

Floating Isogeometric Analysis

Hille, Helge C.; Kumar, Siddhant; De Lorenzis, Laura

DOI

[10.1016/j.cma.2022.114684](https://doi.org/10.1016/j.cma.2022.114684)

Publication date

2022

Document Version

Final published version

Published in

Computer Methods in Applied Mechanics and Engineering

Citation (APA)

Hille, H. C., Kumar, S., & De Lorenzis, L. (2022). Floating Isogeometric Analysis. *Computer Methods in Applied Mechanics and Engineering*, 392, Article 114684. <https://doi.org/10.1016/j.cma.2022.114684>

Important note

To cite this publication, please use the final published version (if applicable).
Please check the document version above.

Copyright

Other than for strictly personal use, it is not permitted to download, forward or distribute the text or part of it, without the consent of the author(s) and/or copyright holder(s), unless the work is under an open content license such as Creative Commons.

Takedown policy

Please contact us and provide details if you believe this document breaches copyrights.
We will remove access to the work immediately and investigate your claim.



Floating Isogeometric Analysis

Helge C. Hille^a, Siddhant Kumar^b, Laura De Lorenzis^{a,*}

^a Department of Mechanical and Process Engineering, ETH Zürich, 8092 Zürich, Switzerland

^b Department of Materials Science and Engineering, Delft University of Technology, 2628 CD Delft, The Netherlands

Received 13 October 2021; received in revised form 24 January 2022; accepted 24 January 2022

Available online xxx

Abstract

We propose Floating Isogeometric Analysis (FLIGA), which extends IGA to extreme deformation analysis. The method is based on a novel tensor-product construction of B-Splines for the update of the basis functions in one direction of the parametric space. With basis functions “floating” deformation-dependently in this direction, mesh distortion is overcome for problems in which extreme deformations occur predominantly along the associated (possibly curved) physical axis. In doing so, we preserve the numerical advantages of splines over many meshless basis functions, while avoiding remeshing. We employ material point integration for numerical quadrature, thus attributing a Lagrangian character to our technique. The paper introduces the method and reviews the fundamental properties of the FLIGA basis functions, including a numerical patch test. The performance of FLIGA is then numerically investigated on the benchmark of Newtonian and viscoelastic Taylor–Couette flow. Finally, we simulate a viscoelastic extrusion-based additive manufacturing process, which served as the original motivation for the new approach.

© 2022 The Author(s). Published by Elsevier B.V. This is an open access article under the CC BY-NC-ND license (<http://creativecommons.org/licenses/by-nc-nd/4.0/>).

Keywords: Isogeometric analysis; Meshless methods; Extreme deformations; Mesh distortion; Extrusion; Additive manufacturing

1. Introduction

Isogeometric analysis (IGA) is a generalization of finite element analysis (FEA) which adopts basis functions from the geometry representation in Computer Aided Design (CAD) instead of the standard C^0 -continuous piecewise polynomials. The first isogeometric techniques were proposed by Hughes et al. [1] with the aim to naturally bridge the gap between CAD and analysis. Since then and beyond the original goal, isogeometric basis functions proved to lead to many advantages for the analysis itself, most notably in relation to their higher and tailorable continuity.

One of the favorable features of IGA is a reduced sensitivity to mesh distortion compared to FEA [2]. Nevertheless, beyond a certain deformation limit, standard (Lagrangian) IGA techniques also suffer from mesh distortion. This issue is common to all mesh-based methods and manifests itself with decreasing result accuracy and final loss of solvability under increasing material deformation.

A common technique to overcome mesh distortion in FEA is remeshing. In such approaches, before critical entanglement occurs, the original mesh is replaced by a new one and the needed information is mapped from the old to the new mesh. Remeshing strategies are computationally expensive and known to accumulate errors whose

* Corresponding author.

E-mail address: ldelorenzis@ethz.ch (L. De Lorenzis).

prediction and control is yet another challenge [3]. What additionally hinders remeshing from extensive application in IGA is its methodical complexity [4]. Nonetheless, research on isogeometric remeshing techniques is currently ongoing, see e.g. [5].

Another alternative to handle extreme deformations in a Lagrangian context are meshless methods, which avoid mesh distortion by giving up the mesh altogether [6,7]. In these methods, the basis functions are computed recurrently from scattered nodal points such that distortion is avoided, and thus follow material deformations in a loosened sense. Note that the Lagrangian character of the evaluation points as well as of the governing equations is preserved, so that many meshless methods may still be considered Lagrangian techniques. Unfortunately, the isogeometric goal of a unification of CAD and analysis is not compatible with the meshless approach. Moreover, meshless methods are frequently less efficient than their mesh-based competitors [8,9] and suffer from well-known instabilities [10,11]. Moreover, for the case of incompressible material behavior which is relevant in this paper, the construction of mixed discretizations to avoid locking is not trivial [12]. The efficiency and stability of mesh-based computations and the flexibility of meshless methods have motivated some hybrid approaches, aiming to benefit from the complementing advantages of both settings, see e.g. [13–17].

A third prominent way to handle extreme deformations is the change from the Lagrangian to the Eulerian viewpoint, which has been widely adopted also within the isogeometric paradigm for fluid mechanics applications [18–22]. However, these approaches typical of computational fluid dynamics (CFD) are not optimal for problems involving solid material behavior such as viscoelasticity or plasticity, which are the focus in this paper. Moreover, Eulerian formulations typically come along with increased numerical complexity since a stabilization of the advective terms as well as a special treatment of moving boundaries are required, see e.g. [18].

In this paper, we propose a hybrid method that adopts meshless concepts to generalize mesh-based IGA. We denote our method as *floating IGA* (FLIGA) and specifically design it for mechanics problems where it is possible to identify a single (possibly curved) direction of predominant deformation, which we call *characteristic direction*. Moderate deformations may still occur perpendicular to this direction. The proposed method is based on the idea of introducing meshless behavior in IGA only as far as the isogeometric concept is preserved, but to such an extent that mesh distortion is overcome. We realize this by breaking the Lagrangian character of isogeometric basis functions and preventing their distortion along the characteristic direction. Yet, the Lagrangian character of governing equations and quadrature points is preserved.

The behavior of FLIGA basis functions is qualitatively illustrated in Fig. 1, which refers to the Taylor–Couette flow with the internal boundary fixed and the external one rotating counterclockwise. Here the characteristic direction of deformation is the circumferential one. Through the floating procedure, the two illustrated basis functions are prevented from undergoing extreme shear deformations. Fig. 2 shows schematically how this behavior is achieved. We start from a classical IGA discretization where one of the straight axes of the rectangular parametric domain (the horizontal one in the figure) is mapped to the characteristic (circumferential) direction in the physical space. We modify the classical tensor-product structure of multivariate B-Spline bases in the parametric domain, in that we grant each basis function the ability to float along the characteristic direction independently from its neighbors in the normal direction. This floating is carried out depending on the deformation and allows to significantly reduce the basis function distortion after finally mapping the parametric domain to physical space. Note that this comes along with a loosening of the classical mesh notion, as the element boundaries perpendicular to the characteristic direction are no longer in place, whereas those along the characteristic direction are kept.

Our development of FLIGA is motivated by the simulation of extrusion-based additive manufacturing (AM) technologies, including e.g. Fused Deposition Modeling (FDM) and bioprinting, which most often involve polymeric materials. In such processes extreme deformations occur along the extrusion direction (Fig. 3). The investigations conducted so far made use of remeshing, meshless methods, or CFD techniques. E.g., Reinold et al. [23] investigated the formation of layer geometries and the layer interaction within large-scale extrusion by the Particle Finite Element Method, a remeshing technique based on FEA. Smoothed Particle Hydrodynamics was adopted in Ouyang et al. [24] to study FDM of a fiber-reinforced polymer, with special attention to the alignment of fibers in the viscous melt. Comminal et al. [25] employed CFD simulations to predict viscous strand deposition flow for small-scale AM.

