Received: 30 March 2022 | Revised: 7 September 2022 | Accepted: 22 September 2022

DOI: 10.1002/fld.5151

RESEARCH ARTICLE

A deforming-mesh finite-element approach applied to the large-translation and free-surface scenario of fused deposition modeling

Felipe A. González[1] | **Stefanie Elgeti**[1,2] | **Marek Behr**[1] | **Ferdinando Auricchio**[3]

[1]Chair for Computational Analysis of Technical Systems (CATS), RWTH Aachen University, Aachen, Germany

[2]Institute of Lightweight Design and Structural Biomechanics, Vienna University of Technology, Vienna, Austria

[3]Department of Civil Engineering and Architecture (DICAr), University of Pavia, Pavia, Italy

Correspondence
Felipe A. González, Chair for Computational Analysis of Technical Systems (CATS), RWTH Aachen University, Schinkelstraße 2, 52062 Aachen, Germany.
Email: gonzalez@cats.rwth-aachen.de

Funding information
Comisión Nacional de Investigación Científica y Tecnológica; Deutscher Akademischer Austauschdienst; Ministero dell'Istruzione, dell'Università e della Ricerca, Grant/Award Number: 2017L7X3CS

Abstract

A numerical study of the fused deposition modeling (FDM) process using a boundary-conforming free-surface finite element approach is performed. Due to the complexity of the FDM process, among all of its parts, we focus on the deposition and spreading of an individual filament. The polymer behavior, that is, the shear rate dependent and temperature-dependent viscosity, is included by the Cross-WLF viscosity model. The moving domain is addressed by the virtual region mesh update method, which, in the present article, is extended to free-surface problems. The particularity of dividing the mesh domain into an activated and a deactivated domain makes it possible to handle large translatory mesh deformation. In this work, we make use of the level of detail offered by a boundary-conforming approach regarding both topology accuracy and the imposition of boundary conditions in order to study the deposition of a single filament at a small scale. Parameters with a direct impact on the mechanical properties of the final object can be straightforwardly computed by a boundary-conforming approach, for instance, the cross-section, the contact area, the temperature distribution, and the heat fluxes over the surfaces. The presented approach is validated by a two-dimensional benchmark test case before the numerical results of the three-dimensional simulation of the filament deposition are shown.

KEYWORDS

free-surface, fused deposition modeling, space-time finite element method

1 | INTRODUCTION

Many problems in engineering involve moving boundaries related to free-surface, for example, liquid in storage tanks and manufacturing applications. Among those, additive manufacturing (AM) processes have rapidly gained attention in recent years. Here, the extrusion-based AM process known as fused deposition modeling (FDM) is another excellent example of a free-surface problem involving moving boundaries. In a typical process, a filament of plastic material is introduced into the extruder, where it is heated and melted. Then it is forced to pass through a print nozzle.

After the liquefied material exits the nozzle, it is deposited over a previously printed layer, where it solidifies. Complex 3D objects can be built by this process. While in the early years, FDM was mainly used to produce visual aids or in rapid-prototyping, more recent improvements in the technology have led to a growing number of applications, where FDM is used to manufacture also the final part. A detailed review of FDM techniques can be found in Turner et al.[1]

When treating free-surface problems numerically, the main challenge is the description of a moving interface or boundary for one or multiple immiscible fluids. Two major families of interface descriptions have been developed so far: interface capturing and interface tracking approaches. The first one typically employs an implicit description of the interface. For example, the distribution of a scalar function is used to identify the location of the interface. This kind of approach shows flexibility in defining complex interfaces and topology changes. However, including discontinuous fields is not straightforward, and an adaptive mesh refinement might be needed. Interface capturing methods include the level-set method[2] and the volume-of-fluid (VOF) method.[3] Boundary conforming and interface tracking approaches are based on the explicit description of a surface. In most cases, this approach is combined with a mesh update method to address the mesh adaptation of the moving domain. Boundary-conforming approaches offer the advantage of accurately describing the fluid surface as the boundary of the deforming domain, allowing to easily apply boundary conditions on this surface. There are several ways to address the mesh adaptation. One approach is to perform a global remeshing, changing the nodal and connectivity information. Consequently, a continuous projection of the solution field is required, increasing the computational cost and introducing inaccuracies to the solution. Alternatively, the computational effort can be reduced by retaining either the node or the connectivity information. For instance, methods that only vary the node position are the elastic mesh update method (EMUM)[4] and methods based on radial basis functions.[5] Moreover, methods that apply local modifications to the connectivity information have also been developed, for example, the Shear-Slip method,[6] the Virtual Ring method,[7] and a method based on local mesh optimization.[8]

Regarding FDM process simulation, different efforts have been made to simulate the fluid behavior during all process stages, that is, inside the nozzle, during deposition and cooling, as well as bonding. Bellini[9] studied the liquefier dynamics and the subsequent deposition of a polymer material. A power-law material model addresses the temperature-dependent viscosity, and a global remeshing CFD technique is used in the two-dimensional simulations. Another two-dimensional CFD study that includes the temperature-dependent viscosity was performed by Fitzharris et al.[10] The authors model different parts of the FDM process including multi-filament deposition and cooling, as well as residual stress and warpage, with the aim of studying the effect of material parameters on the warpage. The two previously mentioned studies use a boundary conforming approach based on global remeshing. Both neglect the effect of the surrounding air and focus on the filament shape and temperature distribution. Taking a different route, an immersed interface tracking approach was used by Xia et al.[11] to perform three-dimensional simulations, making an important contribution towards a fully resolved FDM process. The authors model the polymer flow by the Cross-WLF viscosity model and restricted the maximum viscosity at low temperatures. The focus of this article lies on modeling the flow deposition and cooling, showing results of the temperature distribution and the contact area between filaments.

However, also interface capturing methods are commonly used to model FDM processes. For example, three-dimensional FDM simulations using VOF and an isothermal Newtonian fluid model have been studied for a single filament,[12,13] and multi-layer simulation.[14] In Comminal et al.,[12] the authors quantify the effect of the velocity ratio, the gap between filaments, and the cross-section shape. Later the results were compared with experimental measurements,[13] showing agreement for some conditions, and discrepancy for other ones. Further, in Serdeczny et al.,[14] by studying multi-layer depositions, the authors found that both the layer thickness and the distance between filaments have a significant impact on the porosity of the fabricated parts. Behdani et al.[15] studied the FDM process numerically and considered the temperature and the shear-dependent viscosity by using the power-law viscosity model with the Arrhenius correction. Here, a VOF interface capturing approach is employed to perform the three-dimensional simulations. The authors observe that the non-isothermal power-law model shows a better agreement with the experimental data than the Newtonian flow model.

Following the previously mentioned literature, in the present work, we discuss a boundary-conforming numerical model of a single filament in the FDM process. We use the framework of stabilized space-time finite element methods.[16] The deformation of the computational domain is included by the EMUM,[4] which is coupled with the virtual region mesh update method[17] in order to handle the large translatory movement of the filament. The filament surface is described using a boundary-conforming free-surface approach, and the non-Newtonian behavior of the polymer, that is, shear-thinning and temperature-dependent viscosity, is modeled by the Cross-WLF model. We explore the advantages offered by a boundary-conforming approach regarding shape accuracy and the imposition of boundary conditions. The robustness of the presented approach is shown in two numerical examples, a benchmark test case of two-dimensional Couette flow on a moving background mesh and the three-dimensional FDM simulation of the deposition of a single filament. The simulations using a boundary-conforming approach can provide important insights, with respect to the filament cross-section, the contact area, the temperature distribution, and heat fluxes on the surfaces. All of these quantities have a direct impact on the mechanical properties of the final product.

The article is structured as follows. In Section 2, the numerical details regarding the governing equations for the flow and temperature are presented, together with the non-Newtonian viscosity model. In addition, the stabilized space-time formulation is described for the fluid equations. In Section 3, the mesh update method is presented. In particular, a detailed description of the virtual region method and the coupling scheme for the aforementioned models are presented. The numerical results are presented in Section 4, where two cases are studied. First, a two-dimensional Couette flow with a moving background mesh is used to validate the proposed implementation. Then, the results of the three-dimensional simulation of a single filament deposition are shown. Finally, the concluding remarks are given in Section 5.

2 | PHYSICAL MODEL AND DISCRETIZATION

In the present section, we will detail the governing equations for the flow and the temperature as well as the adopted non-Newtonian constitutive model. Furthermore, we will discuss the chosen discretization scheme.

