

Discontinuous Galerkin FEM with Hot Element Addition for the Thermal Simulation of Additive Manufacturing

Björn Nijhuis^{1,a*}, Bert Geijselaers^{1,b}, Jos Havinga^{1,c}
and Ton van den Boogaard^{1,d}

¹University of Twente, Faculty of Engineering Technology, P.O. Box 217, 7500 AE Enschede, The Netherlands

^ab.nijhuis@utwente.nl, ^bh.j.m.geijselaers@utwente.nl, ^cjos.havinga@utwente.nl,
^da.h.vandenboogaard@utwente.nl

Keywords: discontinuous Galerkin method, thermal analysis, directed energy deposition, local time-stepping

Abstract. Despite its promising advantages, the application of directed energy deposition (DED) to produce large metal parts is hindered by challenges inherent to the process. Undesired residual stresses, distortions and heterogeneous material properties mainly originate from a part's thermal history. Fast part-scale thermal models therefore facilitate improved applicability of DED by enabling the prediction and mitigation of these unwanted effects. In this work, the efficiency of a discontinuous Galerkin-based thermal model with heat input by hot element addition, is evaluated and improved to allow such fast simulations. It is found that the model permits the use of a coarse discretization around the heat source, which significantly reduces simulation time while maintaining accurate solutions. It is also shown that the model naturally facilitates the use of local time stepping, which can considerably improve the efficiency of thermal additive manufacturing simulations.

Introduction

Additive manufacturing (AM) of large metal parts using directed energy deposition (DED) is a promising technique that is becoming more and more competitive with conventional manufacturing methods as casting, forging or machining [1]. Despite its potential, DED still faces challenges. These include undesired residual stresses, distortions and non-uniform and anisotropic material properties [2]. The origin of these phenomena lies in the high local heat input of the process and the resulting thermal history of the part [3]. A good understanding of the thermal history can therefore aid in predicting and resolving such unwanted effects. This indicates the need for fast part-scale thermal models to facilitate enhancing the applicability of DED.

The speed of such models can be improved by employing simplified heat input models. Prescribed temperature heat input models remove both the necessity of solving the temperature field around the heat source and the requirement of using a refined mesh in this region, as is the case for the conventionally applied heat input by moving heat flux distributions [4]. Correctly capturing a specified heat input by prescribed temperature heat input models in the commonly used finite element method (FEM) is however difficult [5]. To overcome this issue, the authors have developed a discontinuous Galerkin (DG) based thermal model with heat input by hot element addition [5]. It has been demonstrated that this framework was able to accurately calculate the thermal response of a large metal AM build [5].

The main objective of this work is to show that the DGFEM framework developed in [5] easily allows further speedup of AM thermal simulations. The first contribution is to show that it enables the use of coarse meshes in the deposition zone to yield fast yet accurate simulations. The second contribution is to show that it can naturally accommodate local timestepping (LTS). This increases computational efficiency by allowing different regions in the model to advance in time with different (larger) step sizes. Recent implementations of LTS in AM FE-simulations highlight its potential to speed up such simulations, but also show that these FE-implementations require additional effort to facilitate communication between the different regions [6, 7].

The outline of this paper is as follows. In the *Methodology* section, first the DGFEM framework for thermal simulation of AM will be briefly explained, followed by the concept and implementation aspects of local time-stepping. In the section *Results and Discussion*, two sets of results to highlight the potential of the developed framework for thermal simulation of large-scale metal AM will be presented and discussed.

Methodology

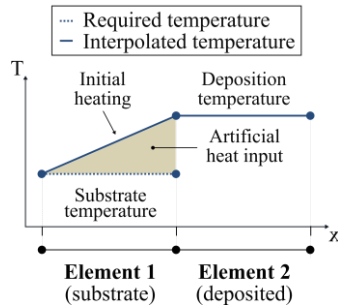
Discontinuous Galerkin FEM. The temperature evolution of a material point is governed by the transient heat equation, which can be written as

$$\begin{aligned} \mathbf{q} &= -\kappa \nabla T, \\ \rho c_p \dot{T} &= -\nabla \cdot \mathbf{q}. \end{aligned} \quad (1)$$

