightarrow \quad M_1 T^{n+1} = F_1 \qquad (3)$$

for the step n with the time step Δt. The M and K matrix come from the assembly of the element matrix M_e and K_e and the matrix free algorithm will work directly with these element matrix. The solution of (3) can be obtained with a conjugate gradient (CG) algorithm and its matrix-free version will be explained in the next section.

2.2. Heat source model

For the part-scale simulation to be affordable from a computational point of view, the scanning strategy is neglected and a uniform volumetric heat generation is applied to an entire layer during the heating time. From the laser speed, hatching space, beam spot size and the layer dimensions, one can recover the heating time fro a specific layer. In the first case that will be presented, the build is a simple rectangle so the layer dimensions are constant during the simulation. It corresponds for example to a heating time of 0.9 s. Then, the cooling time is about 9.1 s. The calculation of the heat generation rate starts with the Goldak's double ellipsoid model, commonly employed as a moving heat source in additive manufacturing simulations, which is expressed as :

$$Q_g = \frac{6\sqrt{3}P\eta}{abc\pi\sqrt{\pi}} \exp\left(-3\frac{(x_0 + v_s t - x)^2}{a^2} - 3\frac{(y_0 - y)^2}{b^2} - 3\frac{(y_0 - y)^2}{c^2}\right) \quad (4)$$

where P is the power, η is the absorptivity, v_s is the scan speed. a and b corresponds to the beam spot size and c is the penetration depth, equals to three layers here. The values of this different parameters are summarized in table 1, coming either from settings of the EOS M290 machine or obtained after experiment calibration (absorptivity).

Parameter	Setting	Value
a	Default	50 µm
b	Default	50 µm
c	Default	120 µm
P	Default	285 W
v_s	Default	1 m/s
η	Calibration	0.72

Table 1: Process parameters used in the double ellipsoid model

First, one can find the volumetric heat input (W/m^3) over a scan line by a time integration :

$$\bar{Q} = \frac{1}{\Delta t}\int_0^{t_f} Q_g \, dt = \frac{3P\eta}{\pi bc v_s \Delta t} \exp\left(-3\frac{y^2}{b^2} - 3\frac{z^2}{c^2}\right)$$
$$\left[\text{erf}\left(\frac{\sqrt{3}(x_f - x)}{a}\right) - \text{erf}\left(\frac{\sqrt{3}(x_0 - x)}{a}\right)\right] \quad (5)$$

7

Next, the heat input (W) over a scan line is given by a spatial integration such as:

$$\tilde{Q} = \iiint_{-\infty}^{+\infty} \bar{Q} \, \mathrm{d}x \, \mathrm{d}y \, \mathrm{d}z \approx \int_0^c \int_{-b/2}^{b/2} \int_{-a/2}^{a/2} \bar{Q} \, \mathrm{d}x \, \mathrm{d}y \, \mathrm{d}z$$

$$\approx \frac{P\eta}{4 v_s \Delta t} \mathrm{erf}\left(\sqrt{3}\right) \left[\mathrm{erf}\left(\frac{\sqrt{3}}{2}\right) - \mathrm{erf}\left(-\frac{\sqrt{3}}{2}\right) \right] \left(\int_{-a/2}^{a/2} \left[\mathrm{erf}\left(\frac{\sqrt{3}(x_f - x)}{a}\right) \right. \right.$$

$$\left. \left. - \mathrm{erf}\left(\frac{\sqrt{3}(x_0 - x)}{a}\right) \, \mathrm{d}x \right] \right)$$

(6)

In (6), Δt and $\Delta x = x_0 - x$ accounts for one scan line but as for a rectangular layer, each scan line is identical, the heat input is also identical for the whole layer as the larger distance Δx is integrated over a larger period Δt. The final step to obtain the volumetric heat generation is to divide $\tilde{Q}$ by the volume of one layer such as :

$$Q = \frac{\tilde{Q}}{V_{\mathrm{layer}}}$$

(7)

A numerical application will be provided as the test case configuration will be presented in the dedicated section.