We believe that FLIGA bears great potential for simulation of extrusion-based AM. Due to the Lagrangian character of FLIGA, we easily treat the viscoelastic behavior of polymers without the need to stabilize advective terms. The meshing efforts are minimized, as the geometry of the nozzle may be obtained seamlessly from its

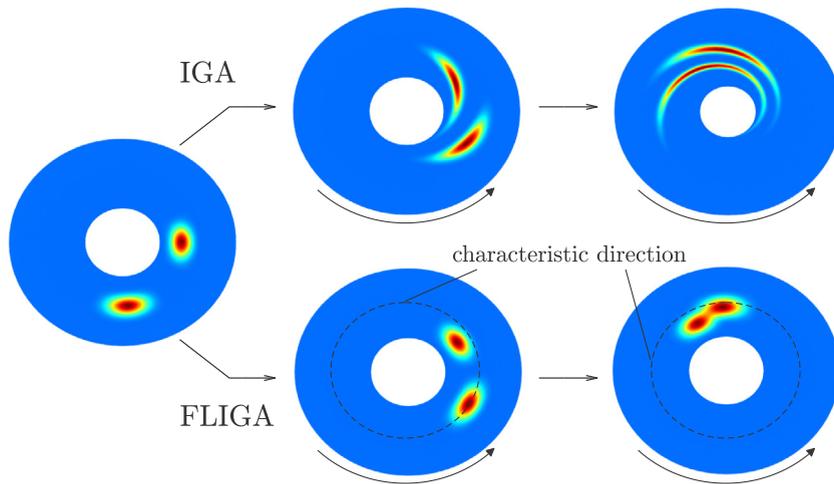


Fig. 1. Comparison of basis function evolution for IGA and FLIGA in the Taylor–Couette flow problem. In IGA, basis functions strictly follow the deformation of the material leading to severe distortion. In FLIGA, we can determine a characteristic direction, where basis functions follow the deformation only in an average sense.

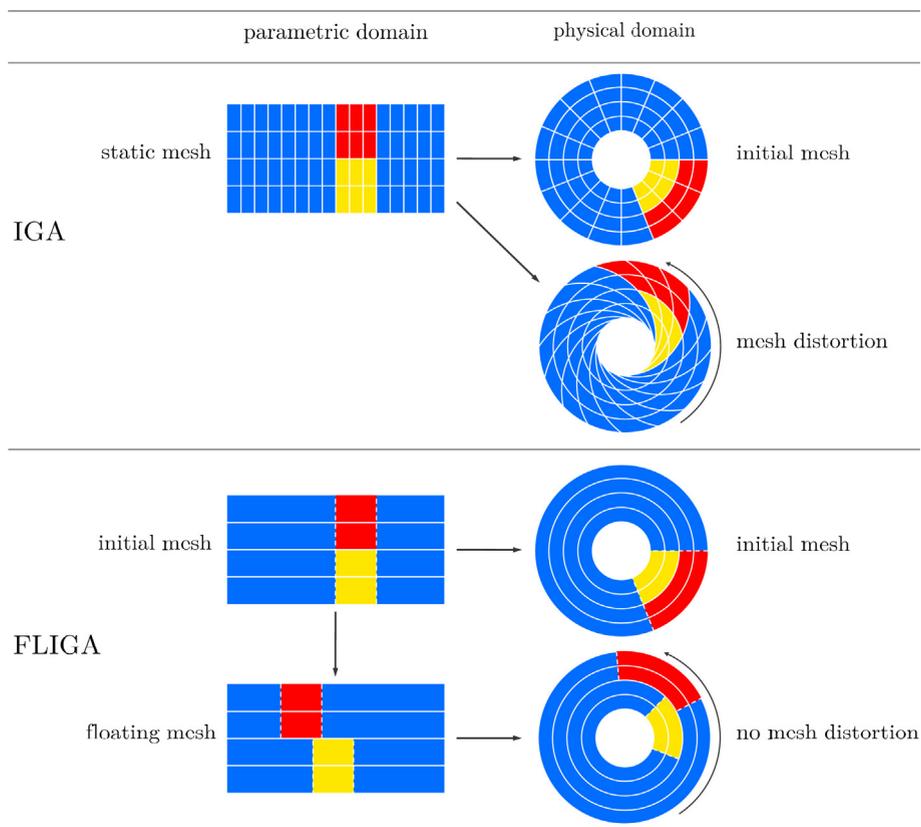


Fig. 2. Evolution of the support of two different basis functions (colored in yellow and red) in the parametric and physical domains for IGA and FLIGA in the Taylor–Couette flow problem.

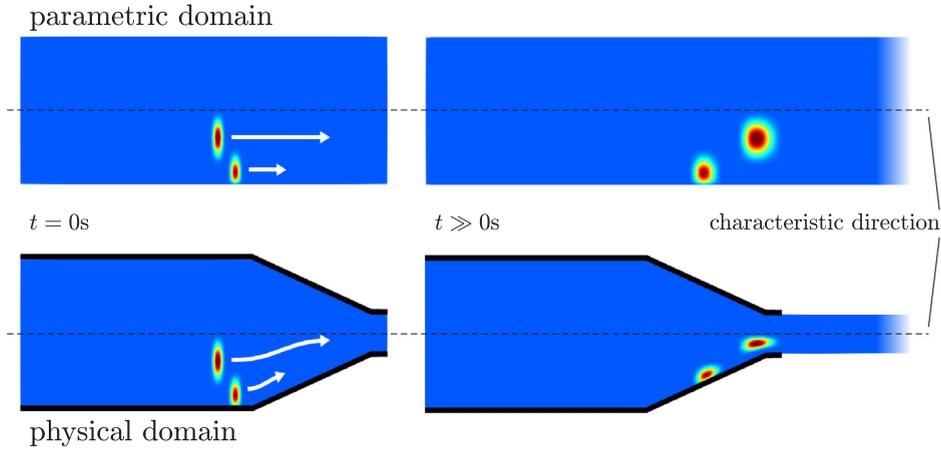


Fig. 3. Schematic basis function evolution for FLIGA in the example of frictional extrusion.

CAD representation. The contact between nozzle and polymer melt is treated robustly due to the smoothness of the basis functions, see [26]. Most importantly, despite the extreme deformations, FLIGA does not suffer from mesh distortion avoiding the drawbacks of remeshing. Free surfaces that form behind the nozzle exit are naturally handled without additional efforts to track boundaries. Incompressible material behavior can be treated by well-established mixed approximation techniques from mesh-based IGA and no stabilization is required. Finally, by staying within the solid mechanics setting we are ideally prepared to study, e.g. the evolution of the polymer melt behavior and properties upon cooling, residual stresses and final part properties.

The outline of the paper is as follows: In Section 2 we review our continuum model and its discretization with IGA, which are the basis for the introduction of FLIGA in Section 3. Numerical examples are provided in Section 4 and conclusions drawn in Section 5.

2. Continuum model and isogeometric analysis

In this section, we briefly recall the fundamental equations of solid mechanics for incompressible materials and review their isogeometric discretization. The formulation is based on velocities rather than on displacements, as it is customary in the literature on viscous flow. For simplicity, in this paper we limit ourselves to the two-dimensional case, however the extension to three dimensions is conceptually straightforward.

2.1. Continuum equations

The basis for our discretization is the Principle of Virtual Power

$$\int_{\Omega} \boldsymbol{\sigma} : \delta \mathbf{D} \, d\Omega = \int_{\partial\Omega_N} \mathbf{h}_N \cdot \delta \mathbf{v} \, d\Gamma, \tag{1}$$

where for simplicity we neglect inertia and body force contributions. Here, $\boldsymbol{\sigma}$ denotes the Cauchy stress, \mathbf{D} is the strain rate tensor related to the velocity \mathbf{v} by

$$\mathbf{D} = \frac{1}{2} (\nabla_{\mathbf{x}} \mathbf{v} + \nabla_{\mathbf{x}}^T \mathbf{v}), \tag{2}$$

and $\delta \mathbf{D}$ and $\delta \mathbf{v}$ are respectively the virtual strain rate and the virtual velocity, also related by

$$\delta \mathbf{D} = \frac{1}{2} (\nabla_{\mathbf{x}} \delta \mathbf{v} + \nabla_{\mathbf{x}}^T \delta \mathbf{v}). \tag{3}$$

We are using the notation $(\nabla_{\mathbf{x}}(\bullet))_{ij} = \partial(\bullet)_i / \partial x_j$ to denote the spatial gradient of a first-order tensor (\bullet) , and $\mathbf{x} \in \Omega$ is the coordinate of a material point in the current (deformed) configuration. The boundary of the spatial domain

$\Omega, \partial\Omega$, consists of two subsets $\partial\Omega_D$ and $\partial\Omega_N$, with $\partial\Omega_D \cup \partial\Omega_N = \partial\Omega$ and $\partial\Omega_D \cap \partial\Omega_N = \emptyset$. On the Dirichlet boundary the velocity is prescribed

$$\mathbf{v} = \mathbf{v}_D \quad \forall \mathbf{x} \in \partial\Omega_D, \tag{4}$$

whereas the external surface force vector \mathbf{h}_N acts on the Neumann boundary

$$\boldsymbol{\sigma} \cdot \mathbf{n} = \mathbf{h}_N \quad \forall \mathbf{x} \in \partial\Omega_N, \tag{5}$$

where \mathbf{n} is the outward normal unit vector at each point of the boundary.