2.1 | Governing equations for the flow and the temperature

A temperature-dependent incompressible viscous fluid is governed by the incompressibility equation, the Navier–Stokes equation, and the heat equation:

$$\boldsymbol{\nabla} \cdot \mathbf{u} = 0 \quad \text{on} \quad \Omega_t, \quad \forall t \in (0, \mathrm{T}), \tag{1}$$

$$\rho \left(\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \boldsymbol{\nabla} \mathbf{u} \right) - \nabla \cdot \boldsymbol{\sigma} = \rho \mathbf{b} \quad \text{on} \quad \Omega_t, \quad \forall t \in (0, \mathrm{T}), \tag{2}$$

$$\rho c_p \left(\frac{\partial T}{\partial t} + \mathbf{u} \cdot \nabla T \right) - \kappa \Delta T - 2\eta \nabla \mathbf{u} : \boldsymbol{\varepsilon}(\mathbf{u}) = 0 \quad \text{on} \quad \Omega_t, \quad \forall t \in (0, \mathrm{T}), \tag{3}$$

where $\mathbf{u}$, p, and T are the velocity, pressure, and temperature fields, respectively; $\mathbf{b}$ collects the body forces, ρ is the density, η the dynamic viscosity, k the thermal conductivity, and c_p the specific heat capacity. The last term of Equation (3) is known as viscous dissipation $\phi = 2\eta \nabla \mathbf{u} : \boldsymbol{\varepsilon}(\mathbf{u})$.

The system of equations is subjected to initial conditions as well as boundary conditions of the Dirichlet and Neumann type, respectively written as:

$$\mathbf{u} = \mathbf{u_0} \text{ on } \Omega_0^f, \quad \mathbf{u} = \mathbf{g}^f \text{ on } \Gamma_g^f, \quad \mathbf{n} \cdot \boldsymbol{\sigma} = \mathbf{h}^f \text{ on } \Gamma_h^f, \tag{4}$$

$$T = T_0 \text{ on } \Omega_0^t, \quad T = g^t \text{ on } \Gamma_g^t, \quad \mathbf{n} \cdot \nabla T = h^t \text{ on } \Gamma_h^t, \tag{5}$$

where Γ_g and Γ_h are complementary parts of Γ, and the superscript f and t represent the flow and temperature, respectively.

For generalized Newtonian fluids, the stress tensor is defined as:

$$\boldsymbol{\sigma} = -p\mathbf{I} + 2\eta(\dot{\gamma}, T)\boldsymbol{\varepsilon}(\mathbf{u}), \tag{6}$$

where the rate of strain tensor is defined as $\boldsymbol{\varepsilon}(\mathbf{u}) = \frac{1}{2}(\nabla \mathbf{u} + \nabla \mathbf{u}^T)$ and the viscosity η is dependent on the shear rate $\dot{\gamma} = \sqrt{2\boldsymbol{\varepsilon}(\mathbf{u}) : \boldsymbol{\varepsilon}(\mathbf{u})}$ and the temperature.

We have chosen to model the non-Newtonian behavior of the polymer melt by the Cross-WLF model. This model considers the shear-thinning behavior and the temperature dependence of the viscosity. Assuming that the infinite-shear viscosity of this model is negligible, the Cross model can be written as:

$$\eta(\dot{\gamma}, T) = \frac{\eta_0}{1 + \left(\frac{\eta_0 \dot{\gamma}}{\tau^*}\right)^{1-n}}, \tag{7}$$

where τ^* is the critical shear stress, η_0 is the zero-shear viscosity, and n is the power-law index.

The temperature dependence is included via the WLF correction, as:

$$\eta_0(T) = D_1 \exp\left[-\frac{A_1(T - T_{\mathrm{ref}})}{A_2 + (T - T_{\mathrm{ref}})}\right], \tag{8}$$

where D_1 is the viscosity at the reference temperature T_{ref}, while A_1 and A_2 are parameters that describe the temperature dependency.

2.2 | Stabilized space-time finite element discretization

The discretization of the governing equation in space and time is based on the DSD/SST space-time finite element formulation.[16] This formulation constructs the weak form on a space-time domain; therefore, the deformation of the spatial domain over time is naturally incorporated in the formulation, and there is no need to modify the governing equations to include it.

The stabilized space-time weak formulation for the Navier–Stokes equations (1) and (2) can be written as follows:
Given $(\mathbf{u}^h)_n^-$, find $\mathbf{u}^h \in \left(\mathcal{S}_{\mathbf{u}}^h\right)_n$ and $p^h \in \left(\mathcal{S}_p^h\right)_n$ such that $\forall \mathbf{w}^h \in \left(\mathcal{V}_{\mathbf{u}}^h\right)_n$ $\forall q^h \in \left(\mathcal{V}_p^h\right)_n$:

$$B^f\left(\left[\mathbf{w}^h, q^h\right]; \left[\mathbf{u}^h, p^h\right]\right) - F^f\left(\left[\mathbf{w}^h, q^h\right]\right) = 0, \tag{9}$$

where

$$\begin{aligned} B^f\left(\left[\mathbf{w}^h, q^h\right]; \left[\mathbf{u}^h, p^h\right]\right) = & \int_{Q_n} \mathbf{w}^h \cdot \rho\left(\frac{\partial \mathbf{u}^h}{\partial t} + \mathbf{u}^h \cdot \nabla \mathbf{u}^h - \mathbf{b}^h\right) \mathrm{d}Q \\ & + \int_{Q_n} \nabla \mathbf{w}^h : \boldsymbol{\sigma}(p^h, \mathbf{u}^h) \mathrm{d}Q + \int_{Q_n} q^h \nabla \cdot \mathbf{u}^h \mathrm{d}Q \\ & + \int_{\Omega_n} (\mathbf{w}^h)_n^+ \cdot \rho\left((\mathbf{u}^h)_n^+ - (\mathbf{u}^h)_n^-\right) \mathrm{d}\Omega \\ & + \sum_{e=1}^{(n_{el})_n} \int_{Q_n^e} \tau_{\mathrm{mom}} \frac{1}{\rho}\left[\rho\left(\frac{\partial \mathbf{w}^h}{\partial t} + \mathbf{u}^h \cdot \nabla \mathbf{w}^h\right)\right] \\ & \cdot \left[\rho\left(\frac{\partial \mathbf{u}^h}{\partial t} + \mathbf{u}^h \cdot \nabla \mathbf{u}^h - \mathbf{b}^h\right) - \nabla \cdot \boldsymbol{\sigma}(p^h, \mathbf{u}^h)\right] \mathrm{d}Q \\ & + \sum_{e=1}^{(n_{el})_n} \int_{Q_n^e} \tau_{\mathrm{cont}}\, \rho \nabla \cdot \mathbf{w}^h \nabla \cdot \mathbf{u}^h \mathrm{d}Q \end{aligned} \tag{10}$$

and

$$F^f\left(\left[\mathbf{w}^h, q^h\right]\right) = \int_{(P_n)_h} \mathbf{w}^h \cdot \mathbf{h}^{f,h} \mathrm{d}P. \tag{11}$$

Likewise, the stabilized space-time formulation for the temperature equation (3), reads:
Given $(T^h)_n^-$, find $T^h \in \left(\mathcal{S}_T^h\right)_n$ such that: $\forall v^h \in \left(\mathcal{V}_T^h\right)_n$:

$$B^t\left(v^h; T^h\right) - F^t\left(v^h\right) = 0, \tag{12}$$

where

$$\begin{aligned}B^t\left(v^h;T^h\right)=&\int_{Q_n} v^h\cdot\rho c_p\left(\frac{\partial T^h}{\partial t}+\mathbf{u}^h\cdot\nabla T^h\right)\mathrm{d}Q\\&+\int_{Q_n}\nabla v^h:\kappa\nabla T^h\mathrm{d}Q-\int_{Q_n}v^h\phi\mathrm{d}Q\\&+\int_{\Omega_n}(v^h)_n^+\rho c_p\left((T^h)_n^+-(T^h)_n^-\right)\mathrm{d}\Omega\\&+\sum_{e=1}^{(n_{el})_n}\int_{Q_n^e}\tau_{\text{temp}}\frac{1}{\rho c_p}\left[\rho c_p\left(\frac{\partial v^h}{\partial t}+\mathbf{u}^h\cdot\nabla v^h\right)\right]\\&\cdot\left[\rho c_p\left(\frac{\partial T^h}{\partial t}+\mathbf{u}^h\cdot\nabla T^h\right)-\nabla\cdot\kappa\nabla T^h-\phi\right]\mathrm{d}Q\end{aligned}\tag{13}$$

and

$$F^t\left(v^h\right)=\int_{(P_n)_h}v^hh^{t,h}\mathrm{d}P.\tag{14}$$

Here, we make use of the following notation:

$$\left(\mathbf{u}^h\right)^{\pm}=\lim_{\epsilon\to 0}\mathbf{u}^h\left(t_n\pm\epsilon\right),\tag{15}$$

$$\int_{Q_n}\dots\ \mathrm{d}Q=\int_{I_n}\int_{\Omega_t^h}\dots\ \mathrm{d}\Omega\mathrm{d}t,\tag{16}$$

$$\int_{(P_n)_h}\dots\ \mathrm{d}P=\int_{I_n}\int_{\Gamma_h}\dots\ \mathrm{d}\Gamma\mathrm{d}t,\tag{17}$$

where $I_n=(t_n,t_{n+1})$ is the time interval, Q_n is the space-time slab, P_n is the space-time surface, and Ω and Γ denote the spatial domain and boundary, respectively. More details about space-time notation can be found in Donea and Huerta.[18] The finite element function spaces for velocity, pressure and temperature ($\mathcal{S}_{\mathbf{u}}^h$, $\mathcal{V}_{\mathbf{u}}^h$, $\mathcal{S}_p^h$, $\mathcal{V}_p^h$, and $\mathcal{S}_T^h$, $\mathcal{V}_T^h$) are based on linear polynomial shape functions and use standard definitions.[18,19]