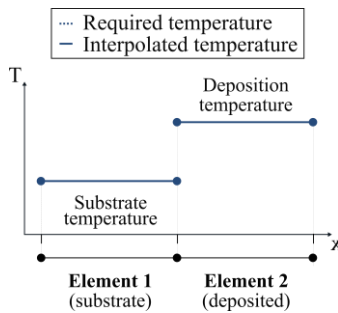
Here, \dot{T} , T and \mathbf{q} denote the temperature rate, temperature and heat flux vector, respectively, and ρ , c_p and κ denote the material's density, specific heat capacity and thermal conductivity, respectively. Eq. 1 can be spatially discretized using the DGFEM as [5]

$$\begin{aligned} \mathbf{M}_i \mathbf{q}_{i,k} &= \kappa_i \mathbf{S}_{i,k} \mathbf{T}_i - \kappa_i \sum_{f=1}^{n_{f,i}} \left[c_{qT,f}^{ii} \mathbf{G}_{k,f}^{ii} \mathbf{T}_i + c_{qT,f}^{ij} \mathbf{G}_{k,f}^{ij} \mathbf{T}_{j_f} \right], \quad k \in (x, y, z), \\ \rho_i c_{p,i} \mathbf{M}_i \dot{\mathbf{T}}_i &= \sum_{k \in (x,y,z)} \left(\mathbf{S}_{i,k} \mathbf{q}_{i,k} - \sum_{f=1}^{n_{f,i}} \left[c_{Tq,f}^{ii} \mathbf{G}_{k,f}^{ii} \mathbf{q}_{i,k} + c_{Tq,f}^{ij} \mathbf{G}_{k,f}^{ij} \mathbf{q}_{j_f,k} \right] \right) - \\ &\quad \sum_{f=1}^{n_{f,i}} \left[c_{TT,f}^{ii} \mathbf{L}_f^{ii} \mathbf{T}_i + c_{TT,f}^{ij} \mathbf{L}_f^{ij} \mathbf{T}_{j_f} \right]. \end{aligned} \quad (2)$$

Here, \mathbf{M}_i , $\mathbf{S}_{i,k}$, $\mathbf{G}_{k,f}^{ii}$, $\mathbf{G}_{k,f}^{ij}$, \mathbf{L}_f^{ii} and \mathbf{L}_f^{ij} are element matrices depending on the element's interpolation order and shape. In this work, linear interpolation is used for both heat flux and temperature rate. $\dot{\mathbf{T}}_i$, \mathbf{T}_i and $\mathbf{q}_{i,k}$ are the nodal temperature rates, temperatures and heat flux components of element i , respectively. $n_{f,i}$ denotes the number of faces of the element. ρ_i , $c_{p,i}$ and κ_i are the element's average density, specific heat capacity and thermal conductivity, respectively. These material properties may be temperature-dependent.

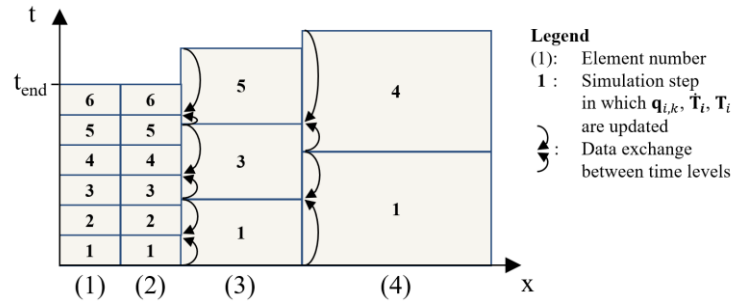


a) FEM

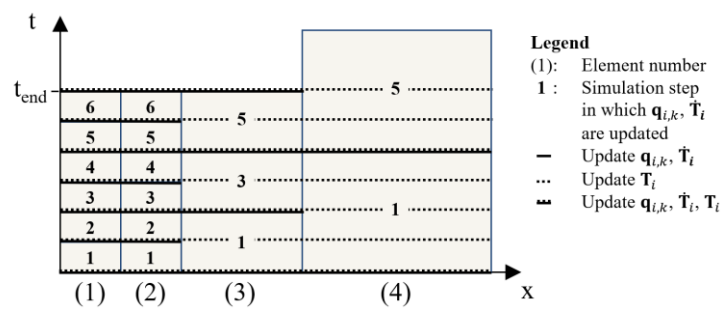


b) DGFEM

Figure 1: Prescribed temperature heat input models



a) LTS without step size rounding and using interpolation



b) LTS with step size rounding and no explicit interpolation

Figure 2: Schematic overview of LTS procedures

In a DGFEM spatial discretization, elements do not share nodes. At every spatial point in a mesh, there may thus exist multiple nodes. Inter-element communication is established using numerical fluxes through element faces f . These fluxes are represented by the terms between square brackets in Eq. 2. Within these terms, nodal quantities with a subscript j_f belong to the neighboring element adjacent to face f . Next to inter-element communication, numerical fluxes can be used to enforce boundary conditions. The type of flux is determined by the choice of the flux parameters c_{qT} , c_{Tq} and c_{TT} . For a detailed overview of the possible choices of these parameters, the reader is referred to [5].

