


An efficient, h-adaptive Finite Volume solver for advanced quality assessment in metal additive 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 surface flows, more commonly known as Marangoni flows. The application is capable of performing adaptive mesh refinement in order to substantially save computational cost in multiphase flows where accurate tracking of the phase boundaries is crucial. 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

This document is a model and instructions for L^AT_EX. Please observe the conference page limits.

II. THEORY

The theory in this work 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. [?], [?]. This foundation is modified and implemented based on the open source library OpenFOAM based on version 2106. The modifications rely on the works of Gueyffier et al. in order to account for the additional Marangoni stresses [?]. The following presents a brief summary of the modelling involved.

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:

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

Where σ is the generally non constant coefficient of surface tension, δ_s is the interface delta function, δ_{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:

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

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. [?]. By performing those rearrangements, we can recover a form that separates the normal from the tangential components of the divergence vector:

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

Here, the interface curvature κ 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 [?]. 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:

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

Additionally, the delta function δ_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 α [?], [?]:

$$\delta_s = \left| \frac{\partial \alpha}{\partial x_i} \right| \quad (5)$$

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

$$n_i = \frac{1}{\delta_s} \frac{\partial \alpha}{\partial x_i} \quad (6)$$

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

Our implementation of the thermo-fluid-dynamic model is based on the Open Source Software library OpenFOAM. In our proposed implementation, we compute the local gradient of the surface tension explicitly, after it is evaluated for every cell. This means that the gradients with respect to temperature and species concentration are interpolated from the given material properties and the field derivatives are computed separately, unlike Eq. 4 suggests. As a result, the capillary stress is evaluated regardless of the type of Marangoni flow. That is, temperature as well as species gradients or in other effects causing spacial gradients are regarded in this model and contribute to a resulting tangential stress on the interface. Using this approach, the solver configuration remains flexible regarding the future implementation of other or different type physics. However, an appropriate modelling of the surface tension coefficient is required.

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 $1 \times 10^7 \frac{K}{m}$ can be present [?], 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 [?], [?].

In order to resolve the complex wetting phenomena and flow patterns in inhomogenous mixtures requires the modelling of several distinct phases. In our case for Ti6Al4V, this equates to a total of $3n+1 = 10$ phases including the separate gaseous phase to model the shielding gas flow.

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 [?].

Naturally, the occurrence of interfaces between dissimilar metals is nothing unique to the setting of in-situ alloying. This effect should be observable in pre-alloyed powders as well and hence also play a role in Powder Bed Fusion processes in general. However, the mixing paths in this case will be in the range of less than the typical powder diameter and thus in the order of $1 \mu m$ to $10 \mu m$. In-situ alloying however leads to a distribution of alloying species much further apart from each other, i.e. about 10x in comparison. This consequently increases the time the flow takes to homogenize the melt notably and thus should not be considered instantaneous and hence neglected anymore.

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 [?]:

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

Here, $\sigma_{1,2}$ denotes the interfacial tension, σ_1 and σ_2 are the surface tensions of the involved species and C is a constant which we assume to be unity as a reasonable approximation [?].

IEEE conference templates contain guidance text for composing and formatting conference papers. Please ensure that all template text is removed from your conference paper prior to submission to the conference. Failure to remove the template text from your paper may result in your paper not being published.