In Equations (10) and (13), the first three terms correspond to the standard Galerkin weak form, and the fourth term is the so-called jump term, responsible for inducing weak continuity between time-slabs. The remaining terms are added to stabilize the formulation. In particular, the flow equation (10) is stabilized by the SUPG/PSPG terms. Additionally, the consistency can be improved by recovering the second-order terms.[20] The stabilization parameters τ_{cont}, τ_{mom}, and τ_{temp} are based on expressions given by Pauli and Behr.[21] In general, space-time formulations show high order of accuracy in time, even though first-order elements are used to discretize space and time.[18]

The nonlinear systems given by Equations (9) and (12) are linearized using the Newton–Raphson method. The solution to the linear equation system is iteratively approximated with the generalized minimum residual (GMRES)[22] method. For more details about the implementation, see Behr and Tezduyar.[23]

3 | MESH UPDATE METHOD

Boundary-conforming approaches offer the advantage of accurately tracking the fluid surface as the boundary of the deforming domain, allowing to apply boundary conditions on this surface easily. Additionally, moving domain problems require either a mesh update method or a remeshing technique. However, most remeshing techniques show clear disadvantages in terms of accuracy and performance, especially in 3D simulations. In this article, we present a mesh update approach suitable for free-surface problems involving large translational movement.

In particular, all steps of the developed mesh update method are detailed in this section. These are:

- Update the coordinates of the boundary nodes that have been displaced, for example, due to free-surface movement or contact.
- Determine the corresponding displacement of the inner nodes by the EMUM.

- Adapt the mesh using the virtual region method in order to set a well-defined activated domain.

At the end of the section, some details about the coupling problem will be given.

3.1 | Elastic mesh update method

The mesh deformation is performed by the well-known EMUM.[4] This method considers the mesh as an elastic body, which occupies the bounded region $\Omega^{\#}$ with boundary $\Gamma^{\#}$. Therefore, the displacement of the internal nodes $\mathbf{d}^{\#}$ is determined by solving the equations of linear elasticity, as follows:

$$\nabla \cdot \boldsymbol{\sigma}^{\#} = \mathbf{0} \quad \text{on} \quad \Omega^{\#}, \tag{18}$$

where the stress tensor $\boldsymbol{\sigma}^{\#} = \lambda(tr\boldsymbol{\varepsilon}^{\#})\mathbf{I} + 2\mu\boldsymbol{\varepsilon}^{\#}$ is related to the elastic strain tensor $\boldsymbol{\varepsilon}^{\#} = \frac{1}{2}(\nabla\mathbf{d}^{\#} + (\nabla\mathbf{d}^{\#})^{T})$ by the Lamé constants λ and μ.

The system is subjected to Dirichlet and Neumann boundary conditions, written as:

$$\mathbf{d}^{\#} = \mathbf{g}^{\#} \ \text{ on } \ \Gamma^{\#}_{g}, \quad \mathbf{n} \cdot \boldsymbol{\sigma}^{\#} = \mathbf{h}^{\#} \ \text{ on } \ \Gamma^{\#}_{h}. \tag{19}$$

Note that in the case considered in this work, the Dirichlet boundary condition is exactly the displacement of the surface nodes due to free-surface evolution or contact.

The steady equilibrium equation is solved by the standard Galerkin finite element method. It is a normal procedure to drop the Jacobian from the weak form in order to increase the stiffness of small elements, resulting in smaller distortion at those elements. As a consequence, the mesh quality is preserved where the mesh is finer, which usually is close to interfaces or moving boundaries.

It should be noted that the domain of the discretized EMUM problem $\Omega^{\#}$ is only the upper level of the space-time domain Q_n introduced earlier.

For more details, see the original work,[4] or more update.[17]

3.2 | Free-surface

The mass conservation property over the free-surface boundary Γ_{fs} translates into the kinematic condition:

$$\mathbf{v}^{\#} \cdot \mathbf{n} = \mathbf{u} \cdot \mathbf{n} \quad \text{on} \quad \Gamma^{\#}_{fs}, \tag{20}$$

where $\mathbf{v}^{\#}$ is the displacement velocity and u is the fluid velocity. Note that Equation (20) only restricts the normal component of the velocity. Different possibilities for the tangential velocity that fulfills the kinematic condition have been studied.[24,25] In the present work, a Lagrangian approach is chosen, that is, $\mathbf{v}^{\#} = \mathbf{u}$, since it is the most appropriate choice considering the translational nature of the filament motion. The mesh displacement $\mathbf{d}^{\#}$ is computed from $\mathbf{v}^{\#}$, and then, it is imposed on the mesh equation as a Dirichlet boundary condition (19).

3.3 | Contact problem

For the contact problem, we take advantage of the boundary-conforming approach; in fact, we use the same strategy as in Lagrangian-formulation solid mechanics. After describing the contact detection approach, the contribution to each governing equation is detailed in the following.

In the present work, the main body is the fluid, which is deformed due to a free-surface motion. The second body is a wall which is defined by a plane that the fluid cannot cross. In particular, the wall is not described by any separate mesh.

The contact detection approach is described as follows. First, a projection point $\overline{\mathbf{x}}$ on the wall surface Γ_w with minimal distance to the free-surface is computed:

$$\overline{\mathbf{x}} = \underset{\mathbf{x}_w \in \Gamma_w}{\arg\min} \, ||\mathbf{x}_w - \mathbf{x}||. \tag{21}$$

Then, the penetration gap is defined as the normal component of the minimal distance:

$$g_n = (\bar{\mathbf{x}} - \mathbf{x}) \cdot \mathbf{n}. \tag{22}$$

Finally, penetration occurs when $g_n < 0$.

Once the contact is enabled, the next step is to include the corresponding contribution to each governing equation, that is, elastic mesh, Navier–Stokes, and heat equations.

The contribution to the mesh problem is the normal penetration gap $\mathbf{g}_p = g_p \cdot \mathbf{n}$, which is included directly as a Dirichlet boundary condition (19) on the portion of the boundary in contact $\Gamma_c^{\#}$:

$$\mathbf{d}^{\#} = \mathbf{g}_p \quad \text{on} \quad \Gamma_c^{\#}. \tag{23}$$

It is important to note that no portion of the mesh belongs to the contact boundary before the contact is enabled. However, when the contact is triggered, part of the mesh in contact that belongs to the free-surface boundary Γ_{fs} changes to the contact boundary Γ_c. Thus, the corresponding boundary condition on Γ_c is applied.

Next, the contribution to the flow equation is added. Here, a no-penetration boundary condition becomes necessary to prevent the fluid motion when it reaches the wall. In particular, we weakly impose a no-slip boundary condition on the portion of the boundary in contact Γ_c^f. The weak imposition is based on the Nitsche method to incorporate Dirichlet boundary condition, which results in adding the following term to the left-hand side of the flow equation (9):

$$\begin{aligned} B_{wbc}^f\left(\left[\mathbf{w}^h, q^h\right]; \left[\mathbf{u}^h, p^h\right]\right) = &- \int_{(P_n)_c} \mathbf{w}^h \cdot \left(-p^h \mathbf{I} + 2\eta\varepsilon(\mathbf{u}^h) \cdot \mathbf{n}\right) \mathrm{d}P \\ &- \int_{(P_n)_c} q^h \mathbf{n} \cdot \left(\mathbf{u}^h - \mathbf{u}_g^h\right) \mathrm{d}P \\ &- \int_{(P_n)_c} \left(2\eta\varepsilon(\mathbf{w}^h) \cdot \mathbf{n}\right) \cdot \left(\mathbf{u}^h - \mathbf{u}_g^h\right) \mathrm{d}P \\ &- \int_{(P_n)_c} \mathbf{w}^h \cdot \tau_b^f \left(\mathbf{u}^h - \mathbf{u}_g^h\right) \mathrm{d}P, \end{aligned} \tag{24}$$

where the first term is the consistency term, and the next two terms are responsible for the Dirichlet boundary condition. $\mathbf{u}_g$ is the velocity of the wall and τ_b is the weak boundary condition penalty parameter.[26] It is important to mention that this contribution translates into the necessary force to stop the fluid but does not include any impact force. Therefore, this restriction is entirely appropriate for our study case, that is, filament deposition, where restitution and impact forces are negligible.