Without loss of generality, we decompose the Cauchy stress $\boldsymbol{\sigma}$ in a volumetric and a deviatoric contribution:

$$\boldsymbol{\sigma} = -p\mathbf{I} + \boldsymbol{\tau}, \tag{6}$$

where p is the pressure, \mathbf{I} the second-order identity tensor and $\boldsymbol{\tau}$ the deviatoric part of the Cauchy stress. We give more details on the constitutive relationships later, when applying the Principle of Virtual Power to specific materials with viscous and viscoelastic behavior.

In addition to the Principle of Virtual Power, we require incompressibility of the velocity field, i.e.

$$\nabla_{\mathbf{x}} \cdot \mathbf{v} = 0 \quad \forall \mathbf{x} \in \Omega, \tag{7}$$

which we recast in weak form as

$$\int_{\Omega} (\nabla_{\mathbf{x}} \cdot \mathbf{v}) \delta p \, d\Omega = 0, \tag{8}$$

where δp is a virtual scalar pressure field.

In problems where we do not impose any Neumann boundary conditions, i.e. $\partial\Omega_D = \partial\Omega$, the pressure is not uniquely determined, but allows for an arbitrary constant offset. In such cases, we achieve a unique pressure solution by enriching the problem with a Dirichlet pressure prescription p_D at a suitable part of the boundary $\partial\Omega_{D,p} \in \partial\Omega$, i.e.

$$p = p_D \quad \forall \mathbf{x} \in \partial\Omega_{D,p}. \tag{9}$$

Note that all the fields involved in the previous equations undergo spatial and temporal variation, which we have not explicitly written for notational simplicity. We can obtain the transient evolution of a material particle \mathbf{x}_p by means of

$$\mathbf{x}_p(\mathbf{x}_p^0, t) = \mathbf{x}_p^0 + \int_0^t \mathbf{v}_p(\mathbf{x}_p(\mathbf{x}_p^0, \tau), \tau) \, d\tau, \tag{10}$$

where \mathbf{x}_p^0 denotes the initial particle position at time $t = 0$.

2.2. B-Splines

In this section, we summarize the construction of isogeometric basis functions, which are needed for the subsequent isogeometric discretization. Here we limit ourselves to B-Splines, however, in the past years, other function types from the CAD community have been employed in the context of IGA, e.g. NURBS, T-Splines, and a few more [1,27].

Given the integers $r \geq 0$ and $I \geq r + 1$, we define a knot vector as

$$\Xi = \{\xi_1, \xi_2, \dots, \xi_i, \dots, \xi_{I+r+1} \mid \xi_i \in \mathbb{R}\}. \tag{11}$$

Here we assume repetition of the first $r + 1$ and last $r + 1$ knots and monotonic increase for the entire knot vector, i.e. $\xi_1 = \xi_{r+1}$ and $\xi_{I+1} = \xi_{I+r+1}$, while $\xi_i \leq \xi_{i+1}$. Note that we will assume the same for all other knot vectors introduced later, also beyond this section.

Let us now span a parametric domain $\hat{\Omega}_{\xi} = [\xi_1, \xi_{I+r+1}]$ over coordinate ξ . A univariate (denoted for brevity as 1D) B-Spline basis $\{\hat{N}_{i,r}(\xi)\}_{i=1}^I$ on $\hat{\Omega}_{\xi}$ with polynomial order r can be constructed by the Cox–de Boor recursion formula

$$\begin{aligned} r = 0 : \quad \hat{N}_{i,0}(\xi) &= \begin{cases} 1 & \text{for } \xi_i \leq \xi < \xi_{i+1} \\ 0 & \text{otherwise} \end{cases}, \\ r \geq 1 : \quad \hat{N}_{i,r}(\xi) &= \frac{\xi - \xi_i}{\xi_{i+r} - \xi_i} \hat{N}_{i,r-1}(\xi) + \frac{\xi_{i+r+1} - \xi}{\xi_{i+r+1} - \xi_{i+1}} \hat{N}_{i+1,r-1}(\xi), \end{aligned} \tag{12}$$

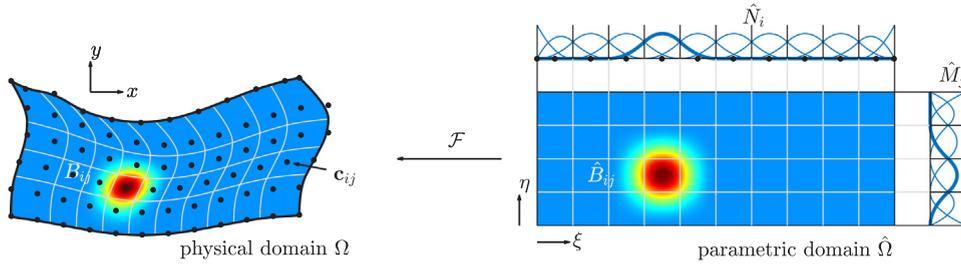


Fig. 4. Construction of classical bivariate tensor-product B-Splines.

where $\xi_i \in \Xi$. It holds $\hat{N}_{i,r-1} = 0$, if $\xi_i = \xi_{i+r}$; as well as $\frac{0}{0} = 0$. For more concise notation, from now on we will omit the polynomial order index r when referring to B-Spline bases or their individual functions.

In classical IGA, a bivariate (also denoted for brevity as 2D) B-Spline basis $\{\hat{B}_{ij}(\xi)\}_{i=1,j=1}^{I,J}$ is constructed by means of a tensor-product structure

$$\hat{B}_{ij}(\xi, \eta) = \hat{N}_i(\xi) \hat{M}_j(\eta), \tag{13}$$

in which $\hat{M}_j \in \{\hat{M}_j(\eta)\}_{j=1}^J$ are again 1D B-Spline basis functions of polynomial order r (assuming equal-order discretization in both parametric directions), however, defined on $\hat{\Omega}_\eta = [\eta_1, \eta_{J+r+1}]$ and associated with knot vector $\mathcal{H} = \{\eta_1, \eta_2, \dots, \eta_j, \dots, \eta_{J+r+1} \mid \eta_j \in \mathbb{R}\}$. Hence, the 2D B-Spline basis consists of $I \cdot J$ basis functions that span the parametric domain $\hat{\Omega} = \hat{\Omega}_\xi \times \hat{\Omega}_\eta$.

For the representation of B-Spline freeform surfaces we employ a linear combination $\mathcal{F}^n : \hat{\Omega} \rightarrow \Omega^n$, mapping the static rectangular parametric domain to the current physical domain, i.e.

$$\mathbf{x} = \mathcal{F}^n(\xi) = \sum_i \sum_j c_{ij}^n \hat{B}_{ij}(\xi). \tag{14}$$

The 2D basis can thus be expressed in terms of physical (current) coordinates as

$$B_{ij}^n(\mathbf{x}) = \hat{B}_{ij}(\mathcal{F}^{n-1}(\mathbf{x})). \tag{15}$$

The linear combination coefficients c_{ij}^n in (14) are the (current) coordinates of the so-called control points and are subject to transient evolution, whereas the B-Spline basis in terms of parametric coordinates does not change over time. We use the Jacobian \mathbf{J}^n to associate parametric and physical derivatives, i.e.

$$\mathbf{J}^n(\xi) = \nabla_\xi \mathbf{x} = \nabla_\xi \mathcal{F}^n = \sum_i \sum_j c_{ij}^n \nabla_\xi^T \hat{B}_{ij}(\xi). \tag{16}$$

For convenience, we summarize the indices i, j in the single running index b , which is uniquely associated to them through

$$b = (j - 1) \cdot I + i. \tag{17}$$

We summarize the above concepts graphically in Fig. 4. The extension to the construction of trivariate B-Spline bases and freeform volumes can be found in Cottrell et al. [28].