2.3. Matrix-free method

The finite element solution of (1) requires both time and spatial discretization. In (2), the matrix dimensions depends on the number of elements describing the computational domain. Let's consider a 3D dimensional domain where z referred to the build direction (i.e the direction in which the layers are printed). One physical layer is about 40 microns for the L-PBF process so the number of elements in the z-direction is pretty important in the z-direction for a part-scale simulation (build height of few centimeters). Next, the small laser sport size (see table 1) allows the creation of small features up to 100 microns so a large number of elements in the x-y plane is required to take into account such features. The matrix M and K in (2) have a dimension of $N \times N$ where N is the number of nodes to be calculated (only one degree of freedom per node for temperature). As N becomes larger and larger, the amount of memory required to store such matrices become

8

prohibitive even with the use of a sparse matrix format. In order to be able to reduce the limitation on the number of elements, a matrix-free approach will be used for the transient heat transfer analysis. It consists in performing the matrix-vector products involved in the solving of the linear system in (2) a the element level, avoiding the global assembly of the matrices M and K. It has been developed in the 80's for low memory computers and gained in popularity recently with the development of GPU computing. In fact, the use of GPU can speed up some calculations compare to a CPU usage but the GPU memory, about 12 or 24 GB depending on the hardware is quite limited compared to its CPU counterpart. That's why the use of matrix-free method is particularly relevant when associated to GPU computing.

Another burden associated to the calculation of 3d domains is the mesh generation, which can be very time consuming for complex geometries. It is even more true in AM simulations as the mesh needs to be regularly modified, when a new layer has to be computed. In order to simplify the treatment of the mesh, a voxel-based approach will be employed and hexahedral elements will be used. The simplest way to use such regular grid is to use the same element dimension for the whole domain but a set of 2 elements will be used here with the same x-y dimension but a modification of the z dimension for an element in the base plate or in the build part, for an higher efficiency. ADD COARSE/FINE MESH IF WORKING.

In order to solve a general linear system $AX = B$, the conjugate gradient algorithm can be used. A Jacobi preconditionner will also be employed here in order to reduce the computational cost of the solving. It has been chosen for its simplicity and its fast calculation, multi-grid has not been considered here as the improvement brought by the Jacobi preconditionner was acceptable here, additional details will be given in section 3. The preconditionned conjugate gradient (PCG) algorithm is indicated in algorithm 1.

Algorithm 1: Preconditionned Conjugate gradient (PCG) algorithm

---

Input: x_0

$k = 0$, $r_0 = b - \boldsymbol{Ax_0}$, $p_1 = P^{-1}b$, $y_0 = P^{-1}r_0$

while $\|r_k\| < tol$ **do**

$\quad$ $k = k + 1$

$\quad$ $\lambda = \boldsymbol{Ap_k}$

$\quad$ $\alpha_k = \frac{y_{k-1}^T r_{k-1}}{p_k^T \lambda}$

$\quad$ $x_k = x_{k-1} + \alpha_k p_k$

$\quad$ $r_k = r_{k-1} - \alpha_k \lambda$

$\quad$ $y_k = P^{-1} r_k$

$\quad$ $\beta_{k+1} = \frac{y_k^T r_k}{y_{k-1}^T r_{k-1}}$

$\quad$ $p_{k+1} = y_k + \beta_{k+1} p_k$

end

A intermediate matrix vector product λ is computed to avoid its double calculation in α_k and r_k. The different matrix-vector products needed in the PCG algorithm are highlighted in red. The preconditionner P should also be involved in general in some matrix vector products but as the Jacobi preconditionner is used here, P is reduced to a diagonal vector which simplify the calculations associated with it. A detail will now be given on how to compute such matrix vector products with the matrix free method. As a simple illustration, let's consider a 1D FEM problem with 4 elements and 5 nodes and with the element conductivity matrix given by :

$$K_e = \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} \tag{8}$$

The global matrix K would be given by the assembly of K_e for the different elements, such as:

$$K = \begin{bmatrix} 1 & -1 & 0 & 0 & 0 \\ -1 & 1+1 & -1 & 0 & 0 \\ 0 & -1 & 1+1 & -1 & 0 \\ 0 & 0 & -1 & 1+1 & -1 \\ 0 & 0 & 0 & -1 & 1 \end{bmatrix} \quad T = \begin{bmatrix} T_1 \\ T_2 \\ T_3 \\ T_4 \\ T_5 \end{bmatrix} \quad F = \begin{bmatrix} F_0 \\ F_1 \\ F_2 \\ F_3 \\ F_4 \\ F_5 \end{bmatrix} \tag{9}$$

10

By using (9), one can compute the matrix vector product KT. the second component of F for example is given by :

$$F_2 = \underbrace{-1 \times T_1 + 1 \times T_2}_{\text{Element 1}} + \underbrace{1 \times T_2 - 1 \times T_3}_{\text{Element 2}} \tag{10}$$