Lastly, a heat flux is added to the heat equation once the contact is enabled. This heat flux is based on the conductivity heat flux or Fourier's law. Thus, the heat flux on the contact surface Γ_c^T is equivalent to a purely conductive heat flux through the previously printed layer of filaments. It is known that the temperature of the build is constant T_l, but since the upper layer is still in the cooling process, we assume that the constant temperature is reached not before the third layer from the top. Therefore, the Neumann boundary condition reads:

$$q = -k\frac{\Delta T}{\Delta x} \rightarrow \mathbf{n} \cdot \nabla T = -\frac{k}{\Delta x}\left(T - T_l\right), \tag{25}$$

where k is the conductivity of the solid plastic material, and Δx is the distance to reach the constant temperature of the build T_l. In particular, we assumed a value of two times the filament width.

Simulations of the filament deposition over a previously printed layer can be found in Bellini[9] and Fitzharris et al.[10] In the named reference, the authors impose the platform temperature on the bottom surface of the previous layer since they simulated the first two layers only. However, no information about inter-filament heat flux for intermediate build filaments was found in the literature. Therefore, to have more precise heat flux parameters, further studies considering the effect of a larger number of layers, have to be performed, for example, see Sun et al.[27]

3.4 | Virtual region mesh update method

In the framework of boundary conforming approaches, one of the biggest challenges is to simulate problems with large boundary movement since they can easily lead to tangled elements, and therefore, an invalid mesh. In particular, free-surface problems fall into this category, which is precisely the kind of problem that we study in this work. Specifically, we make use of the virtual region mesh update method (VRMUM)[17] to handle problems with large translations of the mesh, avoiding continuous remeshing. Furthermore, this method is combined with the EMUM to achieve a good mesh quality. In Section 4.2, we show the robustness of this method in an FDM simulation.

The virtual region method builds on the idea of having a portion of the mesh deactivated and other activated, where the governing equations are solved only on the activated domain.[7] Therefore, the field domain is only a portion of the whole mesh domain, thus significantly reducing the problem size. This method has been successfully applied to a packaging machine problem[7] and fluid-structure interaction problems.[17] Note that the original method includes the virtual ring technique, which is not addressed here.

The VRMUM is capable of handling large mesh translation by separating the mesh domain from the activated domain, where the field equations are solved. For instance, it allows cells to freely enter or exit the activated domain according to the motion of the background mesh. A sketch of the principle of the virtual region method is shown in Figure 1. A virtual region boundary Γ_{vr} is defined, in order to divide the mesh domain $\Omega^{\#}$ into a deactivated part Ω^{D} and an activated one $\Omega^{f,t}$. Note that the mesh domain is formed by the union of the activated and deactivated domain $\Omega^{\#} = \Omega^{f,t} \cup \Omega^{D}$. Note that the number of virtual boundaries is arbitrary as long as an activated domain is consistently defined.

From the point of view of the assembly of the equation system, only the contribution of activated elements is added to the system, whereas deactivated elements are skipped and do not provide any contribution. Therefore, special treatment is required for the jump term in Equations (9) and (12). This is necessary since no information on the solution vector in the previous time step is available at the moment of activation. For this reason, those elements are not considered in the jump term integral. For more details, see Key et al.[7]

It is important to note that currently, this method is limited to problems with one main direction of motion. Thus, Γ_{vr} is normally chosen perpendicular to that direction. However, the mesh may still deform in other directions.

Generally, VRMUM adapts the mesh to make it suitable to solve a field problem in a portion of it. The adaptation can be broken down into the following steps:

1. Set up the node and element activity distribution according to the activated domain.
2. Deactivate invalid elements.
3. Enforce boundary conformity on Γ_{vr}.
4. Check the Jacobian determinant of elements on Γ_{vr}.
5. Apply boundary conditions on Γ_{vr}.

First, we set up the node and element activity distribution. In the framework of space-time elements, the node activity pattern is based on the upper level only. Then, the same distribution is transferred to the lower level to ensure that both

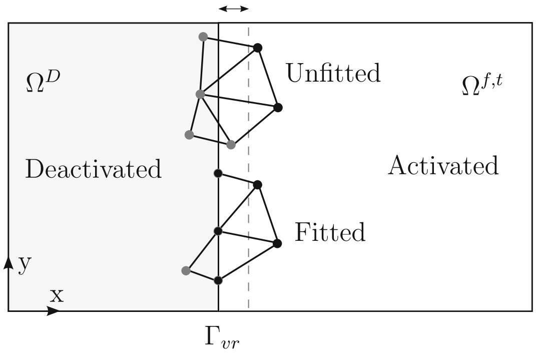

FIGURE 1 Sketch of the virtual region method where it is shown: Activated domain $\Omega^{f,t}$, an example of the fitting process, that is, unfitted and fitted elements, virtual region boundary Γ_{vr}, and tolerance δ to determine the node activity

levels have the same activity pattern. In particular, a node is considered activated if it is located inside of the activated domain. For the node activity only, the position of the virtual boundary is relaxed by a tolerance δ, as shown in Figure 1. The purpose of including this tolerance is to avoid that nodes very close to Γ_{vr} become activated, and therefore, lead to badly shaped elements. For instance, the unfitted portion of the mesh in Figure 1 shows the activated and deactivated nodes in black and gray, respectively. Once the node activity is set up, the element activity is computed. The following criterion is used: Elements with at least one activated node are activated.

In general, the tolerance δ depends on the mesh size and the mesh velocity of the specific problem. In our simulations, values between half of the element size and the element size were used for δ.

After setting up the element activity, we proceed to check the quality of the border elements of the activated domain. Figure 2 shows two examples where enforcing boundary conformity on Γ_{vr} can lead to invalid elements, as previously mentioned by Key et al.[7] To deal with these kinds of elements, we propose to add restrictions to detect these elements before the fitting process is performed, that is, step 3. In Figure 2, the element face of the activated domain border is shown as a thick line. Additionally, the normal vector of the element face $\mathbf{n}_i$ and the normal of the virtual boundary $\mathbf{n}_{vr}$ are also shown. Even though the example is shown for a two-dimensional spatial mesh, the same procedure is valid for three-dimensional spatial meshes.

Thus, we deactivate all elements that fulfill any of the following restrictions:

- Elements with more than one face on the activated border, for example, Figure 2A.
- Elements with face normal $\mathbf{n}_i$ oriented in the opposite direction to the virtual boundary normal $\mathbf{n}_{vr}$, that is, $\mathbf{n}_i \cdot \mathbf{n}_{vr} \leq 0$. For instance, see Figure 2B.

Note that once an element is deactivated, the activated domain border also changes. Consequently, this check is repeated until no elements are deactivated. It should be mentioned that although the checking is performed over every element at the activated border, usually only newly activated elements are deactivated since the other activated elements already fulfilled this condition in the previous time step.

In step 3, the boundary conformity on the activated domain is enforced by a fitting process. This process consists of shifting the closest activated nodes to the virtual region boundary Γ_{vr}, which is illustrated in Figure 1 by the unfitted and fitted state of the mesh. Note that the same procedure is repeated at the lower level of the space-time slab.

Afterward, an extra checking step is included. In particular, the Jacobian determinant is computed on every element at Γ_{vr}. If a tangled element is found, we proceed to deactivate that element, then go back to step 2, and repeat until no tangled elements are found. This second checking stage is performed to ensure that the final fitted mesh is valid. In general, all possibly invalid elements are already deactivated in step 2, and rarely is an element deactivated in this step. It should be mentioned that both checking steps 2 and 4, highly increase the robustness of this method.

Finally, consistent boundary conditions are applied on the virtual region boundary Γ_{vr}, according to the specific field problem.