2.3. Discrete equations

With the B-Spline basis functions at hand, we can now replace the continuum model of Section 2.1 by a discrete formulation, limiting ourselves to the search for approximate solutions. In the literature, discretized quantities are sometimes denoted with a superscript h . However, we drop such an index as computational quantities are from now on always related to the discrete approximate model, unless specified otherwise. We introduce the following

isogeometric field approximations for velocity \mathbf{v}^n , pressure p^n , virtual velocity $\delta \mathbf{v}^n$ and virtual pressure δp^n :

$$\begin{aligned} \mathbf{v}^n(\mathbf{x}) &= \sum_{b=1}^B \mathbf{d}_b^n B_b^n(\mathbf{x}), & \delta \mathbf{v}^n(\mathbf{x}) &= \sum_{b=1}^B \delta \mathbf{d}_b^n B_b^n(\mathbf{x}), \\ p^n(\mathbf{x}) &= \sum_{a=1}^A q_a^n A_a^n(\mathbf{x}), & \delta p^n(\mathbf{x}) &= \sum_{a=1}^A \delta q_a^n A_a^n(\mathbf{x}). \end{aligned} \tag{18}$$

Thus, our bases comprise B B-Spline basis functions $B_b^n(\mathbf{x})$ for the velocity approximation and A B-Spline basis functions $A_a^n(\mathbf{x})$ for the pressure approximation, and \mathbf{d}_b^n and q_a^n are the control point velocities and pressures, respectively. The virtual quantities are interpreted accordingly. We also discretize in time, which is indicated by the superscript n referring to the discrete time step t^n .

With these approximations, we can write the discrete form of the Principle of Virtual Power as

$$\int_{\Omega^n} (-p^n \cdot \mathbf{I} + \boldsymbol{\tau}^n) : \delta \mathbf{D}^n \, d\Omega^n = 0. \tag{19}$$

Here, we incorporated the volumetric–deviatoric split of the Cauchy stress and assumed $\mathbf{h}_N = \mathbf{0}$. Further, $\boldsymbol{\tau}^n$ denotes the discretized deviatoric Cauchy stress. We obtain the discretized version of the kinematics by replacing the continuous velocity fields in (3) with the respective ansatzes from (18).

The incompressibility equation in discrete form reads

$$\int_{\Omega^n} (\nabla_{\mathbf{x}} \cdot \mathbf{v}^n) \delta p^n \, d\Omega^n = 0. \tag{20}$$

If we rewrite (19) by inserting the ansatz for the discretized virtual strain rate and exploiting the assumption of symmetry for the deviatoric stress tensor, for each control point index b we obtain a global residual vector \mathbf{S}_b^n that has to vanish, i.e.

$$\mathbf{S}_b^n = \int_{\Omega^n} (-p^n \cdot \mathbf{I} + \boldsymbol{\tau}^n) \nabla_{\mathbf{x}} B_b^n \, d\Omega^n = \mathbf{0} \quad \forall b = 1, \dots, B. \tag{21}$$

From the weak form of the incompressibility condition we obtain another global residual vector, which again must vanish for each control point index a , i.e.

$$\mathcal{Q}_a^n = \int_{\Omega^n} (\nabla_{\mathbf{x}} \cdot \mathbf{v}^n) A_a^n \, d\Omega^n = 0 \quad \forall a = 1, \dots, A. \tag{22}$$

The expressions (21) and (22) represent a global, potentially non-linear system of equations. In order for the system to be solvable we need to incorporate the discretized Dirichlet boundary conditions, which we briefly discuss in Section 2.4. The non-linear system is commonly solved with the Newton–Raphson strategy, see Appendix A for more details, and the solution delivers the current control point velocity and pressure unknowns \mathbf{d}_b^n and q_a^n . Along with the basis functions, these yield the approximate velocity and pressure fields according to (18).

2.4. Remarks

- Assuming incompressible material behavior, we have introduced a mixed discretization scheme. It is well known that the discretization spaces for the velocity and pressure fields must be chosen appropriately to ensure stability. In this paper, we adopt the subdivision approach proposed in Rüberg and Cirak [29], where the B-Spline bases for the two fields are constructed in an efficient manner from equal polynomial order but different element subdivision, namely, in 2D one pressure element in the parametric space is split into 2×2 velocity elements.
- Integrals of a generic function f over the physical domain at time step t^n can be numerically approximated as

$$\int_{\Omega^n} f(\mathbf{x}) \, d\Omega^n \approx \sum_{q=1}^Q W_q^n \cdot f(\mathbf{x}_q^n). \tag{23}$$

In IGA, the Q quadrature points \mathbf{x}_q^n and quadrature weights W_q^n are often chosen as Gauss quadrature, although much more efficient alternatives are available [30,31].

- In IGA, incorporation of (possibly inhomogeneous) Dirichlet boundary conditions in the discrete global equation system is a straightforward task due to the weak Kronecker-delta property of the basis functions, and is performed by imposing suitable control point values on boundary control points. More details on boundary conditions in IGA are given in Cottrell et al. [28].
- For the evolution of the physical domain in time, the control point velocities are integrated to displacements and added to previous control point positions, e.g. using the forward Euler scheme

$$\mathbf{c}_{ij}^{n+1} = \mathbf{c}_{ij}^n + \Delta \mathbf{c}_{ij}^n = \mathbf{c}_{ij}^n + \mathbf{d}_{ij}^n \cdot \Delta t. \tag{24}$$

Computing the basis functions at the next time step from the same bivariate B-Spline basis (which stays identical in the parametric space) but using updated control point positions attributes a material point character to all parametric points, i.e. a point in the parametric space is always associated to the same physical particle of moving material, which is favorable for the formulation of material time derivatives as well as for accurate integration. On the other hand, in presence of very large deformations, a severe distortion of the mapping between parametric and physical space occurs, and IGA basis functions in the physical domain may no longer be suitable for the analysis. This is qualitatively demonstrated in the upper branch of Fig. 1. In such cases, techniques to control distortion of the basis functions become necessary.

3. Basics of FLIGA

In this section, we introduce FLIGA as a new technique for controlling distortion of isogeometric B-Spline basis functions. FLIGA allows for the extension of isogeometric concepts to extreme deformation analysis. The previously reviewed fundamentals of classical Lagrangian IGA are all maintained, except for those concerning B-Spline basis function design and numerical Gauss quadrature. These are both modified in this section.

3.1. Introduction of floating B-Splines

We start the introduction of FLIGA by proposing an alternative for the construction of 2D B-Spline bases, as visualized in Fig. 5. The core idea is a generalization of the classical tensor-product structure from IGA. We denote this generalization as *floating tensor-product structure*. On the parametric domain, we associate each 1D B-Spline basis function $\hat{M}_j \in \left\{ \hat{M}_j(\eta) \right\}_{j=1}^J$ with its own 1D B-Spline basis, such that instead of (13) we write

$$\hat{B}_{ij}^n(\xi, \eta) = \hat{N}_{ij}^n(\xi) \hat{M}_j(\eta). \tag{25}$$

In this floating tensor-product structure, ξ is the *characteristic* direction and η is the *normal* direction. In the physical space, the characteristic direction is mapped to the (possibly curved) direction along which deformations are expected to be severe, whereas the deformations along the normal direction are assumed to be moderate. We denote as $\left\{ \hat{N}_{ij}^n(\xi) \right\}_{i=j}^I$ the characteristic B-Spline bases, of which we count a total number of J . Similarly, the single basis $\left\{ \hat{M}_j(\eta) \right\}_{j=1}^J$ is termed normal B-Spline basis. Note that the characteristic B-Spline bases in parametric coordinates are no longer the same throughout the analysis as in IGA, but evolve with proceeding time step n . This attribute extends to the 2D B-Spline basis $\left\{ \hat{B}_{ij}^n(\xi) \right\}_{i=1, j=1}^{I, J}$.

We derive the characteristic B-Spline bases from a static (i.e. unchanged throughout the analysis) 1D *parent* B-Spline basis $\left\{ \tilde{N}_i(\tilde{\xi}) \right\}_{i=1}^I$ of order r on $\tilde{\Omega} = \left[\tilde{\xi}_1, \tilde{\xi}_{I+r+1} \right]$ built by the recursive formula (12) with parent knot vector

$$\tilde{\Xi} = \left\{ \tilde{\xi}_1, \tilde{\xi}_2, \dots, \tilde{\xi}_i, \dots, \tilde{\xi}_{I+r+1} \mid \tilde{\xi}_i \in \mathbb{R} \right\}. \tag{26}$$

The term *parent* refers to the descendance of all characteristic bases from this basis. With the parent basis at hand, we introduce a set of J linear mappings $\mathcal{G}_j^n : \tilde{\Omega} \rightarrow \hat{\Omega}_\xi^n$, with $\hat{\Omega}_\xi^n = [a^n, b^n]$, such that

$$\xi = \mathcal{G}_j^n(\tilde{\xi}) = \sum_i h_{ij}^n \tilde{N}_i(\tilde{\xi}). \tag{27}$$