The semidiscrete formulation in Eq. 2 is explicit, so results can be evaluated on an element-by-element basis. This is favorable for additive manufacturing simulations, as it easily allows to perform calculations for active elements only. To retain the explicit nature of Eq. 2 also after temporal discretization, a forward-Euler time-stepping scheme is used to advance the solution from time step n to $n + 1$:

$$\mathbf{T}_i^{(n+1)} = \mathbf{T}_i^{(n)} + \Delta t \dot{\mathbf{T}}_i^{(n)}. \quad (3)$$

Heat input modelling. Heat input modelling and element activation are combined by depositing elements at elevated temperature [5]. To ensure the heat input by a deposited element equals the process heat, its heat capacity can be enhanced according to

$$c_p^* = \frac{\frac{Q}{A} - \int_{T_0}^{T_s} \rho(T) c_p(T) dT}{\int_{T_s}^{T_d} \rho(T) dT}, \quad (4)$$

in between the deposition (T_d) and solidus (T_s) temperature [5]. Here, Q is the heat input per unit bead length, A is the bead's cross sectional area and T_0 the initial substrate temperature. After cooling below solidus temperature, the heat capacity is reset to the physical value $c_p(T)$ again. This heat input method cannot readily be used in a regular FEM model due to interpolation errors resulting from shared nodes at element interfaces, as shown in Fig. 1.

Implementation. The explicit DGFEM formulation as given in Eq. 2-3 allows heat fluxes and temperatures to be calculated in a loop over active elements, while extracting the necessary results of adjacent elements to calculate the numerical fluxes. As such looping and indexing operations are inefficient in MATLAB, global sparse system matrices are set up at the start of a simulation instead. Nodal heat fluxes and temperature rates can then be calculated at once for all elements, according to

$$\begin{aligned} \mathbf{q}_k &= \boldsymbol{\kappa} \circ (\mathbf{C}_{qT,k} \mathbf{T}), \\ \dot{\mathbf{T}} &= \frac{1}{\boldsymbol{\rho} \circ \mathbf{c}_p} \circ (\sum_{k \in (x,y,z)} \mathbf{C}_{Tq,k} \mathbf{q}_k + \mathbf{C}_{TT} \mathbf{T}). \end{aligned} \quad (5)$$

In Eq. 5, the symbol \circ denotes element-wise multiplication. The $n_{\text{dof}} \times 1$ vectors \mathbf{q}_k , \mathbf{T} and $\dot{\mathbf{T}}$ contain the nodal heat fluxes in k -direction, temperatures and temperature rates of all elements, respectively. In the $n_{\text{dof}} \times 1$ vectors $\boldsymbol{\kappa}$, $\boldsymbol{\rho}$ and \mathbf{c}_p the element average thermal conductivities, densities and specific heat values, respectively, are repeated to match the number of nodes in each element. Non-active elements are handled by setting to zero the entries of rows in $\mathbf{C}_{qT,k}$, $\mathbf{C}_{Tq,k}$ and \mathbf{C}_{TT} corresponding to these elements. A sparse diagonal Boolean pre-multiplication matrix is employed for this purpose.

Local time-stepping. The time step in the explicit time integration scheme in Eq. 3 must satisfy the Courant-Friedrich-Lewy (CFL) condition to ensure numerical stability. This condition can be evaluated as [8]

$$\Delta t \leq \min_i (\Delta t_i), \Delta t_i = C \min_{k,T} \left(\frac{1}{\alpha_{i(T)}} (\Delta x_{i,k})^2 \right), k \in (x, y, z). \quad (6)$$

Here, $\Delta x_{i,k}$ is the element size of element i in k -direction, $\alpha = \kappa / (\rho c_p)$ is the thermal diffusivity and C is a constant of order 1 [8]. Eq. 6 implies that the allowable time step size is determined by the smallest element in the mesh.

If there are only a few small elements, the condition in Eq. 6 has a detrimental effect on the overall solution time. This can be alleviated by employing local time-stepping. LTS schemes allow different time steps in different parts of a model. DGFEM readily allows each element to step with its own stable time step due to the fully explicit discretization (Eq. 2-3) and the natural method for inter-element communication through numerical fluxes.