An important remark is that high Courant numbers are not recommended, since this can lead to a large number of newly activated elements. Furthermore, as was mentioned previously, this can introduce errors to the simulation due to how the jump term is treated. In general, the Courant number is restricted by this mesh update method instead of the space-time finite element formulation, which is known to be unconditionally stable.

It is important to mention that EMUM is computed on the unfitted state of the mesh $\Omega^{\#}$. Therefore, after EMUM is performed, the new unfitted state is stored. Then, VRMUM is applied, and the fitted state of the mesh is computed.

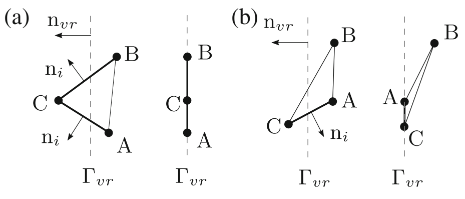

FIGURE 2 Two examples of invalid elements that are deactivated before the fitting process: (A) Collapsed element and (B) inverted element[7]

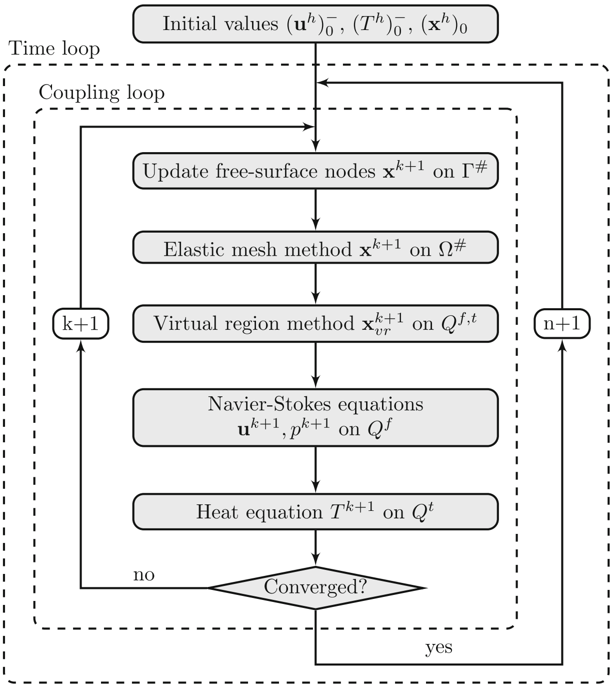
Initial values $(\mathbf{u}^h)_0^-$, $(T^h)_0^-$, $(\mathbf{x}^h)_0$
Time loop
Coupling loop
Update free-surface nodes $\mathbf{x}^{k+1}$ on $\Gamma^\#$
Elastic mesh method $\mathbf{x}^{k+1}$ on $\Omega^\#$
k+1
Virtual region method $\mathbf{x}_{vr}^{k+1}$ on $Q^{f,t}$
n+1
Navier-Stokes equations
$\mathbf{u}^{k+1}, p^{k+1}$ on Q^f
Heat equation T^{k+1} on Q^t
no
Converged?
yes

FIGURE 3 Flowchart of the coupled scheme

3.5 | Coupled problem

All aforementioned equations are coupled strongly in an iterative loop. In particular, Figure 3 illustrates the two main loops of the solution procedure: the time loop and the coupling loop. Here, k and n denote the coupling and time iterator indices, respectively.

In the coupling loop, we first update the mesh boundary nodes on $\Gamma^\#$ according to the free-surface motion (20) or contact (22). If contact is enabled, the corresponding contribution is added to each governing equation.

Subsequently, the displacement of inner nodes of the mesh is computed by solving EMUM (18). As was mentioned before, EMUM is computed on the unfitted state of the mesh. Thus, before and after EMUM, the unfitted state of the mesh is restored and stored, respectively.

Afterward, the virtual region mesh update is performed. As was explained in Section 3.4, this method adapts the mesh in order to create a space-time domain $Q^{f,t}$ suitable for solving field problems, that is, flow equations (1) and (2), and heat equation (3). In particular, the same adapted mesh is used for both the fluid and heat problems. In the current implementation, only one Newton iteration is performed per coupling iteration; the coupling loop is repeated until convergence is reached.

4 | NUMERICAL RESULTS

4.1 | Couette flow on a moving background mesh

A Couette flow with a temperature gradient and a moving background mesh is simulated to validate the proposed boundary-conforming approach. Consider an incompressible and Newtonian viscous fluid in a space defined by two parallel surfaces, separated by a distance h. The top surface moves at a constant speed U and is held at temperature T_h. The bottom stays stationary at temperature T_c ($T_h > T_c$), as shown in Figure 4.

Two virtual region boundaries Γ_{vr} are defined in $\Omega^\#$. Together with the top and bottom borders, they enclose a square activated domain $\Omega^{f,t}$. On Γ_{vr}, traction-free condition is imposed on the flow equation and zero heat flux on the heat equation. In addition, an oscillating motion of the background mesh is considered $d^\#(t) = A\sin(\omega t)$, where A is the amplitude of motion, and ω is the angular frequency.

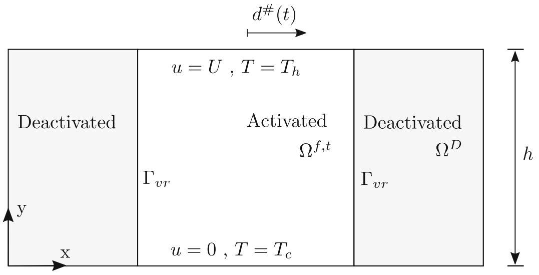

FIGURE 4 Couette flow: Boundary conditions, activated domain, and direction of the background mesh displacement

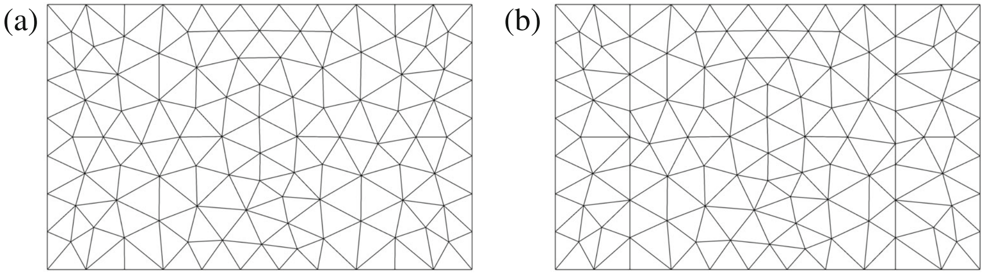

FIGURE 5 (A) Background mesh and (B) the mesh after the fitting process

The analytical solution of the temperature profile in the channel can be written as follows:

$$\frac{T - T_c}{T_h - T_c} = \frac{y}{h} + \frac{\eta U^2}{2k(T_h - T_c)} \frac{y}{h} \left(1 - \frac{y}{h}\right), \tag{26}$$

where y is the coordinate normal to the bottom wall, h is the distance that separates both plates, η is the viscosity, and k is the conductivity.

The following parameters were used in the simulation: velocity $U = 1$, viscosity $\eta = 0.1$, conductivity $k = 1$, height $h = 1$, hot temperature $T_h = 20$, cool temperature $T_c = 10$, displacement amplitude $A = 0.15$, and $\omega = 2\pi/P$ with a period $P = 1$.

The time step was chosen according to a Courant number $C \approx 0.2$, computed with respect to the maximum fluid velocity and the element size.

It is important to note that the viscosity is independent of the shear rate and temperature for Newtonian fluids. Additionally, the mesh motion is independent of the other governing equations. Thus, only a weak coupling scheme is required for this problem.

The initial condition is the steady solution for the static mesh. Then, the unsteady simulation starts, and after a few oscillation periods, a steady state for the moving mesh is reached.

A triangular unstructured mesh was used in the simulations. As an example, the coarsest mesh used is shown in Figure 5A. To illustrate the result of the VRMUM, Figure 5B shows the fitted state of the same mesh.

A mesh convergence study is performed, and the relative error for the temperature with respect to the analytical solution (26) is computed. In this study, we compute the elemental size as the length of the longest edge. Five different sizes are simulated, 0.15, 0.1, 0. 05, 0.025, and 0.01, using the following time steps, 0.03, 0.02, 0.01, 0.005, and 0.002 s, respectively. Figure 6 shows the results of the maximum relative error, once the steady state of the problem is reached. The maximum error is found on the virtual region boundary Γ_{vr} when new elements become activated. This same behavior was previously noted by Key et al.[7]

The relative error for the velocity is not shown since its value remains low for all simulations (order between −14 and −15).