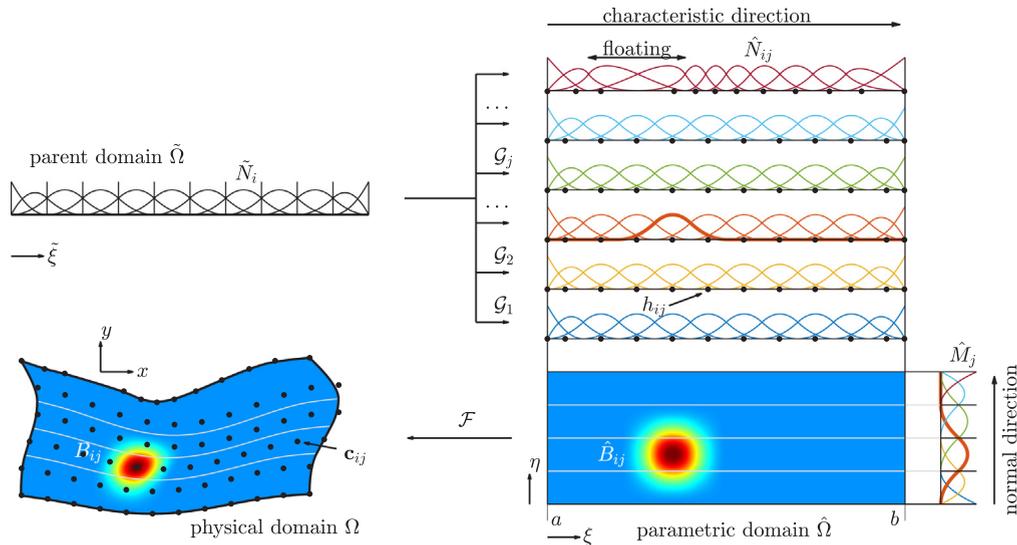


Fig. 5. Schematic overview of bivariate B-Spline basis construction in FLIGA.

We introduce the term *floating maps* for these \mathcal{G}_j^n in order to distinguish them from the later introduced geometric mapping to physical space. It is these floating maps that grant a floating character to the characteristic basis functions by translating the parent basis into the characteristic basis functions

$$\hat{N}_{ij}^n(\xi) = \tilde{N}_i(\mathcal{G}_j^{n-1}(\xi)). \quad (28)$$

As the arrangement of the linear combination coefficients h_{ij}^n allows to control this floating, we refer to these as *floating regulation points*. Due to weak Kronecker-delta property of the parent basis at its boundaries, the span of $\hat{\Omega}_\xi^n$ requires $\forall j : h_{1j}^n = a^n$ and $h_{lj}^n = b^n$. The scalar Jacobians of the floating maps are important quantities and read

$$J_j^n(\tilde{\xi}) = \sum_i h_{ij}^n \nabla_{\tilde{\xi}} \tilde{N}_i(\tilde{\xi}). \quad (29)$$

Finally, the characteristic bases are combined by means of (25) with the normal basis, see also Fig. 5.

The procedure described above yields the 2D B-Spline basis supported on $\hat{\Omega}^n = \hat{\Omega}_\xi^n \times \hat{\Omega}_\eta$. Then, as in classical IGA, we use a geometric mapping $\mathcal{F}^n : \hat{\Omega}^n \rightarrow \Omega^n$ according to (14) in order to obtain bivariate B-Spline bases defined in the physical space, see (15).

The above procedure leads to the capability of basis functions corresponding to different normal parametric coordinates to evolve independently from each other along the characteristic direction. This feature underlies the terminology of *floating* isogeometric analysis and proves crucial for the treatment of extreme deformations associated to the characteristic direction. Through the introduction of the floating regulation points, the degrees of freedom for basis function construction are significantly increased, however, as will be shown shortly, at a limited computational cost. Let us assume having a parent and normal knot vector at hand. From the enriched basis function space that we obtain by introducing the additional mapping (27), we choose for the analysis the basis functions that are ideally suited in terms of distortion, while efficient to obtain. This leads to the updating schemes for the floating regulation points and the control points, which are given in Section 3.2.

Unlike in standard IGA, in FLIGA we do not let the basis functions in physical space follow the deformation fully, but only as far as a sufficiently undistorted shape is preserved. Unlike in remeshing techniques, the evolution of the basis functions in FLIGA is incremental throughout the simulation. Conceptually, we lie in between and the recognition of the material deformation from basis function deformation is possible by averaging the movement of all basis functions.

The FLIGA strategy we have described is reminiscent of core strategies in meshless methods, which also incrementally evolve basis functions while relaxing constraints from a rigid and Lagrangian element division. In

FLIGA we loosen mesh division in the characteristic direction, however we preserve a generalization of the classical mesh, where elemental subdivision persists at the level of single basis functions. Thus, FLIGA can be viewed as a generalization of IGA to the floating tensor-product structure that we obtain by incorporating some aspects of meshless behavior.

3.2. Floating regulation point updates

The proper choice of update rules for the position of control points and floating regulation points is essential for the success of FLIGA, as this combination offers control over the distortion of the basis functions in the physical space along the characteristic direction. By keeping for the control points, \mathbf{c}_{ij}^n , the deformation-dependent update (24) from classical IGA, the task reduces to finding an updating scheme for the floating regulation points, h_{ij}^n , such that distortion of basis functions is minimized. The update can be expressed as

$$h_{ij}^{n+1} = h_{ij}^n + \Delta h_{ij}^n, \tag{30}$$

and we need to determine Δh_{ij}^n . Let us introduce a time-invariant level function \mathcal{L} which associates a desired parametric coordinate in the characteristic direction ξ to each generalized physical coordinate $\chi \in S^n(\mathbf{c}_{ij}^n)$, where S denotes the polygonal area spanned by the control net $\{\mathbf{c}_{ij}^n\}_{i=1, j=1}^{I, J}$, i.e.

$$\xi = \mathcal{L}(\chi). \tag{31}$$

Note that \mathcal{L} cannot be chosen as the ξ -component of the inverse mapping \mathcal{F}^{n-1} , since it has to be defined also for $\chi = \mathbf{c}_{ij}^n \notin \Omega^n$. In the choice of \mathcal{L} we have some freedom, provided it is sufficiently regular in space and it has no local minima or maxima on all S^n .

Due to the association of floating regulation points and control points, an ideal update is

$$d\xi = \nabla_{\chi} \mathcal{L}(\chi) \cdot d\chi \quad \rightarrow \quad \Delta \bar{h}_{ij}^n = \nabla_{\chi} \mathcal{L}(\mathbf{c}_{ij}^n) \cdot \Delta \mathbf{c}_{ij}^n = \nabla_{\chi} \mathcal{L}(\mathbf{c}_{ij}^n) \cdot \mathbf{d}_{ij}^n \cdot \Delta t. \tag{32}$$

However, this in general would not preserve the rectangular structure of the parametric domain, as for $j = 1, \dots, J$ different updates would be introduced on h_{1j}^n (and also on h_{Ij}^n). To correct this, we introduce a blending function $\gamma^n(\xi)$ taking the value $\gamma^n = 0$ at the boundaries $\xi = a^n$ and $\xi = b^n$, where we do not apply the update (32), but one that preserves the rectangular geometry of $\hat{\Omega}$. A transition is defined towards the region $g_1^n < \xi < g_2^n$ in which a plateau is reached, $\gamma^n = 1$. We typically choose $a^n < g_1^n \ll b^n$ and $a^n \ll g_2^n < b^n$ to obtain a reasonably large plateau in which updates are then fully addressed by (32). For $0 < \gamma^n < 1$ a transitional treatment is applied.

Let us now apply \mathcal{L} and γ^n in order to obtain the incremental update of a floating regulation point h_{ij}^n corresponding to the physical control point \mathbf{c}_{ij}^n as

$$\Delta h_{ij}^n = \gamma^n(h_{ij}^n) \cdot \Delta \bar{h}_{ij}^n + (1 - \gamma^n(h_{ij}^n)) \cdot \Delta \bar{h}_{(I^*) (J/2)}^n. \tag{33}$$

The uniform update at the boundaries $\xi = a^n$ and $\xi = b^n$ is thus achieved by employing for all $j = 1, \dots, J$ the update (32) for the respective central boundary control point ($j = J/2$, rounded if required). To distinguish the boundaries $\xi = a^n$ and $\xi = b^n$, we choose in (33) $I^* = 1$ for $i < I/2$ and $I^* = I$ for $i > I/2$. Note that the transition of the blending function between boundary and central regions in the characteristic direction ensures a proper distribution of the floating regulation points.