This concept of LTS is shown schematically in Fig. 2a. In every simulation step, the elements with the smallest accumulated time advance one time step. This is repeated until all elements reach the final time. In case information is required from a neighboring element that is not at the same time level, this information is evaluated as function of the neighbor's solution at its previous and/or future time level.

In this work, it is chosen to group elements according to rounded step sizes, with each element's stable step size being rounded down to the nearest integer power of two [9], and use linear interpolation to evaluate data of neighboring elements. This facilitates periodic synchronization of the solution and simplifies the implementation of the LTS scheme as follows:

Consider that from an element j that has stepped from $t_j^{(m)}$ to $t_j^{(m+1)}$, the temperature $\tilde{T}_j^{(n+1)}$ at $t_i^{(n+1)}$ is required for calculations in a neighboring element i . This temperature can be linearly interpolated as:

$$\tilde{T}_j^{(n+1)} = T_j^{(m)} + w_i(T_j^{(m+1)} - T_j^{(m)}), w_i = \frac{t_i^{(n+1)} - t_j^{(m)}}{t_j^{(m+1)} - t_j^{(m)}} = \frac{t_i^{(n+1)} - t_j^{(m)}}{\Delta t_j}. \quad (7)$$

By using Eq. 3 and noting that the term $t_i^{(n+1)} - t_j^{(m)}$ in Eq. 7 always equals an integer multiple n of Δt_i for rounded time step sizes, the expression for $\tilde{T}_j^{(n+1)}$ is written as

$$\tilde{T}_j^{(n+1)} = T_j^{(m)} + n \frac{\Delta t_i}{\Delta t_j} \Delta t_j \dot{T}_j^{(m)} = T_j^{(m)} + n \Delta t_i \dot{T}_j^{(m)}. \quad (8)$$

As a result, no explicit interpolation is required if in *all* simulation steps the temperature of *all* elements is incremented by an amount Δt (Eq. 6) times their last-calculated temperature rate. Element temperatures are thus updated by cheap addition operations in every simulation step, as with global time-stepping (GTS), whereas the expensive calculation of heat fluxes and temperature rates is only performed for element groups that require an update, as in LTS. The resulting LTS scheme is shown in Fig. 2b.

The validity of using LTS in combination with heat input by hot element addition was confirmed by comparing hot element addition onto a one-dimensional rod with non-homogeneous element sizes, to an analytical solution alike the validation performed in [5]. Excellent agreement with the analytical solution was obtained with both GTS and LTS.

Results and Discussion

Speedup by mesh coarsening of deposited elements. In this section, the effect of the mesh size of deposited elements on the temperature evolution is investigated. This provides insight into the behavior of the heat input model under changes in mesh size. To this end, the deposition of a $150 \times 5 \times 8 \text{ mm}^3$ four layer straight thin wall on top of a $150 \times 60 \times 12 \text{ mm}^3$ substrate will be simulated. Symmetry allows only half of the geometry to be modelled. Four different mesh geometries are considered, as shown in Fig. 3. In the most refined mesh, the cross section of a wall layer is modelled with 4×4 elements. In the most coarse mesh, the cross section of a layer is modelled with a single element. The length of the wall elements is equal to 2.083 mm in all simulations.

Process parameters and boundary conditions are taken equal to those in [5, 10]. The process heat input is 270 J/mm, the deposition speed is 8.33 mm/s. Heat is lost to the surrounding air at 25°C through free convection ($h=5.7 \text{ W}/(\text{m}^2\text{K})$) and radiation ($\epsilon=0.2$). Heat is lost to the build plate at 25°C through forced convection ($h=300 \text{ W}/(\text{m}^2\text{K})$). The symmetry face is fully isolated. Temperature-

