*Proc. of the International Conference on Electrical, Computer, Communications and Mechatronics Engineering (ICECCME)*
*16-18 November 2022, Maldives*

# An efficient, high-order, adaptive Finite Volume solver for modelling fluid behavior in advanced manufacturing processes

Patrick Zimbrod
*Digital Manufacturing*
*University of Augsburg*
Augsburg, Germany
patrick.zimbrod@uni-a.de

Johannes Schilp
*Digital Manufacturing*
*University of Augsburg*
Augsburg, Germany
johannes.schilp@uni-a.de

***Abstract*—We present a free and open source implementation of the Finite Volume Method that captures thermal and solute-driven free-surface flows, more commonly known as Marangoni flows. This type of physics commonly appears within additive manufacturing and is known to be challenging to resolve. The application is capable of performing adaptive mesh refinement coupled with high order WENO schemes in order to substantially save computational cost in multiphase flows where accurate tracking of the phase boundaries is crucial. By accurately simulating the temporal evolution of melt pools, unique insight into the physics of melt pool formation is possible that is otherwise hard to gain experimentally. We demonstrate our work by a simple two-dimensional benchmark case and outline possible applications with a mesoscopic model of Laser Powder Bed Fusion additive manufacturing.**

***Index Terms*—finite volume method, adaptive refinement, marangoni flow, open source software**

## I. Introduction

Accurate modelling of complex physics has received considerable attention within the last two decades. This can mostly be attributed to the fact that an increased understanding of the phenomena relevant for manufacturing directly leads to more finely tuned or even new processes that tend to increase output quality.

Two such examples that shall serve as a guideline for this work are microfluidic applications and additive manufacturing of metals. Both groups of processes are characterized by complex flow phenomena that involve several physical phenomena on a microscopic scale [1], [2]. The fact that those effects tend to have fast dynamics further complicates the analysis [3]. This leads to an increased amount of empirical effort necessary to capture and quantify the flow patterns involved.

Hence, simulation has gained considerable traction as an alternative to extensive experimental research. Within computational simulation frameworks, the involved physics can be precisely monitored, even to a nanoseconds scale, which would oftentimes be probhibitively expensive. At the same time, a considerable drawback of this approach is the additional amount of work that has to be put into the mathematical modelling of the involved physics. Especially within fluid dynamics, gaining a stable as well as accurate solution can oftentimes be challenging, as the resulting set of partial differential equations can have a small stability region in the temporal domain.

Therefore in this work, we propose and demonstrate a performant, free and open source software framework based on the Finite Volume method that is able to capture complex, non-isothermal surface flows. We discuss the necessary mathematical modelling and briefly address

## II. Theory

The theory of the proposed implementation relies on the Continuum Surface Stress Method initially presented by Lafaurie et al., extending the works on modelling surface tension with the Volume of Fluid Method as described by Brackbill et al. [4], [5]. To describe the necessary linear algebra, we will use the index notation of tensors in a cartesian coordinate system alongside with the Einstein summation convention where the sum over matching indices is implied. First, the capillary stress tensor is introduced:

$$T_{ij} = -\sigma \delta_s (\delta_{ij} - n_i n_j) \tag{1}$$

Where $\sigma$ is the generally non constant coefficient of surface tension. $\delta_s$ is the interface delta function that serves as an indicator function of where the phase boundaries are located. $\delta_{ij}$ is the Kronecker Delta function and $n_i$ is the interface unit normal vector. The local coordinate system of the interface needs to be an orthogonal system in order to separate the purely geometry based capillary force and the tangential Marangoni forces. The resulting force accounting for all surface tension effects can now be expressed as the divergence of the capillary stress tensor, yielding:

$$\frac{\partial T_{ij}}{\partial x_j} = \frac{\partial \sigma}{\partial x_j}[\delta_s(\delta_{ij}-n_i n_j)]+\frac{\partial \delta_s}{\partial x_j}[\sigma(\delta_{ij}-n_i n_j)]+\sigma\delta_s\frac{\partial}{\partial x_j}(\delta_{ij}-n_i n_j) \tag{2}$$