As F_2 needs some contributions from element 1 and element 2, one can also write (10) in terms of the element conductivity matrices of element 1 and element 2 :

$$F_2 = Ke_1[1][2] \times T_1 + Ke_1[2][2] \times T_2 + Ke_2[1][1] \times T_2 + Ke_2[2][1] \times T_3 \tag{11}$$

In the same way, F_3 can be computed as

$$F_3 = Ke_2[1][2] \times T_2 + Ke_2[2][2] \times T_3 + Ke_3[1][1] \times T_3 + Ke_3[2][1] \times T_4 \tag{12}$$

In (11) and (12) lies the basic principle of the matrix-free method : the global K matrix is not needed for the matrix vector product calculation. The calculation of the matrix product $F = KT$ involves three nested loops:

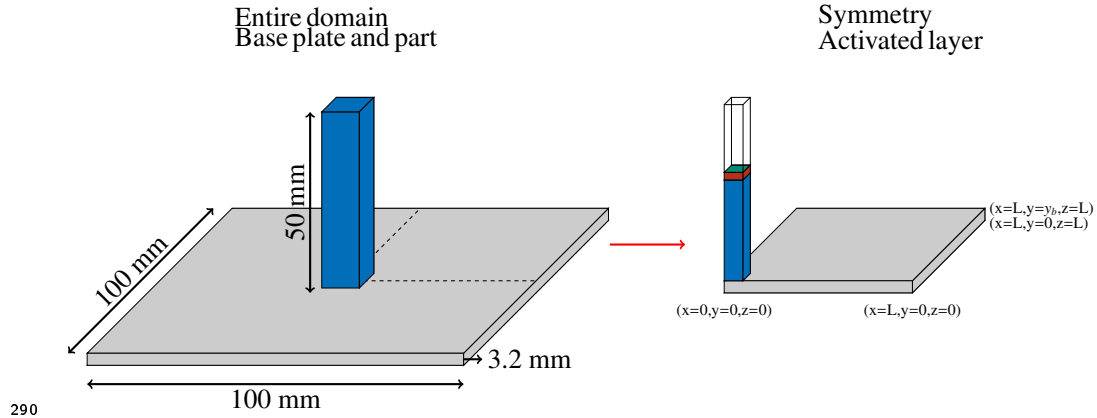1. loop over elements (example : element 2)
2. loop on each row of the Ke matrix (element 2 : row 1 and row 2 will be used respectively to compute F_2 and F_3)
3. loop on each column of the Ke matrix for the nodal contribution (for F_2 (row 1 of Ke), column 1 and 2 will be multiplied respectively by T_2 and T_3)

This procedure implies that the connectivity between elements and nodes is known. In order to facilitate the mesh generation, a structured grid composed of block elements will be used. The height of one element in the build direction is constrained by the size of a physical layer, which is about 40 microns. At the part scale, one can assume that the build height has a similar dimension compared to the length and width of the computational domain. If a 100 mm^3 domain is considered, a block size 40 microns^3 will imply too many elements for the simulation to be computed. So the build direction being along the z-axis, the maximal resolution in the x-y plane is set to 1 mm^2, corresponding to an aspect ratio of $(1, 1, 25)$ for the block element. This latter is composed of 8 nodes so it means that for the matrix-vector calculation, the loops 2 and 3 needs 8 iterations each and so the complete process will needs a number of iterations equal to 64 times the number of elements. It is the most demanding task in the PCG algorithm. An acceptable
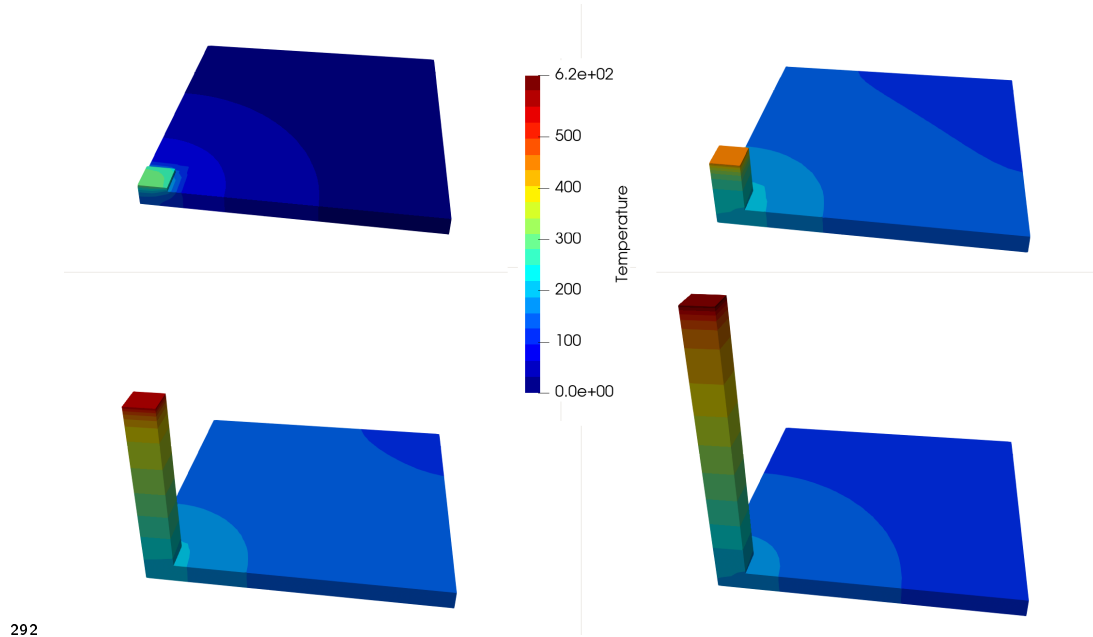
computational time will be obtained with GPU computing. In fact, modern hardware have an important number of CUDA cores and memory bandwidth which can speed up such calculations compared to a CPU version.