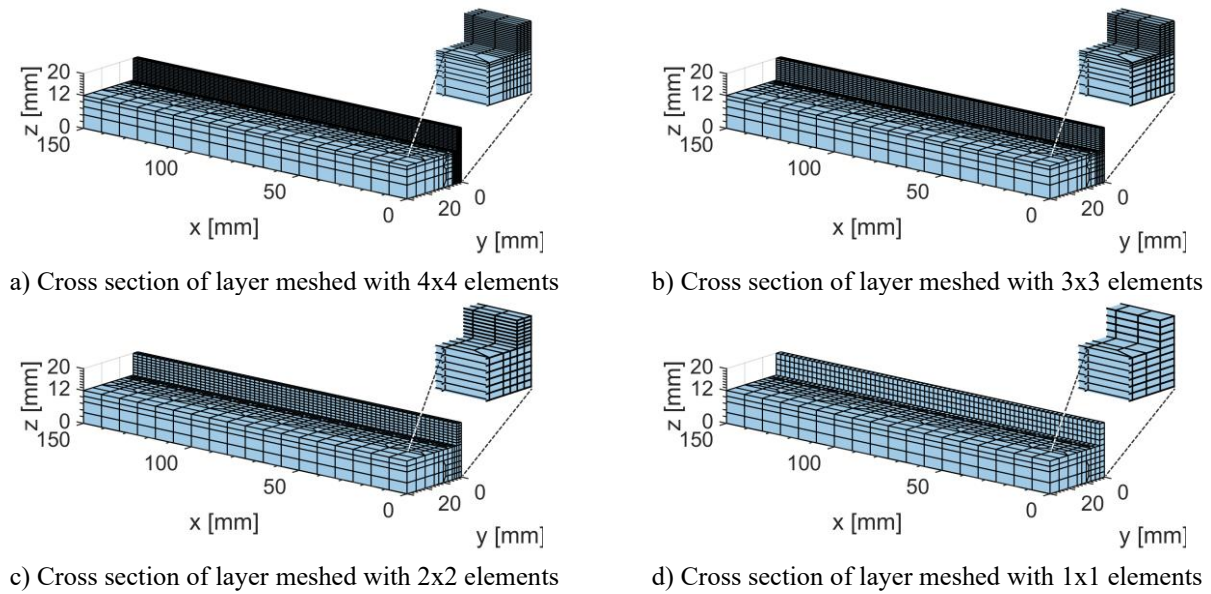


Figure 3: Mesh geometries used to evaluate thermal histories of the short thin wall build

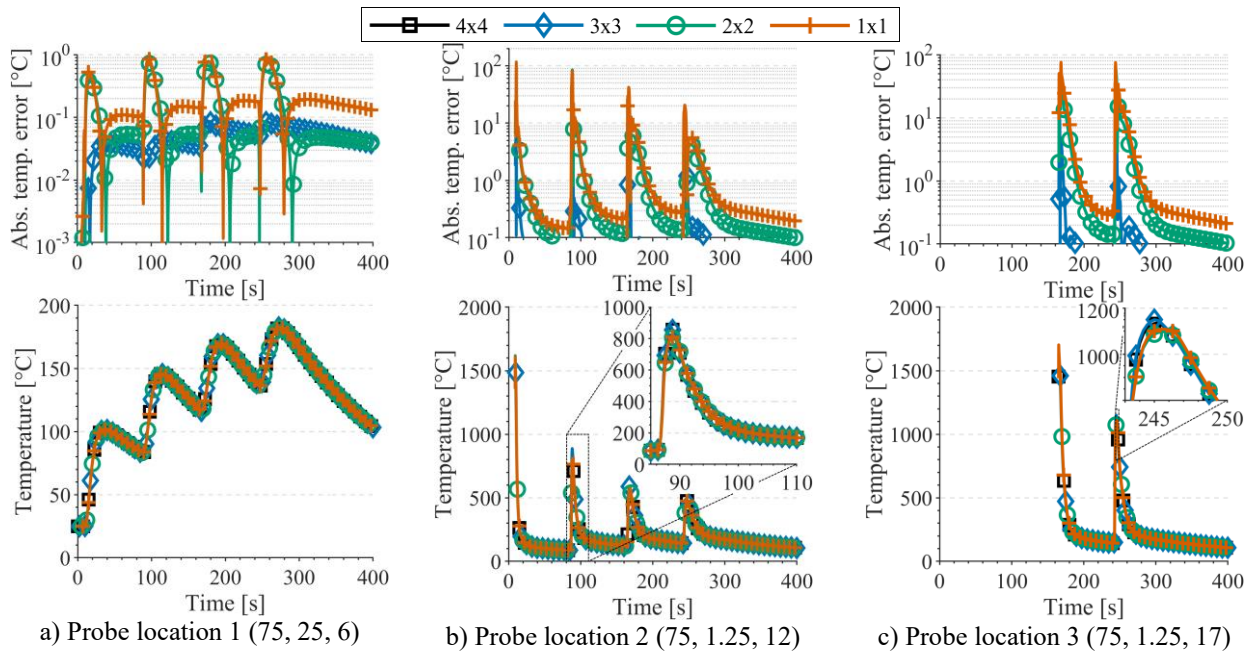


Figure 4: Thermal histories and temperature errors (with respect to finest discretization) evaluated at different locations of the short thin wall build for different mesh geometries

dependent material properties for S355 [5] are used. For these examples, the interlayer time is set to 60 s. Elements are deposited at 2000°C with an enhanced heat capacity of 4538 J/(kgK) to match the process heat [5].