In the initial state, the floating regulation points h_{ij}^0 are chosen at the Greville abscissae [32] of the parent knot vector, such that $\forall j : \xi = \mathcal{G}_j^0(\tilde{\xi}) = \tilde{\xi}$. As a result, the floating mesh of the initial time step $n = 0$ conforms to a B-Spline mesh with classical element division. This ensures the compatibility of FLIGA with the isogeometric concept of using CAD geometry representations further for analysis.

We illustrate the idea of the level function on the example of extrusion in Fig. 6, where the physical domain Ω^n is colored in blue. Obviously, a suitable mapping of the parametric space to Ω^n would associate the dashed lines in the figure to a constant ξ each, with ξ increasing from one dashed line to the next. Accordingly, we design \mathcal{L} to take constant values along these levels and evolve monotonically along the characteristic extrusion direction (yellow). An update of the floating regulation points based on such \mathcal{L} preserves the constancy of ξ along the desired levels throughout simulation. We also depict the resulting behavior for two basis function candidates aligned to both

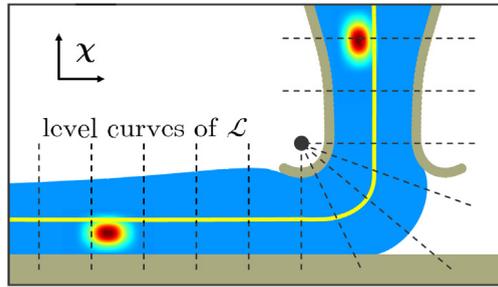


Fig. 6. Levels defined by a suitable level function for extrusion simulation.

the characteristic direction and the levels of \mathcal{L} . Note that the lack of regularity of the level function at the point evidenced with the black dot is unproblematic, as this point is never inside Ω^n . Nonetheless, a certain distance between this or similar points and the physical domain is required.

3.3. Basis function evaluation and connectivities

Let us now consider the location of a parametric point at time step t^n , $\xi_q^n = (\xi_q^n, \eta_q^n)^T$, and aim at computing the basis functions, their physical derivatives and the control point connectivity.

We begin by identifying the normal connectivities $\mathcal{J}_q^n = \{j \in \{1, \dots, J\} : \hat{M}_j(\eta_q^n) \neq 0\}$ associated to ξ_q^n . Due to the classical 1D B-Spline basis character of the normal basis, this task can be performed just like in IGA. As the considered parametric point might in general pass element boundaries in normal direction during the simulation, these connectivities must be obtained for the current time step.

Then, for each $j \in \mathcal{J}_q^n$, the corresponding parent coordinate

$$\tilde{\xi}_{qj}^n = \mathcal{G}_j^{n-1}(\xi_q^n), \tag{34}$$

is computed by means of Newton–Raphson schemes, see [Appendix B](#) for details.

Given the different parent coordinates $\tilde{\xi}_{qj}^n$ and the unique normal coordinate η_q^n , we can compute the parametric basis function values at ξ_q^n as

$$\hat{B}_{ij}^n(\xi_q^n) = \hat{N}_{ij}^n(\xi_q^n) \cdot \hat{M}_j(\eta_q^n) = \tilde{N}_i(\tilde{\xi}_{qj}^n) \cdot \hat{M}_j(\eta_q^n), \tag{35}$$

while the scalar Jacobians (29) yield the parametric derivatives

$$\begin{aligned} \nabla_{\xi} \hat{B}_{ij}^n(\xi_q^n) &= \begin{pmatrix} \nabla_{\xi} \hat{N}_{ij}^n(\xi_q^n) \cdot \hat{M}_j(\eta_q^n) \\ \hat{N}_{ij}^n(\xi_q^n) \cdot \nabla_{\eta} \hat{M}_j(\eta_q^n) \end{pmatrix} \\ &= \begin{pmatrix} J_j^n(\tilde{\xi}_{qj}^n)^{-1} \cdot \nabla_{\tilde{\xi}} \tilde{N}_i(\tilde{\xi}_{qj}^n) \cdot \hat{M}_j(\eta_q^n) \\ \tilde{N}_i(\tilde{\xi}_{qj}^n) \cdot \nabla_{\eta} \hat{M}_j(\eta_q^n) \end{pmatrix}. \end{aligned} \tag{36}$$

The mapping of the evaluated B-Spline basis functions to the physical point $\mathbf{x}_q^n = \mathcal{F}^n(\xi_q^n)$ does not alter the basis function value

$$B_{ij}^n(\mathbf{x}_q^n) = \hat{B}_{ij}^n(\xi_q^n), \tag{37}$$

but controls the physical gradients via the Jacobian (16), i.e.

$$\nabla_{\mathbf{x}} B_{ij}^n(\mathbf{x}_q^n) = \mathbf{J}^n(\xi_q^n)^{-T} \nabla_{\xi} \hat{B}_{ij}^n(\xi_q^n). \tag{38}$$

For assembling the contribution of the evaluation point into the global system of equations, its global control point connectivities are required. We complement the normal connectivities \mathcal{J}_q^n by the J sets of characteristic connectivities $\mathcal{I}_{qj}^n = \{i \in \{1, \dots, I\} : \tilde{N}_i(\tilde{\xi}_{qj}^n) \neq 0\}$, obtained again as in IGA. The global connectivities of the

FLIGA B-Spline basis are then defined as

$$\mathcal{B}_q^n = \left\{ b \in \{1, \dots, I \cdot J\} : \hat{B}_{ij}(\xi_q^n) \neq 0 \right\} = \{ b = (j - 1) \cdot I + i \mid i \in \mathcal{I}_{qj}^n, j \in \mathcal{J}_q^n \}. \quad (39)$$

Profiting from the parametric substructure, this control point search procedure is significantly simpler than nodal search algorithms of classical meshless methods.

3.4. Basis function properties

As follows, we list and discuss the most important properties of FLIGA B-Splines in order to assess their suitability for analysis:

- *Continuity*: Basis function continuity follows the same rules as for classical tensor-product B-Splines. However, in FLIGA, the lower-continuity locations of the univariate B-Splines are not all aligned in the normal parametric direction as in IGA, see [Appendix C](#) for an exemplary illustration.
- *Support*: The number of supported basis functions at an arbitrary evaluation point is constant over the entire domain and is the same as in IGA - e.g. for splines of order r in both parametric directions, this number is $(r + 1)^2$.
- *Partition of unity*: The proof can be found in [Appendix D](#).
- *First-order consistency*: It is demonstrated with a simple patch test in Section 4.2.
- *Weak Kronecker-delta property*: The proof can be found in [Appendix E](#).
- *Boundary preservation*: Physical boundaries are preserved while basis functions are floating; see [Appendix F](#) for a proof.

The above list shows that FLIGA B-Splines preserve many attributes of classical tensor-product B-Splines that are favorable for analysis. For instance, a minimum order of global continuity can be prescribed by the choice of the polynomial B-Spline degree. Continuity can thus be adapted to the problem at hand. The fact that the number of supported basis functions at an evaluation point is constant over the entire domain and known *a priori* is convenient for the implementation, as there is no need to adaptively tune parameters to adjust the overlap of the basis functions during the simulation (as sometimes necessary in meshless methods). In FLIGA, such overlap is straightforward to identify on the parametric domain, hence the search for connectivities can be carried out with lower effort than in meshless methods. Mixed discretization pairs can be easily obtained by adjusting the polynomial orders for velocity and pressure (e.g. as Taylor–Hood pair) or by properly subdividing parent and normal domain (as mentioned earlier, we apply the latter strategy in this work, see Section 3.5). FLIGA B-Splines always exactly cover the analysis domain, allowing for the natural treatment of moving boundaries as in Lagrangian IGA. Boundaries are precisely defined in FLIGA and can be consistently refined with strategies from IGA. The imposition of essential boundary conditions is straightforward due to the weak Kronecker-delta property. First-order consistency is naturally satisfied, whereas it is not present in many meshless techniques. In fact, the lack of (weak) Kronecker-delta property and first-order consistency is believed to be related to most numerical issues encountered in the meshless context [33].

However, FLIGA also has limitations, most importantly, the restriction that mesh distortion can only be overcome in one parametric direction. The fact that the parametric domain must have rectangular geometry is a limitation inherited from IGA that can be partially solved by the extension to multi-patch geometries — however the freedom of meshless methods in dealing with complex topologies and/or topological changes cannot be achieved. Also, refinement is affected by the anisotropic nature of basis function construction and thus is non-local in the normal direction. Finally, exact quadrature of FLIGA B-Splines is also challenging, see Section 3.6.