Although VRMUM can introduce errors on Γ_{vr} at the moment of new element activation, the convergence analysis showed that this error quickly decreases with the mesh size.

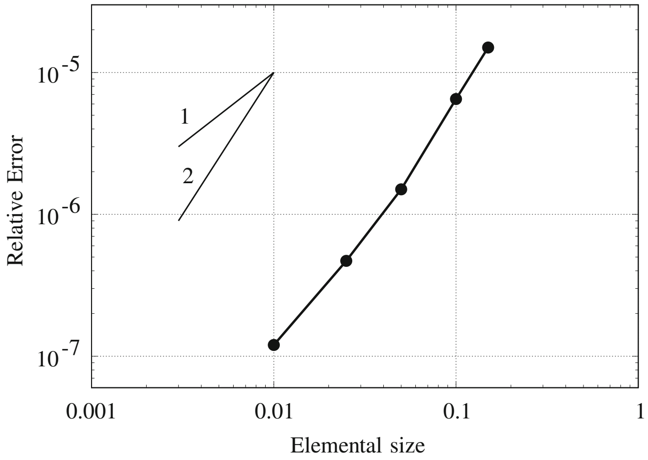

FIGURE 6 Mesh convergence study: Temperature relative error at different elemental sizes

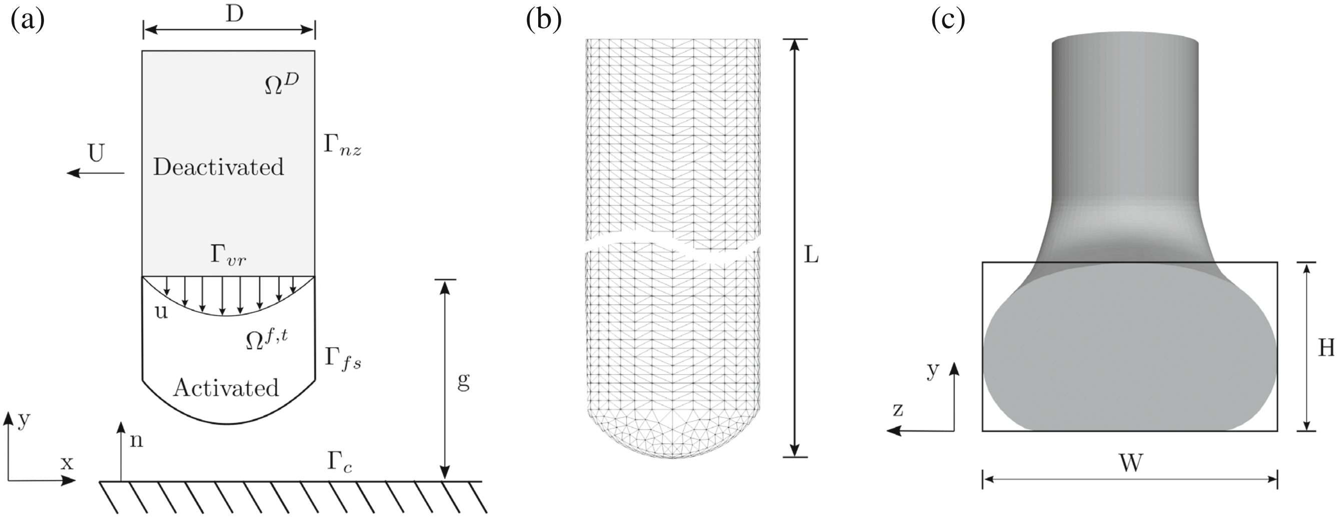

FIGURE 7 Single filament deposition: (A) Sketch of the problem, activated domain, and boundary conditions. (B) Cross-section view of the 3D mesh. (C) Filament cross-section dimensions

4.2 | Single filament deposition

In this section, we will show the result of the simulation of a single filament in an FDM process.

The computational domain is shown in Figure 7. The shape of the three-dimensional mesh is a cylinder with a rounded bottom surface. The mesh domain $\Omega^{\#} = \Omega^{f,t} \cup \Omega^D$ is bounded by the nozzle boundary Γ_{nz} and the free-surface boundary Γ_{fs}, as is shown in the figure. The field equations, that is, flow and heat, are solved on the activated domain $\Omega^{f,t}$, enclosed by the free-surface Γ_{fs} and the virtual boundary Γ_{vr}. The filament length is $L = 8D$, and the other parameters can be found in Table 1. The parameters of the simulation are chosen from the work of Serdeczny et al.,[13] so further comparison can be performed.

A PLA polymer material is considered in the simulation. The Cross-WLF model is used to account for the temperature-dependent viscosity and the shear-thinning behavior of this fluid. Table 2 shows the Cross-WLF parameters.[28] Once the temperature drops below a limit value, the viscosity increases drastically, thus approaching the behavior of a solid body. For the molten polymer, we assume density $\rho = 1240$ kg/m^3, thermal conductivity $k = 0.195$ W/m K, and specific heat capacity $c_p = 2000$ J/kg K. The influence of gravity and surface tension are neglected.

The motion of the nozzle is modeled by including the nozzle speed V as a mesh boundary condition on Γ_{nz}. Therefore, the whole deactivated part of the mesh is moving at the nozzle speed V along the x-direction. In other words, the filament source moves with the nozzle speed.

TABLE 1 Filament extrusion parameters

Parameter	Value
Nozzle diameter, D	0.4 mm
Gap height, g	0.4 mm
Normalized gap height, g/D	1
Flow rate, Q	2.51 mm^3/s
Extrusion average fluid velocity, U	20 mm/s
Nozzle speed, V	20 mm/s
Normalized nozzle speed, V/U	1

TABLE 2 Cross-WLF parameters

Parameter	Value
τ^*	1.009×10^5 Pa
n	0.25
D_1	3.317×10^9 Pa s
T_{ref}	373 K
A_1	20.19
A_2	51.6

TABLE 3 Mesh information

Mesh	Cell number	Min. cell size
Coarse	38,864	6.7 μm
Medium	139,685	4.0 μm
Fine	531,835	2.2 μm

For the flow equation, an input parabolic velocity in y-direction is imposed on Γ_{vr}. This velocity profile is modified by adding a velocity component at the corner nodes of Γ_{vr} to avoid a zero velocity on the free-surface nodes at the nozzle exit. This velocity varies from 30% to 50% of the maximum velocity, but the average velocity U stays constant. The input flow rate $Q = U\pi D^2/4$ is computed from the extrusion average fluid velocity U and the nozzle diameter D. On Γ_{fs}, a traction-free condition is imposed on the flow equation.

For the temperature, a constant temperature $T_h = 473.15$ K is imposed on Γ_{vr}, that is, at the nozzle exit. Once the molten filament exits the nozzle and comes into contact with the surrounding air, it is subject to convective cooling. Here, the surrounding air is not simulated; instead, we impose a convective heat flux directly on the filament border Γ_{fs}. An ambient temperature $T = 293.15$ K is considered, and a convective heat coefficient $h_c = 20$ W/m^2 K as suggested by Bellini.[9]

Note that when contact is enabled, a portion of the free-surface Γ_{fs} becomes Γ_c, and the corresponding boundary conditions are added to the governing equations, as described in Section 3.3. The bottom surface is assumed as planar.

Three mesh sizes are considered for the single filament simulation, coarse, medium, and fine mesh. For example, the 3D medium mesh cross-section is shown in Figure 7B. The space and time domains are meshed by space-time tetrahedron elements with spatial refinement on the boundaries. Elements on the border have an aspect ratio of 5, where the long and short edges are in the flow and transversal directions, respectively. Furthermore, the short edge of border elements is where the minimum cell size of the mesh is found. On the contrary, the maximum cell size is found in the middle part of the mesh, and it is approximately ten times the minimum cell size. The number of cells and the minimum cell size for each mesh are shown in Table 3.

The spatial refinement at the border has two primary purposes: better velocity definition at boundary nodes and small deformation of these elements by EMUM. As was mentioned in Section 3.1, smaller elements are stiffer than the larger ones. Thus, the mesh quality at the free surface is maintained, and the risk of invalid mesh due to distortion of small elements is reduced.