The temperature evolution of three probe points, along with the temperature error, is depicted in Fig. 4. The results of the most refined mesh are taken as reference for the error calculations. It is seen that the overall evolution of the temperature field is the same for all four mesh geometries in both trend and magnitude. For probe location 1 in the substrate, the maximum error is 1.1°C (0.8%) for the coarsest mesh. This indicates that despite the large size of the deposited elements, the process heat input is captured correctly.

The largest errors occur for the coarsest mesh at temperature peaks in the deposited area, with at most 116.5°C (11.8%) for probe location 2 and 75.3°C (8.15%) for probe location 3. Besides peak temperatures, cooling rates between 800°C and 500°C mainly determine the microstructure of steels

[11]. These cooling rates agree very well for both the coarsest and finest mesh, with relative errors below 1%.

The runtimes for the 4 simulations are given in Table 1. All simulations in this work are performed on 2 cores of an Intel® Xeon® Gold 3248R CPU with a clock speed of 3.0 GHz. The runtime has reduced by 96% as a result of coarsening the mesh in the wall. The reason for this is twofold: on one hand the problem size is reduced as the number of elements is reduced, on the other hand the allowable simulation time step size increases due to the increasing element size.

The good representation of the temperature profiles and the fast runtime of the coarsest mesh shows that the DGFEM with heat input through hot element addition can very well be used for optimizing the overall thermal history of the AM process. The good prediction of cooling rates between 800°C and 500°C for the coarsest mesh suggests that an insight into material properties can be obtained in a fast way, but experimental validation is required to determine the effect of errors in peak temperature on these results.

Speedup by local time-stepping. It was reasoned in the *Local time-stepping* section that the presence of a few small elements in the mesh can have a detrimental effect on the simulation runtime. This will be illustrated in this section by considering three models: 1) the wall build with 1x1 elements in a layer's cross-section (Fig. 3d), 2) the wall build with 4x4 elements in a layer's cross-section (Fig. 3a) and 3) a square plate upon which five thin-walled cylinders are deposited (Fig. 5). For each geometry, the distribution of element step sizes is shown in Fig. 6. A comparison of the simulation runtimes with and without LTS is provided in Fig. 7. Timing results are obtained during the first 5000 simulation steps for each model and are averaged over 3 runs. For the cylinder build, process and material parameters and boundary conditions were taken identical to those listed in the previous section.

The achieved speedup versus the theoretically possible speedup is shown in Table 2. The achieved speedup is calculated by comparing the time spent on updating the heat fluxes and temperature rates between simulations with and without LTS. The theoretical speedup is calculated as

$$s_{th} = 100\% \times \sum_g \left(n_g \frac{\Delta t_{max}}{\Delta t_{min}} \right) / \sum_g \left(n_g \frac{\Delta t_{max}}{\Delta t_g} \right), \quad (9)$$

where Δt_{min} and Δt_{max} denote the minimum and maximum stable step size in the model, respectively, Δt_g is the stable step size of element group g and n_g is the number of elements in that group. It is assumed here that the computational costs to perform the calculations in Eq. 2, depend linearly on the number of elements to be updated. Evaluating the values in Table 2 and the histograms in Fig. 6 shows that the largest theoretical speedup is indeed possible for models having a small number of elements with a small stable timestep.

Table 1: Thermal simulation runtimes for various mesh sizes (using GTS)

Cross-section mesh	Step size Δt [s]	Number of elements	Runtime
4x4	2^{-11}	11160	2h 27 m
3x3	2^{-10}	7992	0h 53 m
2x2	2^{-9}	4464	0h 16 m
1x1	2^{-8}	2736	0h 6 m