One can show that Eq. 2 can be re-arranged using some tensor algebra and geometric identities into a much shorter and more useful form. The derivation is given in more detail by Lafaurie et al. [4]. By performing those rearrangements, we can recover a form that separates the normal from the tangential components of the divergence vector:

$$\frac{\partial T_{ij}}{\partial x_j} = \frac{\partial \sigma}{\partial x_i}[\delta_s(\delta_{ij} - n_i n_j)] - \sigma\kappa n_i \delta_s \tag{3}$$

Here, the interface curvature $\kappa$ is introduced. The second term of the right hand side corresponds to the normal capillary force directed in the normal direction of the interface, effecting a contractional movement of the interface. The first term resembles the marangoni-type forces present. By taking the derivative of the surface tension coefficient, this term does not vanish iff there are gradients present at the interface. This is normally the case when there are multiple species involved or temperature gradients present [6]. Note that we then simply evaluate the ordinary form of surface tension without any additional physics present. However, as the surface tension coefficient is otherwise not a direct local variable, one must further differentiate the term:

$$\frac{\partial \sigma(c,T)}{\partial x_j} = \frac{\partial \sigma(c,T)}{\partial T}\frac{\partial T}{\partial x_j} + \frac{\partial \sigma(c,T)}{\partial c}\frac{\partial c}{\partial x_j} \tag{4}$$

Additionally, the delta function $\delta_s$ needs to be discretized in a suitable manner in order to capture the physics within the Finite Volume framework. This can be done using the Volume of Fluid Method by taking the gradient of the phase volume fraction $\alpha$ [7], [8]:

$$\delta_s = \left|\frac{\partial \alpha}{\partial x_i}\right| \tag{5}$$

The interface unit normal vector $n_i$ can be computed in a similar manner by using the previously computed interface function:

$$n_i = \frac{1}{\delta_s}\frac{\partial \alpha}{\partial x_i} \tag{6}$$

To perform the finite volume simulations, the Open Source C++ Library OpenFOAM is used in Version 2106. In order to accommodate the extended surface stress formulation, some of the existing solver capabilities need to be modified. The changes necessary are elaborated on in more detail in the following chapter.

## III. Results

### A. Implementation

This foundation is modified and implemented based on the open source library OpenFOAM based on version 2106, which is a popular software library that implements the Finite Volume Method [9]. The modifications rely on the works of Gueyffier et al. in order to account for the additional Marangoni stresses [7].

The proposed code computes the divergence of the capillary stress tensor for finite volume cell centers. However, to handle source terms within the PIMPLE loop, it is necessary to also supply the surface tension force as a scalar interpolated to the cell face. Therefore, an additional method is implemented that computes the $L_2$ norm of the surface tension force vector and interpolates the scalar value via finite differencing.

For powder bed fusion applications where large spacial temperature gradients in the order of $1 \times 10^7 \, \frac{\mathrm{K}}{\mathrm{m}}$ can be present [10], it is advisable to use a second order accurate and stable discretisation scheme for the temperature flux, such as the already within OpenFOAM implemented TVD schemes minmod, superBee or vanLeer [11], [12].

A very useful property of this approach is that not only thermally driven surface tension effects can be incorporated, but also gradients arising from any kind of inhomogeneity. This means that among others solute-driven effects can also be modelled. This scenario is common in manufacturing processes involving mixing of multiple solvents. Within the context of additive manufacturing, in-situ alloying is a scenario where such effects are not negligible [13].

To account for thermal and solute-driven effects simultaneously, we need to incorporate the temperature dependence of the surface tension as well as interfacial tension between the alloying elements in the liquid state. This allows us to capture not only flow patterns on the liquid-gas interface, but also between liquid interfaces of dissimilar species. We may find accurate values for temperature dependent surface tension in the literature. However, determining interfacial tension poses a non-trivial problem that has been discussed in the literature [14].