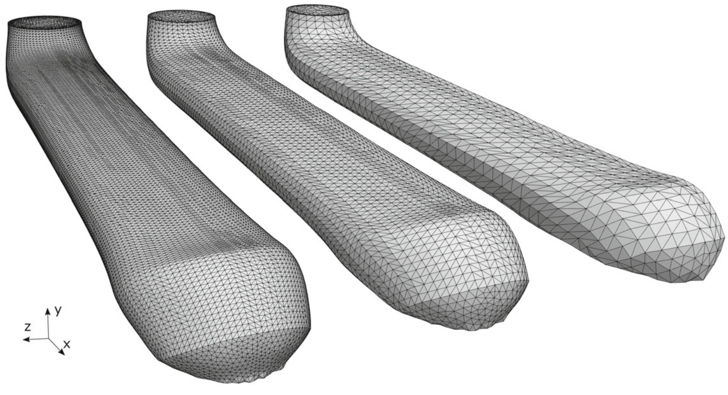

FIGURE 8 View of the three different studied meshes, fine (left), medium (middle), and coarse (right)

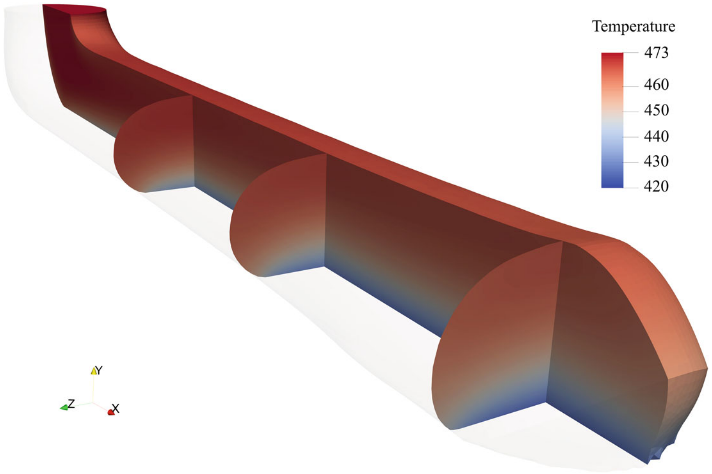

FIGURE 9 Temperature field of the filament inside at time $t = 0.16$ s, including three different cross-section at $x = -0.2$, -1.0, and -1.8 [Colour figure can be viewed at wileyonlinelibrary.com]

The time step is selected according to a Courant number $C < 0.5$. The Courant number is computed with respect to the average input velocity V and the mesh size in the flow direction. Time steps of 1×10^{-3}, 6×10^{-4}, and 4×10^{-4} s are used for the coarse, medium, and fine mesh, respectively.

In Figure 7C, a view of the cross-section of the extruded filament is shown. In particular, the cross-section is measured by the height H and the width W of the extruded filament.

A view of the three studied meshes is shown in Figure 8. One can visualize the mesh deformation of the extruded filament and the mesh quality on the border elements.

The results of the filament deposition simulation for the fine mesh at a time of $t = 0.16$ s is presented in Figure 9. An inside view of the extruded filament allows examining the temperature distribution along the filament and at three different cross-sections. The result shows that the lower temperature is found at the bottom wall, indicating that the heat flux highly influences the cooling of the filament over Γ_{fs}.

The cross-section evolves from a distorted filament at the beginning of the deposition to a rounded oblong shape as a final shape. The filament's distortion can be reduced by lowering the nozzle position, as Bellini[9] documented. It is important to mention the cross-section is directly obtained from the mesh boundary, and no further post-processing computations are required. According to the Cross-WLF model, viscosity increases as the deposited filament cools down, showing a solid behavior when the viscosity is high enough. Therefore, any wetting effect naturally stops, even though surface tension is not considered.

A comparison of the cross-section for different mesh sizes and experimental data[13] is presented in Table 4. The normalized height H/D, the normalized width W/D, and the aspect ratio W/H are computed for the comparison.

During the simulation, one can observe that the predicted height H of the filament is in good agreement with the experimental data. As a result, the values H/D and W/H approach the experimental value as the mesh is refined. This is

TABLE 4 Comparison of height (H/D), width (W/D), and aspect ratio (W/H) of the filament cross-section for different mesh sizes and an experimental result[13]

Results	Coarse	Medium	Fine	Experimental[13]
H/D	0.65	0.69	0.72	0.75
W/D	1.33	1.29	1.25	1.36
W/H	2.04	1.86	1.75	1.81

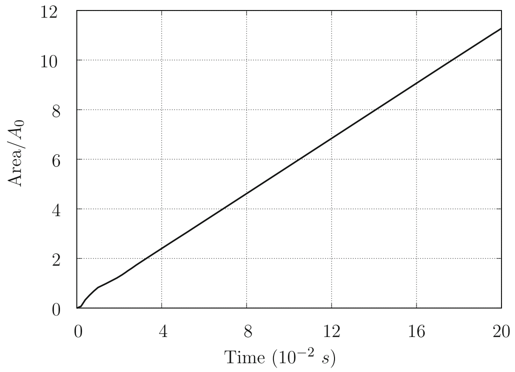

FIGURE 10 Area in contact over time

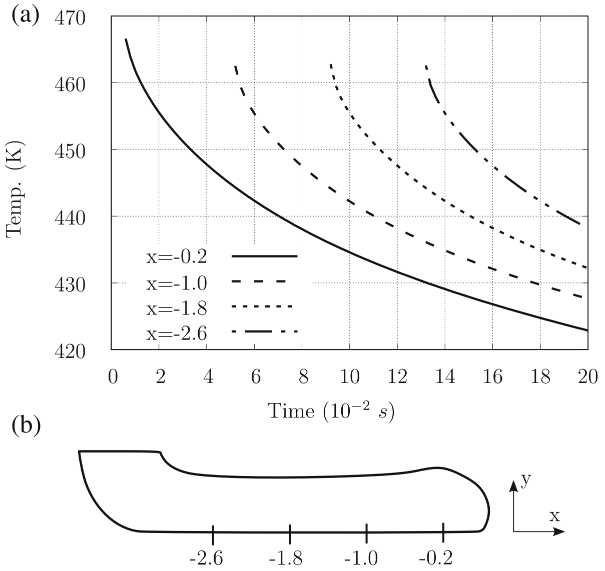

FIGURE 11 (A) Temperature at four fixed control points over time and (B) location of each control point

not true for the width W and therefore also not for the comparative value W/D. We assume that this discrepancy is due to the definition of the heat flux coefficients, which in our simulation were only estimated and are furthermore assumed to be constant. Further studies of these coefficients would improve the numerical results.

Another important measurement that this boundary-conforming approach can also accurately compute is the contact area. In particular, the area of the element faces attached to Γ_c can be directly measured, and therefore, no further post-processing is required. It is well known that the mechanical properties of the final object depend on the inter-filament bonding, and the contact area is one factor of good bonding. Figure 10 shows the evolution of the contact area over time, normalized with the nozzle area A_0. It can be seen that after the filament comes into contact, it quickly reaches a constant rate over time. The same pattern was found by Xia et al.,[29] but only a qualitative comparison is possible since different parameters were used in their simulation.

Figure 11A shows the evolution of the temperature at four evaluation points, and Figure 11B shows the location of each point at the bottom wall. Note that every line starts when the filament comes into contact with the wall at its respective evaluation point, and they are in a range of temperature between 463 and 467 K. Furthermore, they show a common behavior of starting from a high-temperature gradient at contact and then decreasing over time. It is important to mention that the bottom wall temperature depends more on the conductivity heat flux condition imposed on Γ_c, rather than the convective one set on Γ_{fs}.

5 | CONCLUSIONS

A numerical study of a single filament deposition in an FDM process was carried out. This study is in the framework of a stabilized space-time finite element formulation, where a boundary-conforming free-surface approach describes the filament surface. The mesh deformation is performed by the EMUM, which is combined with the virtual region method, in order to handle large mesh displacement. The non-Newtonian behavior—shear-thinning and temperature-dependent viscosity—of the polymer melt is modeled by the Cross-WLF model.

A 2D Couette flow with temperature gradient and a moving background mesh was used as a test case to validate the proposed approach. The results were compared with the analytical solutions, and the effect of a moving background mesh was studied. A mesh convergence study was performed, revealing that even though the virtual region method can introduce a small error at the virtual boundary, this error quickly decreases with the mesh size.

Subsequently, a study of the deposition of a single filament in an FDM context was performed. This problem is numerically challenging since it includes non-Newtonian material behavior, large mesh displacements, and free-surface motion. This work focuses on the behavior and evolution of an individual filament during the deposition, cooling, and spreading of the filament.

It was found that the temperature distribution inside the filament was dependent on the heat flux by conduction and convection, where conduction dominates close to the bottom wall. A comparison of the cross-section using different mesh sizes with experimental results found in the literature was also performed. Overall, a good agreement with the experimental data was found. Although, the main difference with the experimental results is observed at the filament's width, consequently affecting the cross-section aspect ratio. This discrepancy is associated with using approximate and constant values for the heat transfer coefficients. It is known that these values can vary with environmental factors, for example, surrounding airflow and temperature conditions on the platform. So further studies that can provide more precise values or a limited range of values would significantly improve the numerical results.