3. Results

The configuration of the first case is represented in figure 1.

Figure 1: Configuration of the rectangular column simulation

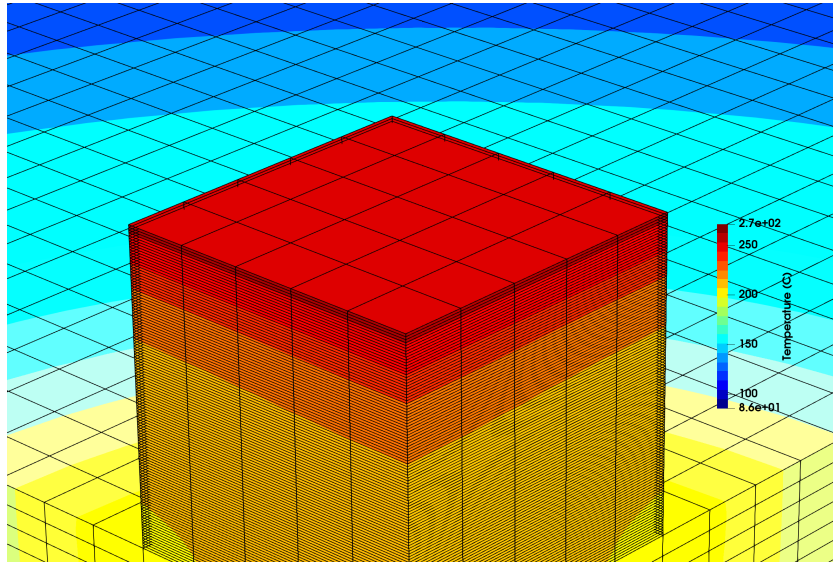Figure 2: Contour plots for the rectangular column at different time steps