3.5. Mixed subdivision technique

Since our intention is to apply FLIGA to the study of incompressible materials (the polymer melt in extrusion-based AM), we need mixed-field approximations to avoid volumetric locking. Thus, we employ the previously introduced design principles to obtain a pair of FLIGA B-Spline bases: one for geometry and velocity (as per the isoparametric concept), and the other one for pressure approximation.

For the choice of the velocity and pressure bases, we propose an adaption of the mixed subdivision strategy from classical IGA [29] to FLIGA, see the illustration in [Fig. 7](#). We construct the 2D pressure approximation from

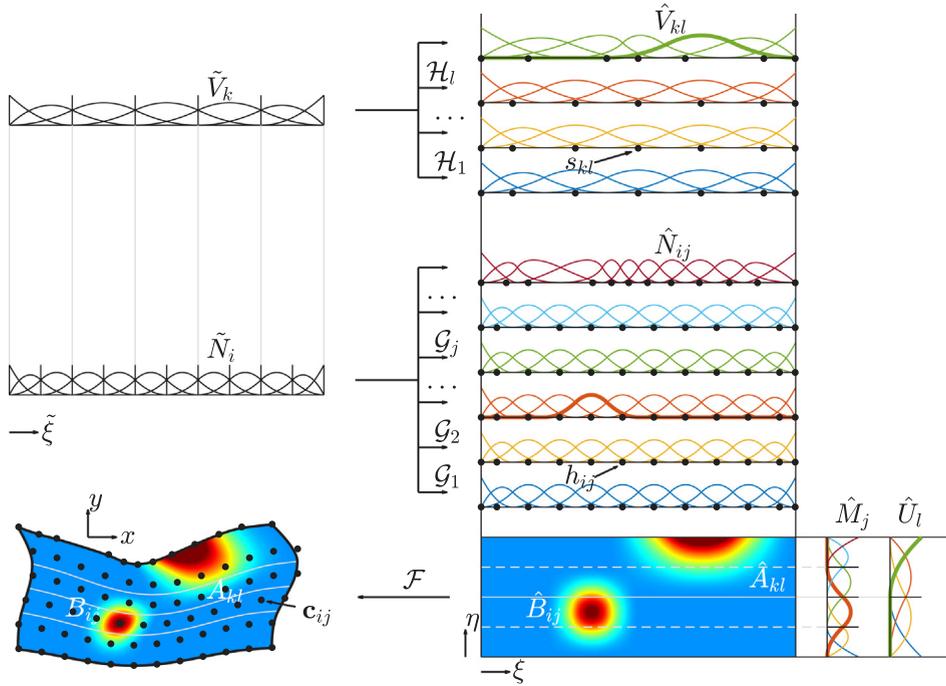


Fig. 7. Schematic overview of the mixed subdivision technique in FLIGA.

coarser parent and normal bases, $\{\tilde{V}_k\}_{k=1}^K$ and $\{\hat{U}_l\}_{l=1}^L$, respectively, than the equivalents for velocity, $\{\tilde{N}_i\}_{i=1}^I$ and $\{\hat{M}_j\}_{j=1}^J$. In our computations, locking is successfully suppressed if one pressure parent knot span comprises two equal-sized parent knot spans for velocity and the same holds for the normal bases. Based on these observations, we propose the use of an equal parent domain $\tilde{\Omega}$ for velocity and pressure.

Then, we introduce one set of floating regulation points h_{ij}^n for design of the velocity basis, as well as another set s_{kl}^n for design of the pressure basis. The corresponding floating maps \mathcal{G}_j^n and \mathcal{H}_l^n are unique to the velocity or pressure basis, respectively.

Unlike the floating maps, the physical mapping is universal and determined by the velocity basis and the physical control points c_{ij}^n . Thus, the velocity and pressure bases must be constructed on the same parametric domain $\hat{\Omega}^n$. Note that, despite the coarser division, the physical boundaries of the pressure basis coincide with those of the velocity basis.

We can now apply the update procedure of Section 3.2 to the control points c_{ij}^n and velocity floating regulation points h_{ij}^n and introduce another update strategy for aligning the choice of s_{kl}^n in a subsequent step by enforcing $\forall l$

$$\xi_{ml} = \sum_k \tilde{V}_k(\tilde{\xi}_m) \cdot s_{kl}^n = \sum_i \tilde{N}_i(\tilde{\xi}_m) \cdot h_{ij}^n = \xi_{mj}, \quad j = 1 + \text{round}\left(\frac{l-1}{L-1}(J-1)\right). \quad (40)$$

The $\tilde{\xi}_m$ with $m = 1, \dots, K$ can be considered as collocation points in the parent domain, which we choose as Greville abscissae. In this way, we determine all s_{kl}^n such that the parent collocation points are mapped to the same parametric characteristic points ξ_{ml} as the floating maps for velocity would do. To this end, due to the subdivision of the normal basis functions, the indices l and j must be associated, here for simplicity by an average scaling on the index numbers. The s_{kl}^0 in the initial step may already be derived by the collocation scheme given above and satisfy similarly to the velocity floating maps $\forall l : \xi = \mathcal{H}_l^0(\tilde{\xi}) = \tilde{\xi}$.

3.6. Material point integration

As mentioned earlier, a parametric point (e.g. a Gauss point) in IGA is always associated to the same material point during its movement through time and physical space. In FLIGA, one natural consequence of breaking the Lagrangian character of the physical mapping is that one physical material point occupies different positions in the parametric and in the parent space during the course of the analysis. Thus, we cannot preserve both material and Gauss point character at the same time.

In FLIGA, we employ the material point integration strategy, which is well known from meshless methods, see e.g. Kumar et al. [11,34]. This technique attributes to the quadrature points a Lagrangian particle character in the sense of (10). In the following, we elaborate on the derivation of these points and their quadrature weights, defining integration by (23).

At the initial time step $n = 0$, we construct a standard Gauss integration scheme for the velocity mesh, which corresponds to a classical IGA mesh at this stage. This provides parametric material points ξ_q^0 and parametric weights \hat{W}_q^0 , which are well suited for quadrature at this time step. We map these to the physical domain, obtaining

$$\begin{aligned} \mathbf{x}_q^0 &= \mathcal{F}^0(\xi_q^0), \\ W_q^0 &= \hat{W}_q^0 \cdot \det(\mathbf{J}^0(\xi_q^0)). \end{aligned} \quad (41)$$

We then evolve this quadrature set in the physical space during the analysis, following the deformation of the material:

$$\begin{aligned} \mathbf{x}_q^{n+1} &= \mathbf{x}_q^n + \mathbf{v}^n(\mathbf{x}_q^n) \cdot \Delta t, \\ W_q^{n+1} &= W_q^n \cdot \det(\mathbf{I} + \nabla_x \mathbf{v}^n(\mathbf{x}_q^n) \cdot \Delta t). \end{aligned} \quad (42)$$

In order to obtain the parametric material point positions at which the basis functions have to be evaluated according to Section 3.3, we have to map the physical positions back to the parametric space, i.e.

$$\xi_q^n = \mathcal{F}^{n-1}(\mathbf{x}_q^n). \quad (43)$$

This is accomplished by a Newton–Raphson scheme described in Appendix G.

Note that material point integration may result in excessive and/or highly irregular spacings between material points when deformations are too complex. Such phenomena also affect material point methods [35], and remedies from such methods may be adapted to FLIGA in future work. In any case, for the purpose of the present investigation even standard material point integration appears well suited. This is because, when a characteristic direction dominates the deformation as assumed in this work, the particle distribution typically stays relatively regular. Finally, material point integration in FLIGA clearly profits from the similarly deformation-dependent update of control points, which ensures a compatible alignment of material points with control points throughout the simulation.

The absence of advective terms makes material point integration particularly attractive for viscoelastic, inertial or non-isothermal flow, which otherwise would often require stabilization. Material derivatives from the continuous model are naturally discretized by

$$\left. \frac{D}{Dt}(\bullet)_q \right|_{t=t^{n-1}} \approx \frac{(\bullet)_q^n - (\bullet)_q^{n-1}}{\Delta t}, \quad (44)$$

where $(\bullet)_q$ and $(\bullet)_q^n$ are the respective continuous and discrete quantity carried by the material point \mathbf{x}_q .

3.7. Code structure

We conclude the introduction of FLIGA by brief remarks on a potential code structure, see Algorithm 1.