The filament spreading and later the bonding are phenomena that highly depend on surfaces properties, for example, surface forces, fluxes, and material properties. In this matter, a boundary-conforming approach offers significant advantages, mainly related to shape definition and imposition of surface boundary conditions. In the present work, we introduced the capabilities of boundary-conforming approaches to provide insight into the FDM process.

Finally, the present boundary-conforming free-surface approach can satisfactorily be used to resolve the complex problem of the deposition of a single filament in an FDM process.

ACKNOWLEDGMENTS

The authors gratefully acknowledge the financial support provided by the National Agency for Research and Development (ANID) and the German Academic Exchange Service (DAAD) through the scholarship CONICYT PFCHA/DOCTORADO ACUERDO BILATERAL CONICYT-DAAD/2018-62180006. This work was also partially supported by the Italian Minister of University and Research through the project "A BRIDGE TO THE FUTURE: Computational methods, innovative applications, experimental validations of new materials and technologies" (No. 2017L7X3CS) within the PRIN 2017 program. Open Access funding enabled and organized by Projekt DEAL.

CONFLICT OF INTEREST

The authors declare no potential conflict of interest.

DATA AVAILABILITY STATEMENT

The data that support the findings of this study are available from the corresponding author upon reasonable request.

ORCID

Felipe A. González https://orcid.org/0000-0001-8638-7864

REFERENCES

1. Turner BN, Strong R, Gold SA. A review of melt extrusion additive manufacturing processes: I. Process design and modeling. *Rapid Prototyp J*. 2014;20(3):192-204. doi:10.1108/RPJ-01-2013-0012
2. Sethian JA. *Level Set Methods and Fast Marching Methods: Evolving Interfaces in Computational Geometry, Fluid Mechanics, Computer Vision, and Materials Science*. Vol 3. Cambridge University Press; 1999.
3. Hirt CW, Nichols BD. Volume of fluid (VOF) method for the dynamics of free boundaries. *J Comput Phys*. 1981;39(1):201-225. doi:10.1016/0021-9991(81)90145-5
4. Johnson AA, Tezduyar TE. Mesh update strategies in parallel finite element computations of flow problems with moving boundaries and interfaces. *Comput Methods Appl Mech Eng*. 1994;119(1-2):73-94. doi:10.1016/0045-7825(94)00077-8
5. De Boer A, Van Zuijlen A, Bijl H. Radial basis functions for interface interpolation and mesh deformation. *Advanced Computational Methods in Science and Engineering*. Springer; 2009:143-178.
6. Behr M, Tezduyar T. The shear-slip mesh update method. *Comput Methods Appl Mech Eng*. 1999;174(3):261-274. doi:10.1016/S0045-7825(98)00299-0
7. Key F, Pauli L, Elgeti S. The virtual ring shear-slip mesh update method. *Comput Fluids*. 2018;172:352-361. doi:10.1016/j.compfluid.2018.04.006
8. Barral N, Alauzet F. Three-dimensional CFD simulations with large displacement of the geometries using a connectivity-change moving mesh approach. *Eng Comput*. 2019;35(2):397-422. doi:10.1007/s00366-018-0607-5
9. Bellini A. *Fused Deposition of Ceramics: A Comprehensive Experimental, Analytical and Computational Study of Material Behavior, Fabrication Process and Equipment Design*. PhD thesis. Drexel University; 2002.
10. Fitzharris ER, Watanabe N, Rosen DW, Shofner ML. Effects of material properties on warpage in fused deposition modeling parts. *Int J Adv Manuf Technol*. 2018;95(5):2059-2070. doi:10.1007/s00170-017-1340-8
11. Xia H, Lu J, Dabiri S, Tryggvason G. Fully resolved numerical simulations of fused deposition modeling. Part I: fluid flow. *Rapid Prototyp J*. 2018;24(2):463-476. doi:10.1108/RPJ-12-2016-0217
12. Comminal R, Serdeczny MP, Pedersen DB, Spangenberg J. Numerical modeling of the strand deposition flow in extrusion-based additive manufacturing. *Addit Manuf*. 2018;20:68-76. doi:10.1016/j.addma.2017.12.013
13. Serdeczny MP, Comminal R, Pedersen DB, Spangenberg J. Experimental validation of a numerical model for the strand shape in material extrusion additive manufacturing. *Addit Manuf*. 2018;24:145-153. doi:10.1016/j.addma.2018.09.022
14. Serdeczny MP, Comminal R, Pedersen DB, Spangenberg J. Numerical prediction of the porosity of parts fabricated with fused deposition modeling. Proceedings of the 29th Annual International Solid Freeform Fabrication Symposium; 2018:1849-1854.
15. Behdani B, Senter M, Mason L, Leu M, Park J. Numerical study on the temperature-dependent viscosity effect on the strand shape in extrusion-based additive manufacturing. *J Manuf Mater Process*. 2020;4(2):46. doi:10.3390/jmmp4020046
16. Tezduyar TE, Behr M, Mittal S, Liou J. A new strategy for finite element computations involving moving boundaries and interfaces–the deforming-spatial-domain/space-time procedure: II. Computation of free-surface flows, two-liquid flows, and flows with drifting cylinders. *Comput Methods Appl Mech Eng*. 1992;94(3):353-371. doi:10.1016/0045-7825(92)90060-W
17. Hilger D, Hosters N, Key F, Elgeti S, Behr M. A novel approach to fluid-structure interaction simulations involving large translation and contact. *Isogeometric Analysis and Applications*. Springer; 2018, 2021:39-56.
18. Donea J, Huerta A. *Finite Element Methods for Flow Problems*. John Wiley & Sons; 2003.
19. Helmig J, Behr M, Elgeti S. Boundary-conforming finite element methods for twin-screw extruders: unsteady-temperature-dependent-non-Newtonian simulations. *Comput Fluids*. 2019;190:322-336. doi:10.1016/j.compfluid.2019.06.028
20. Jansen KE, Collis SS, Whiting C, Shaki F. A better consistency for low-order stabilized finite element methods. *Comput Methods Appl Mech Eng*. 1999;174(1-2):153-170. doi:10.1016/S0045-7825(98)00284-9
21. Pauli L, Behr M. On stabilized space-time FEM for anisotropic meshes: incompressible Navier–Stokes equations and applications to blood flow in medical devices. *Int J Numer Methods Fluids*. 2017;85(3):189-209. doi:10.1002/fld.4378
22. Saad Y, Schultz MH. GMRES: a generalized minimal residual algorithm for solving nonsymmetric linear systems. *SIAM J Sci Stat Comput*. 1986;7(3):856-869. doi:10.1137/0907058
23. Behr M, Tezduyar T. Finite element solution strategies for large-scale flow simulations. *Comput Methods Appl Mech Eng*. 1994;112(1):3-24. doi:10.1016/0045-7825(94)90016-7
24. Zwicke F, Eusterholz S, Elgeti S. Boundary-conforming free-surface flow computations: interface tracking for linear, higher-order and isogeometric finite elements. *Comput Methods Appl Mech Eng*. 2017;326:175-192. doi:10.1016/j.cma.2017.08.022
25. Elgeti S, Sauerland H. Deforming fluid domains within the finite element method: five mesh-based tracking methods in comparison. *Arch Comput Methods Eng*. 2016;23(2):323-361. doi:10.1007/s11831-015-9143-2
26. Helmig J, Key F, Behr M, Elgeti S. Combining boundary-conforming finite element meshes on moving domains using a sliding mesh approach. *Int J Numer Methods Fluids*. 2021;93(4):1053-1073. doi:10.1016/j.cma.2019.112740
27. Sun Q, Rizvi GM, Bellehumeur CT, Gu P. Effect of processing conditions on the bonding quality of FDM polymer filaments. *Rapid Prototyp J*. 2008;14(2):72-80. doi:10.1108/13552540810862028

28. Labs MP. Moldflow material testing report, MAT2238. NatureWorks PLA 7000D; 2007.
29. Xia H, Lu J, Tryggvason G. Fully resolved numerical simulations of fused deposition modeling. Part II–solidification, residual stresses and modeling of the nozzle. *Rapid Prototyp J*. 2018;24(6):973-987. doi:10.1108/RPJ-11-2017-0233

How to cite this article: González FA, Elgeti S, Behr M, Auricchio F. A deforming-mesh finite-element approach applied to the large-translation and free-surface scenario of fused deposition modeling. *Int J Numer Meth Fluids*. 2023;95(2):334-351. doi: 10.1002/fld.5151