Naturally, the occurence of interfaces between dissimilar materials is not unique to the setting of in-situ alloying. Mixing of different species is an effect that is oftentimes observed in microfluidic applications as well.

In a pursuit to keep model evaluation simple and not rely on Molecular Dynamics calculations, we use the following, relatively simple approximation for interfacial tension given by Girifalco and Good [15]:

$$\sigma_{1,2} = \sigma_1 + \sigma_1 + 2C(\sigma_1\sigma_2)^{1/2} \tag{7}$$

Here, $\sigma_{1,2}$ denotes the interfacial tension, $\sigma_1$ and $\sigma_2$ are the surface tensions of the involved species and $C$ is a constant which we assume to be unity as a reasonable approximation [14].

In order to approximate the differential terms in the governing equations efficiently, we use high order weighted essentially non-oscillatory (WENO) schemes that have been developed separately for OpenFOAM by Gärtner et al. [16], [17].

TABLE I
MATERIAL PROPERTIES OF THE OSCILLATING DROPLET CASE. THE INDICES INDICATE THAT CORRESPONDING PHASE.

| Quantity | Value [Unit] |
|---|---|
| Density $\rho_1$ | $250\,\mathrm{kg\,m^{-3}}$ |
| Density $\rho_2$ | $2\rho_1$ |
| Heat capacity $c_{p,1}$ | $5 \times 10^{-5}\,\mathrm{J\,kg^{-1}\,K^{-1}}$ |
| Heat capacity $c_{p,2}$ | $2c_{p,1}$ |
| Viscosity $\mu_1$ | $0.012\,\mathrm{kg\,m^{-1}\,s^{-1}}$ |
| Viscosity $\mu_2$ | $2\mu_1$ |
| Thermal conductivity $\lambda_1$ | $1.2 \times 10^{-6}\,\mathrm{W\,m^{-1}\,K^{-1}}$ |
| Thermal conductivity $\lambda_2$ | $2\lambda_1$ |
| Surface tension coefficient $\sigma$ | $0.1\,\mathrm{N\,m^{-1}}$ |
| Marangoni coefficient $\sigma_T$ | $0.02\,\mathrm{N\,m^{-1}\,K^{-1}}$ |

### B. Numerical experiments

First, we examine a well-known setup proposed by Ma et al. that models the flow around a stationary bubble subject to a temperature gradient [18]. We aim to accurately predict the velocity field that arises due to marangoni convection at the interface.

We then proceed to investigate the mesoscopic nanosecond dynamics during selective laser melting of steel, a common problem within additive manufacturing. Here, we show that our implementation is able to handle geometrically and physically complex simulations in 3D.

### C. Marangoni Flow around a droplet

We now investigate a simple, two-dimensional benchmark case that illustrates the ability to capture Marangoni effects for the present work.

An overview of the material parameters is given in table I.

We initialize the droplet in the center of a rectangular domain of length $4a \times 4a$ where $a = 1.44 \times 10^{-3}\,\mathrm{m}$ with a radius of $a$. The droplet is subject to a temperature gradient of $200\,\mathrm{K\,m^{-1}}$ where the bottom temperature is fixed to $290\,\mathrm{K}$.

We discretize the domain using a coarse grid of 64 x 64 quadrilaterals. The spatial discretization of the divergence is done using a mixture of first order bounded Gaussian schemes and third order WENO schemes. We impose no-slip boundary conditions at the walls as well as Neumann type boundaries for the temperature at the left and right walls. The temperature gradient is enforced during the entire simulation using Dirichlet boundaries at the bottom and top. We simulate the evolution of the involved fields up to $t = 0.12\,\mathrm{s}$.

The remainder of the involved numerics and parameters, as well as all boundary conditions can be accessed via the online repository given in section V.