13

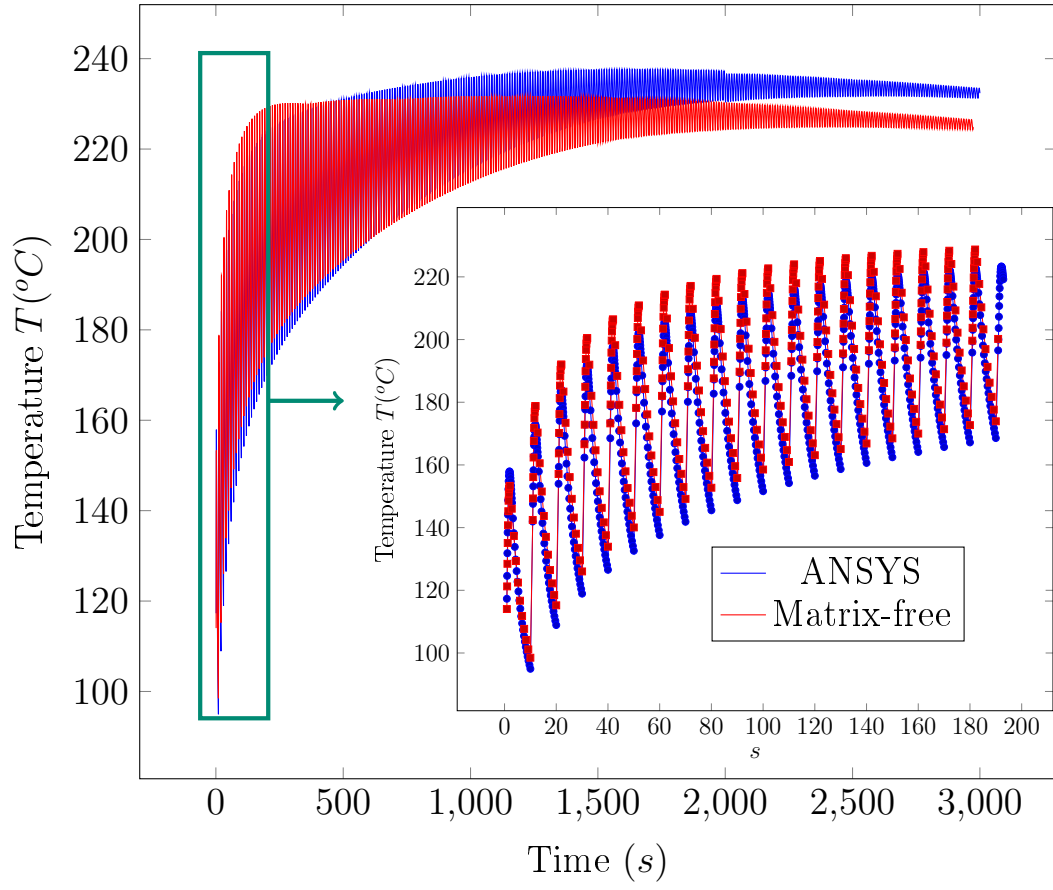Figure 3: Rectangular column : zoom with voxel mesh

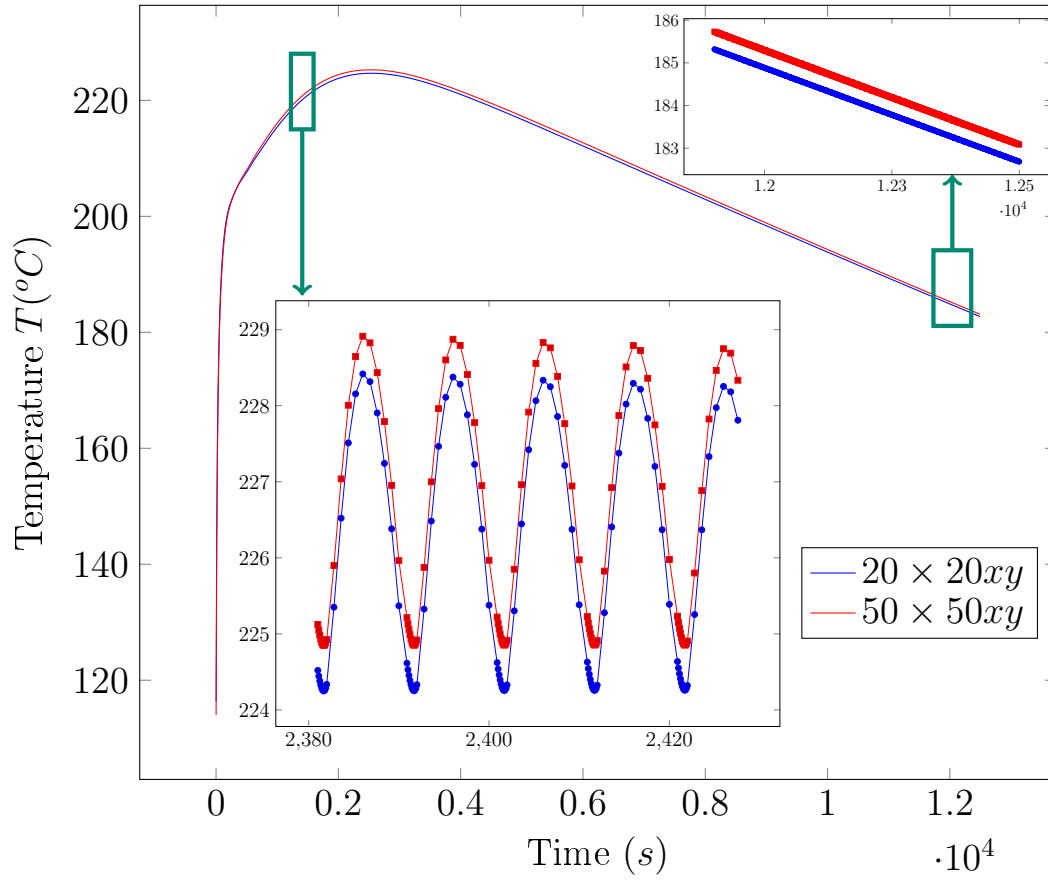Figure 4: Comparison of the far-field temperature between ANSYS and the matrix-free code