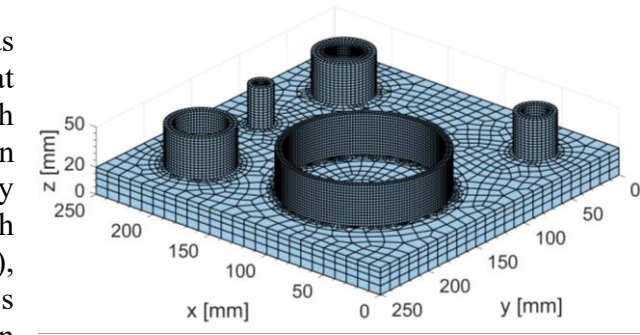


Figure 5: Mesh for model 3

Table 2: Theoretical and realized speedups for different models (based on 5000 simulation steps)

Model	Theoretical speedup (Based on total element count)	Theoretical speedup (Based on element count at simulation start)	Realized speedup
1	160.0%	156.3%	118.7%
2	153.9%	246.9%	154.0%
3	1379.9%	1543.0%	174.2%

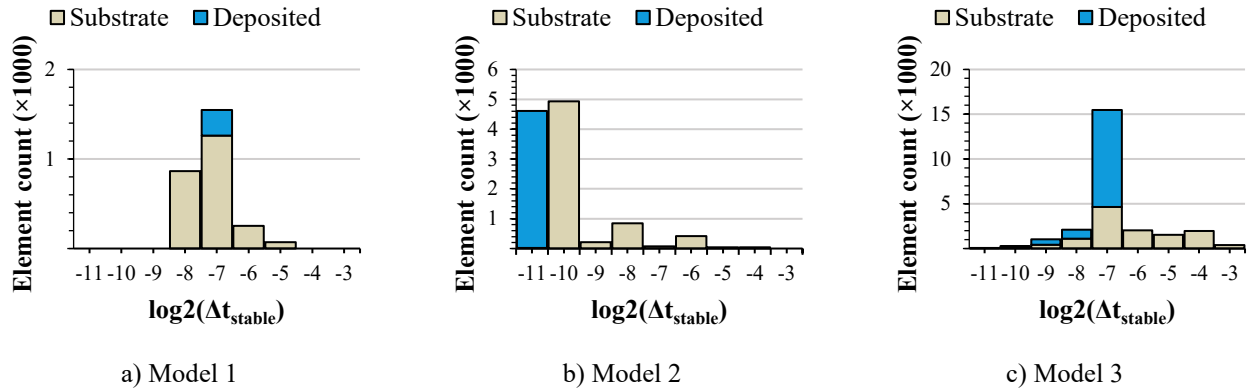


Figure 6: Histograms of (rounded) element stable step sizes for three different models

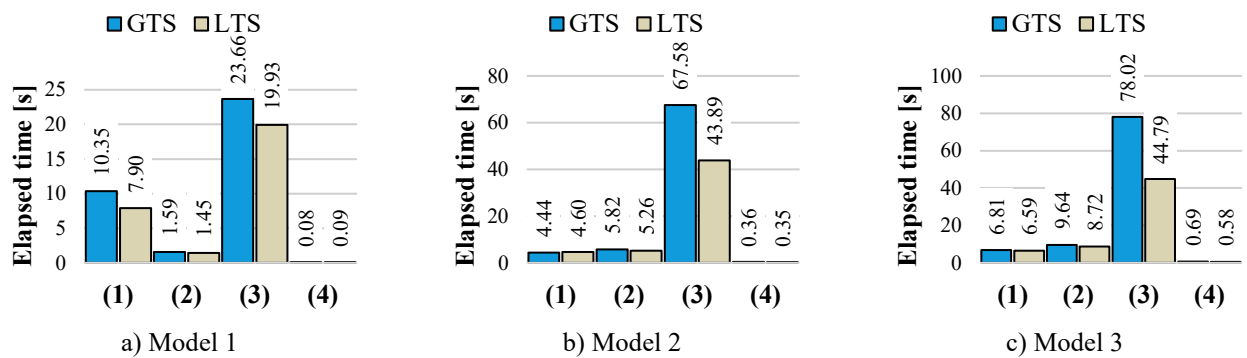


Figure 7: Runtime analysis for three different models. Operations: (1) Element activation, (2) Material parameter evaluation, (3) Heat flux and temperature rate calculations, Eq. 2, (4) Temperature updates, as in Eq. 3.

In the *Local time-stepping* section, it was motivated that a simple implementation of LTS can be formulated if the temperatures of all elements are updated every step. Temperature updates then take place irrespective of whether or not the (interpolated) temperatures are required for calculations. Fig. 7 shows that updating the temperature field takes significantly less time than calculating the nodal heat fluxes and temperature rates. This justifies the current implementation.