The resulting fields at the final time step $t = 0.12\,\mathrm{s}$ is shown in figure 1. We can clearly observe the field of vortices that forms around the bubble, initiated from varying pressures along the interface. This is due to the Marangoni effect, resulting in an uneven pressure distribution in the domain. Following equation 3, the first term involving the gradient of the surface tension coefficient - i.e. the marangoni coefficient times the gradient of the temperature field - does not vanish here and hence induces a non homogeneous pressure contribution.

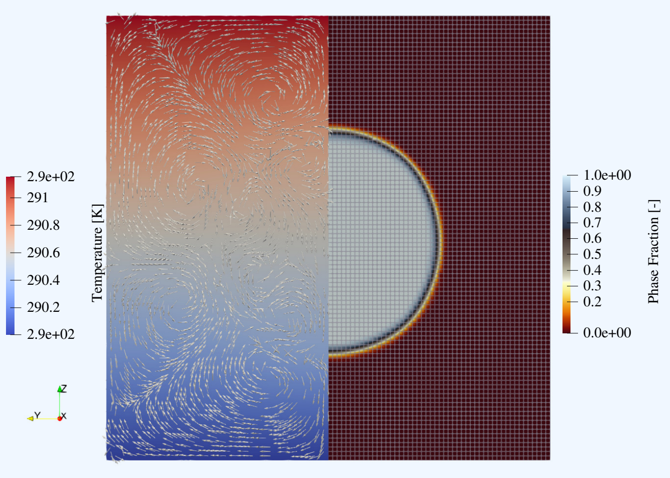

Fig. 1. Flow field around the droplet at final time $t = 0.12\,\mathrm{s}$. The resulting velocity vector field (arrows) and temperature field are shown on the left slice of the domain. The right side shows the phase fraction of the droplet as well as the spatial discretisation.

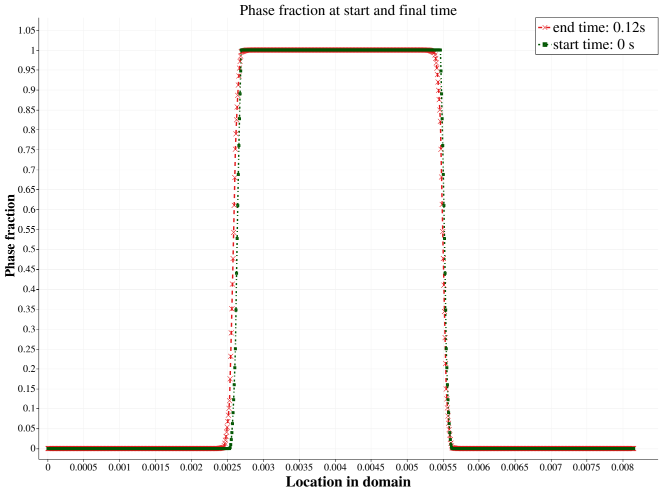

Fig. 2. Phase fraction profiles of the droplet at beginning and end of the investigated temporal domain. This cross-section represents a diagonal slice from the left top to the right bottom of the grid shown in figure 1.

Despite the comparatively low mesh resolution, the phase boundary is conserved reasonably well after solving for 6000 time steps. A comparison of the sharp interface between the two phases at the beginning and end of the simulation is given in figure 2. During the solution process, only minimal numerical diffusion of the interface occurs, indicated by the slightly tapered off edges of the rectangular profile at end time. In addition, we don't observe any oscillatory behaviour in the fields which is essential for obtaining a stable solution.

### D. Powderbed-scale physics during selective laser melting

We now turn to a more complex and relevant simulation setup that appears regularly in additive manufacturing. We