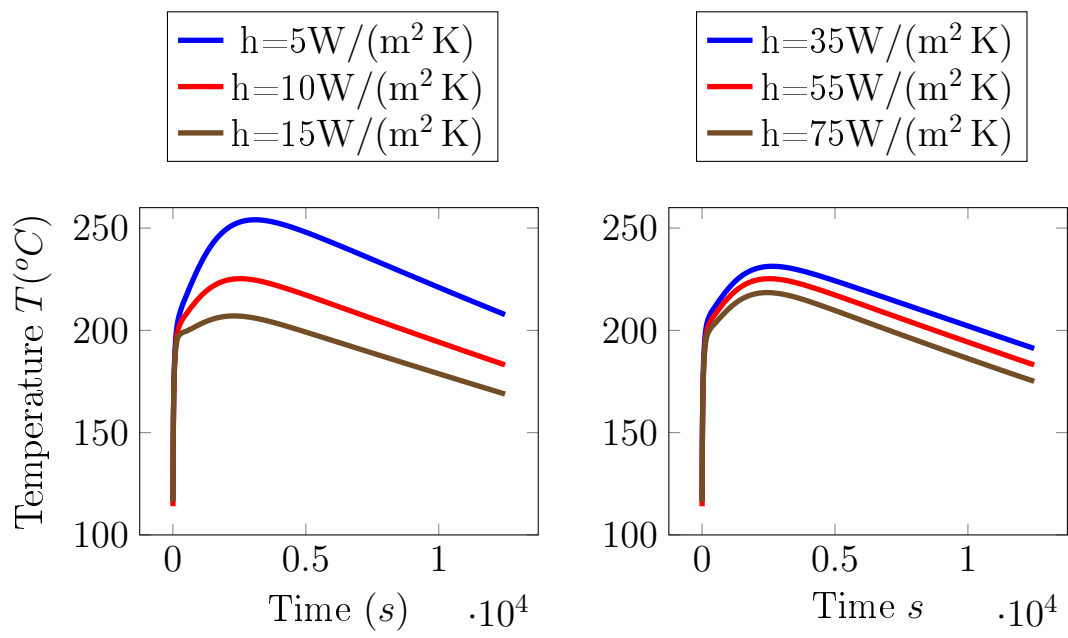Figure 5: Comparison of the far-field temperature for different mesh sizes

Figure 6: Heat transfer coefficient influence

4. Conclusion

References

[1] W. Yan, Y. Qian, W. Ge, S. Lin, W. K. Liu, F. Lin, G. J. Wagner, Meso-scale modeling of multiple-layer fabrication process in Selective Electron Beam Melting: Inter-layer/track voids formation, Materials and Design 141 (2018) 210–219. doi:10.1016/j.matdes.2017.12.031.
URL https://doi.org/10.1016/j.matdes.2017.12.031

[2] K. Dai, L. Shaw, Thermal and mechanical finite element modeling of laser forming from metal and ceramic powders, Acta Materialia 52 (1) (2004) 69–80. doi:10.1016/j.actamat.2003.08.028.

[3] C. Li, M. F. Gouge, E. R. Denlinger, J. E. Irwin, P. Michaleris, Estimation of part-to-powder heat losses as surface convection in laser powder bed fusion, Additive Manufacturing 26 (January) (2019) 258–269. doi:10.1016/j.addma.2019.02.006.