Even though employing LTS in the current MATLAB model improves simulation efficiency, the achieved speedups do not approach the estimated values. Updates of individual groups were found to take approximately the same time, irrespective of their element counts. This is caused by the assembly of the system matrices and the way MATLAB handles sparse multiplications. For each separate group g with $n_{\text{dof},g}$ nodal degrees of freedom, the system matrices have size $n_{\text{dof},g} \times n_{\text{dof}}$. The column size n_{dof} is equal for all groups. The computational complexity of sparse multiplications in MATLAB is proportional to this column size [12]. This explains the observed timings. The current calculation routine thus does not scale linearly with the number of updated elements and hence cannot yet utilize the full potential of LTS.

Conclusion and Outlook

From the comparison of the thermal histories resulting from different mesh geometries for the thin wall build, it was shown that increasing the elements size in the deposition area does not significantly affect the thermal history, while significantly reducing computational time. Employing DGFEM in combination with heat input by hot element addition thus allows fast yet accurate thermal simulation of the DED process even when using a coarse mesh for the deposited material.

DGFEM also allows a straightforward implementation of local timestepping, which was theoretically shown to enable substantial computational savings for realistic AM problems. Even though the current MATLAB implementation of the simulation framework is not able to reach the

estimated speedups, preliminary results already indicate the potential for LTS to reduce the overall computational time of DED simulations. Based on the results of this work, it is expected that a further increase in efficiency can be realized if the thermal solver is executed as a compiled and optimized program that avoids the current use of global system matrices.

Altogether, it is concluded that the DGFEM with heat input by hot element addition is a promising alternative to conventional FEM models for simulating the AM process of large metal parts.

Acknowledgements

This research was carried out under project number P16-46/S17024K, which is part of the AiM2XL program framework of the Partnership Program of the Materials innovation institute M2i (www.m2i.nl) and the Netherlands Organization for Scientific Research (www.nwo.nl). The research was conducted in collaboration with industrial partners and supported by the Rotterdam Fieldlab Additive Manufacturing BV (RAMLAB) (www.ramlab.com).

References

- [1] T. Gatsos, K. A. Elsayed, Y. Zhai, and D. A. Lados, Review on Computational Modeling of Process–Microstructure–Property Relationships in Metal Additive Manufacturing, *JOM* 72 (2020) 403-419.
- [2] S. Cooke, K. Ahmadi, S. Willerth, and R. Herring, Metal additive manufacturing: Technology, metallurgy and modelling, *J. Manuf. Process.* 57 (2020) 978-1003.
- [3] T. DebRoy, H. L. Wei, J. S. Zuback, T. Mukherjee, et al., Additive manufacturing of metallic components - Process, structure and properties, *Prog. Mater. Sci.*, 92 (2018) 112-224.
- [4] K. Seleš, M. Perić, and Z. Tonković, Numerical simulation of a welding process using a prescribed temperature approach, *J. Constr. Steel Res.*, 145 (2018) 49-57.
- [5] B. Nijhuis, H. J. M. Geijselaers, and A. H. van den Boogaard, "Efficient thermal simulation of large-scale metal additive manufacturing using hot element addition," *Comp. Struct.*, 245 (2021) 106463.
- [6] D. Soldner and J. Mergheim, "Thermal modelling of selective beam melting processes using heterogeneous time step sizes," *Comp. Math. Appl.*, 78 (2019) 2183-2196.
- [7] L. Cheng and G. J. Wagner, "An optimally-coupled multi-time stepping method for transient heat conduction simulation for additive manufacturing," *Comp. Meth. Appl. Mech. Eng.*, 381 (2021) 113825.
- [8] J. S. Hesthaven and T. Warburton, *Nodal discontinuous Galerkin methods: algorithms, analysis, and applications*, Springer, New York, 2007.
- [9] J. E. Flaherty, R. M. Loy, M. S. Shephard, B. K. Szymanski, et al., Adaptive Local Refinement with Octree Load Balancing for the Parallel Solution of Three-Dimensional Conservation Laws, *J. Parallel Distrib. Comp.*, 47 (1997) 139-152.
- [10] J. Ding, *Thermo-mechanical Analysis of Wire and Arc Additive Manufacturing Process*, PhD Thesis, Cranfield University, Cranfield, 2012.
- [11] L. E. Lindgren, *Computational Welding Mechanics: Thermomechanical and Microstructural Simulations*, Woodhead Publishing Limited, Cambridge, 2007.
- [12] Information on <https://www.mathworks.com/help/matlab/math/sparse-matrix-operations.html>