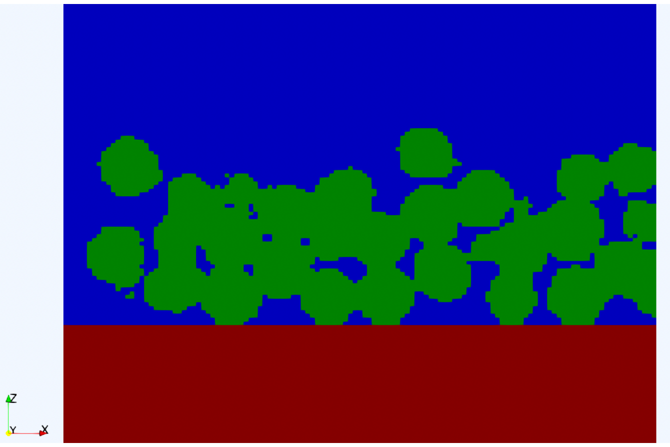

Fig. 3. Composition of the computational domain at the initial time step. Solidified build platform (red), discretised powder bed in solid state (green) and gaseous argon atmosphere (blue).

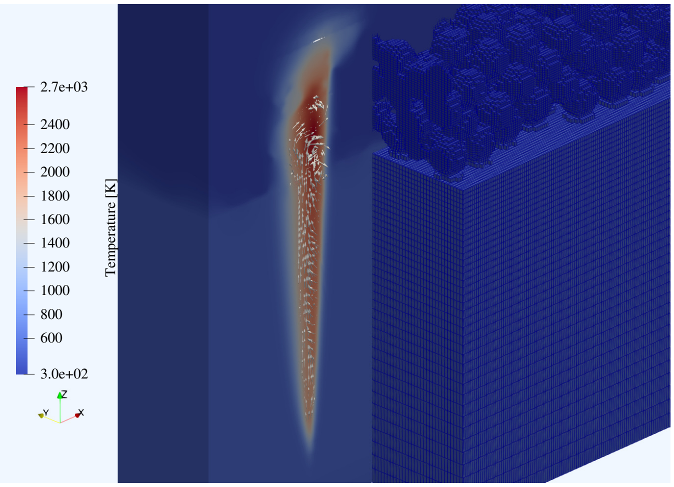

Fig. 4. Temperature field of the investigated Laser Powder Bed Fusion process. The velocity field of the melt pool is indicated by arrows. Additionally, the discretised powder bed is shown along with the computational grid.

aim to investigate the melting, solidification and evaporation behavior of molten metal during the Laser Powder Bed Fusion Process (PBF-LB/M).

This case shall serve as a showcase of a physically complex problem that involves phase changes, simultaneous handling of solid, liquid and gaseous phases (and hence strong pressure gradients) as well as spatially and temporally localized heat transfer.

The computational domain with the initial conditions for the phase fractions is shown in figure 3. We create the packing of the metallic powder particles using a rain drop model implemented in the open source Discrete Element Method software Yade [19].

The physical parameters of the PBF-LB/M model are given in table II. The numerical parameters of the model are given in the case files referenced in section V.

TABLE II
Simulation properties of the Laser Powder Bed Fusion test case. The subscripts l, s and g signify the liquid, solid and gaseous phases involved in the model. [20], [21]

| Symbol | Value [Unit] |
|---|---|
| Laser Power | $300\,\mathrm{W}$ |
| Laser scan speed | $0.9\,\mathrm{m\,s^{-1}}$ |
| Density $\rho_s$ | $7950\,\mathrm{kg\,m^{-3}}$ |
| Heat capacity $c_{p,s}$ | $412\,\mathrm{J\,kg^{-1}\,K^{-1}}$ |
| Thermal conductivity $\lambda_s$ | $1.2 \times 10^{-6}\,\mathrm{W\,m^{-1}\,K^{-1}}$ |
| Enthalpy of fusion $H_f$ | $2.6 \times 10^{5}\,\mathrm{J\,kg^{-1}}$ |
| Density $\rho_l$ | $6881\,\mathrm{kg\,m^{-3}}$ |
| Heat capacity $c_{p,l}$ | $790\,\mathrm{J\,kg^{-1}\,K^{-1}}$ |
| Viscosity $\mu_l$ | $5.85 \times 10^{-3}\,\mathrm{kg\,m^{-1}\,s^{-2}}$ |
| Thermal conductivity $\lambda_l$ | $6.6\,\mathrm{W\,m^{-1}\,K^{-1}}$ |
| Heat capacity $c_{p,g}$ | $1900\,\mathrm{J\,kg^{-1}\,K^{-1}}$ |
| Viscosity $\mu_g$ | $1.8 \times 10^{-5}\,\mathrm{kg\,m^{-1}\,s^{-2}}$ |
| Thermal conductivity $\lambda_g$ | $6.6\,\mathrm{W\,m^{-1}\,K^{-1}}$ |
| Surface tension coefficient $\sigma$ | $1.908\,\mathrm{N\,m^{-1}}$ |
| Marangoni coefficient $\sigma_T$ | $-1.622 \times 10^{-5}\,\mathrm{N\,m^{-1}\,K^{-1}}$ |