[4] J. Goldak, A. Chakravarti, M. Bibby, A new finite element model for welding heat sources, Metallurgical Transactions B 15 (2) (1984) 299–305. doi:10.1007/BF02667333.
URL https://www.tandfonline.com/doi/full/10.1080/21681805.2017.1363816
http://link.springer.com/10.1007/BF02667333

[5] J. Irwin, P. Michaleris, A Line Heat Input Model for Additive Manufacturing, Journal of Manufacturing Science and Engineering, Transactions of the ASME 138 (11) (2016) 1–9. doi:10.1115/1.4033662.

[6] Q. Chen, X. Liang, D. Hayduke, J. Liu, L. Cheng, J. Oskin, R. Whitmore, A. C. To, An inherent strain based multiscale modeling framework for simulating part-scale residual deformation for direct metal laser sintering, Additive Manufacturing 28 (December 2018) (2019) 406–418. doi:10.1016/j.addma.2019.05.021.
URL https://doi.org/10.1016/j.addma.2019.05.021

[7] M. F. Zaeh, G. Branner, Investigations on residual stresses and deformations in selective laser melting, Production Engineering 4 (1) (2010) 35–45. doi:10.1007/s11740-009-0192-y.

[8] L. Papadakis, G. Branner, A. Schober, K. H. Richter, T. Uihlein, Numerical modeling of heat effects during thermal manufacturing of aero

19

engine components, Lecture Notes in Engineering and Computer Science 3 (2012) 1518–1523.

[9] M. Gouge, E. Denlinger, J. Irwin, C. Li, P. Michaleris, Experimental validation of thermo-mechanical part-scale modeling for laser powder bed fusion processes, Additive Manufacturing 29 (December 2018) (2019) 100771. doi:10.1016/j.addma.2019.06.022.
URL https://doi.org/10.1016/j.addma.2019.06.022

[10] D. Li, W. Liao, N. Dai, G. Dong, Y. Tang, Y. M. Xie, Optimal design and modeling of gyroid-based functionally graded cellular structures for additive manufacturing, CAD Computer Aided Design 104 (2018) 87–99. doi:10.1016/j.cad.2018.06.003.
URL https://doi.org/10.1016/j.cad.2018.06.003

[11] Z. Luo, Y. Zhao, Efficient thermal finite element modeling of selective laser melting of Inconel 718, Computational Mechanics 65 (3) (2020) 763–787. doi:10.1007/s00466-019-01794-0.
URL https://doi.org/10.1007/s00466-019-01794-0

[12] G. F. Carey, B.-N. Jiang, Element-by-element linear and nonlinear solution schemes, Communications in Applied Numerical Methods 2 (2) (1986) 145–153. doi:10.1002/cnm.1630020205.
URL http://doi.wiley.com/10.1002/cnm.1630020205

[13] J. Martínez-Frutos, D. Herrero-Pérez, Efficient matrix-free GPU implementation of Fixed Grid Finite Element Analysis, Finite Elements in Analysis and Design 104 (2015) 61–71. doi:10.1016/j.finel.2015.06.005.
URL http://dx.doi.org/10.1016/j.finel.2015.06.005

[14] B. C. Prabhune, K. Suresh, A fast matrix-free elasto-plastic solver for predicting residual stresses in additive manufacturing, CAD Computer Aided Design 123 (2020) 102829. doi:10.1016/j.cad.2020.102829.
URL https://doi.org/10.1016/j.cad.2020.102829

[15] M. F. Gouge, J. C. Heigel, P. Michaleris, T. A. Palmer, Modeling forced convection in the thermal simulation of laser cladding processes, International Journal of Advanced Manufacturing Technology 79 (1-4) (2015) 307–320. doi:10.1007/s00170-015-6831-x.

20

[16] J. C. Heigel, P. Michaleris, E. W. Reutzel, Thermo-mechanical model development and validation of directed energy deposition additive manufacturing of Ti-6Al-4V, Additive Manufacturing 5 (2015) 9–19. doi:10.1016/j.addma.2014.10.003.
URL http://dx.doi.org/10.1016/j.addma.2014.10.003

[17] M. Chiumenti, E. Neiva, E. Salsi, M. Cervera, S. Badia, J. Moya, Z. Chen, C. Lee, C. Davies, Numerical modelling and experimental validation in Selective Laser Melting, Additive Manufacturing 18 (2017) 171–185. doi:10.1016/j.addma.2017.09.002.
URL https://doi.org/10.1016/j.addma.2017.09.002

[18] D. Logan, First Course in the Finite Element Method, Thomson, 2007.
URL https://books.google.com/books?id=wjr3ArdvAc4C