## IV. Summary

In this work, we proposed a numerical solver based on the Finite Volume method that is able to incorporate various physics that are relevant for modelling non-isothermal surface flows of arbitrary many phases. We demonstrated the capabilities of the framework using one simple, two-dimensional case showing the effects of marangoni convection and subsequently with a considerably more complex and application-driven process modelling a scenario common in additive manufacturing.

We expect that the results of this work will help to improve understanding of the complex physics in metal-additive manufacturing substantially. Furthermore, the high fidelity data that can be generated using the proposed solver can be used in order to train physics informed machine learning models [22]. It has previously been shown that such models can speed up simulations by order of magnitudes by execution of a simple forward pass. However, accurate and rich training data are needed in order to train such models [23], [24]. We anticipate that this work can help supply this data and further enhance the field of modelling for complex manufacturing processes.

## V. Data Availability

The presented solver is available via Github at pzimbrod/thermocapillaryInterFoam. All data used in this article can be accessed at pzimbrod/ieee-iceccme-2022.

## References

[1] E. K. Sackmann, “The present and future role of microfluidics in biomedical research,” p. 9.

[2] T. DebRoy, H. L. Wei, J. S. Zuback, T. Mukherjee, J. W. Elmer, J. O. Milewski, A. M. Beese, A. Wilson-Heid, A. De, and W. Zhang, “Additive manufacturing of metallic components – Process, structure and properties,” *Progress in Materials Science*, vol. 92, pp. 112–224, Mar. 2018.

[3] S. A. Khairallah, A. A. Martin, J. R. I. Lee, G. Guss, N. P. Calta, J. A. Hammons, M. H. Nielsen, K. Chaput, E. Schwalbach, M. N. Shah, M. G. Chapman, T. M. Willey, A. M. Rubenchik, A. T. Anderson, Y. M. Wang, M. J. Matthews, and W. E. King, "Controlling interdependent meso-nanosecond dynamics and defect generation in metal 3D printing," *Science (New York, N.Y.)*, vol. 368, pp. 660–665, Jan. 2020.

[4] B. Lafaurie, C. Nardone, R. Scardovelli, S. Zaleski, and G. Zanetti, "Modelling Merging and Fragmentation in Multiphase Flows with SURFER," *Journal of Computational Physics*, vol. 113, pp. 134–147, July 1994.

[5] J. Brackbill, D. Kothe, and C. Zemach, "A continuum method for modeling surface tension," *Journal of Computational Physics*, vol. 100, pp. 335–354, June 1992.

[6] J. Straub, A. Weinzierl, and M. Zell, "Thermokapillare Grenzflächenkonvektion an Gasblasen in einem Temperaturgradientenfeld," *Wärme- und Stoffübertragung*, vol. 25, p. 281288, Jan. 1990.

[7] D. Gueyffier, J. Li, A. Nadim, R. Scardovelli, and S. Zaleski, "Volume-of-Fluid Interface Tracking with Smoothed Surface Stress Methods for Three-Dimensional Flows," *Journal of Computational Physics*, vol. 152, pp. 423–456, July 1999.

[8] C. W. Hirt and B. D. Nichols, "Volume of fluid (VOF) method for the dynamics of free boundaries," *Journal of Computational Physics*, vol. 39, pp. 201–225, Jan. 1981.

[9] H. Jasak, A. Jemcov, and Z. Tukovic, "OpenFOAM: A C++ Library for Complex Physics Simulations," *International Workshop on Coupled Methods in Numerical Dynamics*, p. 20, Sept. 2007.

[10] P. A. Hooper, "Melt pool temperature and cooling rates in laser powder bed fusion," *Additive Manufacturing*, vol. 22, pp. 548–559, Jan. 2018.

[11] P. L. Roe, "Characteristic-Based Schemes for the Euler Equations," *Annual Review of Fluid Mechanics*, vol. 18, pp. 337–365, Jan. 1986.

[12] B. van Leer, "Towards the ultimate conservative difference scheme. II. Monotonicity and conservation combined in a second-order scheme," *Journal of Computational Physics*, vol. 14, pp. 361–370, Mar. 1974.

[13] A. Katz-Demyanetz, A. Koptyug, and V. V. Popov, "In-situ Alloying as a Novel Methodology in Additive Manufacturing," in *2020 IEEE 10th International Conference Nanomaterials: Applications Properties (NAP)*, pp. 02SAMA05–1–02SAMA05–4, Nov. 2020.

[14] A. Marmur and D. Valal, "Correlating Interfacial Tensions with Surface Tensions: A Gibbsian Approach," *Langmuir*, vol. 26, pp. 5568–5575, Apr. 2010.

[15] L. A. Girifalco and R. J. Good, "A Theory for the Estimation of Surface and Interfacial Energies. I. Derivation and Application to Interfacial Tension," *The Journal of Physical Chemistry*, vol. 61, pp. 904–909, July 1957.

[16] J. W. Gärtner, A. Kronenburg, and T. Martin, "Efficient WENO library for OpenFOAM," *SoftwareX*, vol. 12, p. 100611, July 2020.

[17] T. Martin and I. Shevchuk, "Implementation and Validation of Semi-Implicit WENO Schemes Using OpenFOAM®," *Computation*, vol. 6, p. 6, Jan. 2018.

[18] C. Ma and D. Bothe, "Direct numerical simulation of thermocapillary flow based on the Volume of Fluid method," *International Journal of Multiphase Flow*, vol. 37, pp. 1045–1058, Nov. 2011.

[19] V. Smilauer, V. Angelidakis, E. Catalano, R. Caulk, B. Chareyre, W. Chèvremont, S. Dorofeenko, J. Duriez, N. Dyck, J. Elias, B. Er, A. Eulitz, A. Gladky, N. Guo, C. Jakob, F. Kneib, J. Kozicki, D. Marzougui, R. Maurin, C. Modenese, G. Pekmezi, L. Scholtès, L. Sibille, J. Stransky, T. Sweijen, K. Thoeni, and C. Yuan, "Yade documentation," Nov. 2021.

[20] J. J. Valencia and P. N. Quested, "Thermophysical Properties," *ASM Handbook*, vol. 15, pp. 468–481, Jan. 2008.

[21] R. F. Brooks and P. N. Quested, "The surface tension of steels," *Journal of Materials Science*, vol. 40, pp. 2233–2238, Jan. 2005.

[22] G. E. Karniadakis, I. G. Kevrekidis, L. Lu, P. Perdikaris, S. Wang, and L. Yang, "Physics-informed machine learning," *Nature Reviews Physics*, vol. 3, pp. 422–440, June 2021.

[23] Z. Li, N. Kovachki, K. Azizzadenesheli, B. Liu, K. Bhattacharya, A. Stuart, and A. Anandkumar, "Fourier Neural Operator for Parametric Partial Differential Equations," *arXiv:2010.08895 [cs, math]*, May 2021.

[24] L. Lu, P. Jin, and G. E. Karniadakis, "DeepONet: Learning nonlinear operators for identifying differential equations based on the universal approximation theorem of operators," *arXiv:1910.03193 [cs, stat]*, Apr. 2020.