The initial step of the analysis procedure is the collection of relevant data, i.e. parameters specifying material properties, loading conditions and numerical parameters. Afterwards, parent and normal knot vectors are set up according to the desired element subdivision, and the Greville point set ξ_m for the later updates (40) of the pressure floating regulation points is created. The definition of the polynomial order, of the initial floating regulation points for velocity as well as of the initial control points completes the geometry preparation for the first time step. Also,

Algorithm 1: Analysis procedure

Input: Problem data

- 1 **Procedure:** Floating isogeometric analysis
- 2 define: knot vectors; Greville points $\tilde{\xi}_m$; polynomial order: r ; initial configuration: $h_{ij}^0, \mathbf{c}_{ij}^0, \mathbf{x}_q^0, W_q^0$
- 3 provide tools for velocity floating regulation point update: $\nabla_x \mathcal{L}, \gamma^n$
- 4 **for** time step $n = 0, 1, \dots$ **do**
- 5 compute floating regulation points for pressure: s_{kl}^n
- 6 compute parametric material points: ξ_q^n
- 7 compute basis functions and connectivities at each material point: $A_a^n, B_b^n, \nabla_x A_a^n, \nabla_x B_b^n, \mathcal{A}^n, \mathcal{B}^n$
- 8 update definition of boundary conditions
- 9 **for** Newton-Raphson step $r = 0, 1, \dots$ **do**
- 10 **for** material point index $q = 0, 1, \dots, Q$ **do**
- 11 assemble material point contribution to residual vector: $\mathbf{R}^{n,r} += \mathbf{R}_q^{n,r}$
- 12 apply Dirichlet boundary conditions on $\mathbf{R}^{n,r}$
- 13 **if** $\|\mathbf{R}^{n,r}\| < \text{tol}$ **then**
- 14 time step solution: $\mathbf{u}^n \leftarrow \mathbf{u}^{n,r}$
- 15 leave Newton-Raphson loop
- 16 **for** material point index $q = 0, 1, \dots, Q$ **do**
- 17 assemble material point contribution to tangent stiffness matrix: $\mathbf{K}^{n,r} += \mathbf{K}_q^{n,r}$
- 18 apply Dirichlet boundary conditions on $\mathbf{K}^{n,r}$
- 19 update solution guess by solving Newton-Raphson step: $\mathbf{u}^{n,r+1} = \mathbf{u}^{n,r} - (\mathbf{K}^{n,r})^{-1} \mathbf{R}^{n,r}$
- 20 update: $\mathbf{c}_{ij}^{n+1}, h_{ij}^{n+1}, \mathbf{x}_q^{n+1}, W_q^{n+1}$, internal variables

Output: Approximate fields: \mathbf{v}^n, p^n

we distribute material points as Gaussian quadrature points of this initial classical IGA mesh. Lastly, we design a suitable level function and blending function, which will be required for the later updates (33) of the velocity floating regulation points. Note that the design of level and blending function is not automatic, but requires some knowledge on the expected deformations.

Having specified the problem setup, a time step loop is started. Floating regulations points for pressure are computed from those for velocity (40). Parametric material points are obtained by mapping back current physical positions of the material points (43). This provides the information to compute for each material point the values of the supported basis functions and the control point connectivity (Section 3.3). The treatment of Dirichlet boundary conditions is identical as in IGA (Section 2.4).

For solving the global system of discrete equations (Section 2.3), we apply a Newton–Raphson procedure (Appendix A) and therefore introduce a nested iteration loop. Within this loop, we first compute the residual vector for a current guess of the unknowns. Other than in IGA, we do not assemble quadrature point contributions to intermediate element vectors. Instead, assembly occurs directly into the global system. To this end, we do not iterate over elements, but once over all material points. Contributions are computed for the global system (21) and (22), where we approximate integrals by (23) and apply material point integration (Section 3.6). Dirichlet boundary conditions are now incorporated to the global residual vector.

Next, the residual norm is computed. If this norm lies below a prescribed tolerance, the Newton–Raphson procedure is considered converged. Otherwise, we assemble all material point contributions to the tangent stiffness matrix in similar manner as previously done for the residual vector. Dirichlet boundary conditions are applied. Finally, the guess of the unknowns is improved and the next Newton–Raphson iteration initiated. This process repeats until the residual tolerance is met.

Upon convergence, the current time step is finalized by updating the locations of control points (24), velocity floating regulations points (33), and material points, as well as the material point weights (42) and possibly the internal variables from the constitutive laws (see Section 4.1). This step requires the evaluation of the level function and current blending function.

Finally, the next time step is initiated and the procedure is repeated.

4. Numerical examples

In this section, we provide insight into the numerical accuracy and stability of FLIGA using three numerical examples: patch test, Taylor–Couette flow, and polymer extrusion. Preliminarily, we specify two constitutive models to be adopted in the examples. We conclude the section with some considerations on the computational cost.

4.1. Constitutive models

In the numerical examples, we will use two material models, namely, the Newtonian fluid and the viscoelastic Oldroyd-B fluid, which are briefly described as follows. For details on their theoretical background we refer to Phan-Thien [36]. In the Newtonian fluid model, $\boldsymbol{\tau} = \boldsymbol{\tau}_s$ is linearly related to the strain rate tensor

$$\boldsymbol{\tau}_s = 2\eta_s \mathbf{D}, \quad (45)$$

where η_s denotes the Newtonian (or solvent) viscosity. The viscoelastic Oldroyd-B model is more complex and accounts for viscous and elastic components. Even under large deformations, it appears well suited for representing material behavior in certain polymer deformation problems. In this model, $\boldsymbol{\tau} = \boldsymbol{\tau}_{OldB}$ is split into two contributions:

$$\boldsymbol{\tau}_{OldB} = \boldsymbol{\tau}_s + \boldsymbol{\pi}. \quad (46)$$

While $\boldsymbol{\tau}_s$ can immediately be adopted from the solvent Newtonian fluid model, $\boldsymbol{\pi}$ accounts for a polymeric contribution and is given in terms of the differential equation in time

$$\frac{D}{Dt} \boldsymbol{\pi} = 2\eta_p \lambda^{-1} \mathbf{D} - \lambda^{-1} \boldsymbol{\pi} + \nabla_x \mathbf{v} \cdot \boldsymbol{\pi} + \boldsymbol{\pi} \cdot \nabla_x^T \mathbf{v}. \quad (47)$$

Here, λ is the relaxation modulus and η_p is the polymeric viscosity. We require an initial condition $\boldsymbol{\pi}(t=0) = \boldsymbol{\pi}^0$ to complete the problem statement.

Let us now seek the discretized versions of the deviatoric stresses defined above. From now on, we will remain in the discretized setting. In the case of the Newtonian fluid, the task is trivial. Here suffices a replacement of the continuous strain rate in (45) by the discretized strain rate \mathbf{D}^n . This yields the discretized solvent deviatoric stress $\boldsymbol{\tau}_s^n$. For the Oldroyd-B model, we further require the discretized polymeric deviatoric stress $\boldsymbol{\pi}^n$. Using (44) for material point integration, we discretize (47) obtaining $\boldsymbol{\pi}^n$ at material point \mathbf{x}_q as

$$\boldsymbol{\pi}_q^n = \boldsymbol{\pi}_q^{n-1} + \frac{\Delta t}{\lambda} [2\eta_p \mathbf{D}_q^{n-1} - \boldsymbol{\pi}_q^{n-1}] + \Delta t [\nabla_x \mathbf{v}_q^{n-1} \cdot \boldsymbol{\pi}_q^{n-1} + \boldsymbol{\pi}_q^{n-1} \cdot \nabla_x^T \mathbf{v}_q^{n-1}]. \quad (48)$$

These computations do not require unknown information from the current time step, yielding an explicit time integration. History information from the previous step is available and stored as internal variable at the material point. The total deviatoric Oldroyd-B stress $\boldsymbol{\tau}_{OldB}^n$ is finally the sum of discrete solvent and discrete polymeric stress contribution.

For setting up the tangent stiffness matrix, the linearizations of the deviatoric Cauchy stresses $\boldsymbol{\tau}_s^n$ and $\boldsymbol{\tau}_{OldB}^n$ w.r.t. the current control point velocities are required, see Appendix H. Assembling all information, we give a compact description of the global tangent stiffness matrix for implementation purposes in Appendix I. The governing equations are linear in control point velocity and pressure. However, the final system may still lose its linearity after application of nonlinear boundary conditions.

Alternative to the explicit time integration scheme proposed here, one could also apply an implicit analog. Being unconditionally stable, such an implicit formulation would lower the restrictions on the time step size. However, this introduces non-linearity in the governing equations and requires a more demanding linearization.

4.2. Patch test

We begin our numerical investigations by a patch test in order to assess first-order consistency. For simplicity, we adopt dimensionless units. A square domain of Newtonian fluid ($\eta_s = 50$) is subject to the boundary conditions

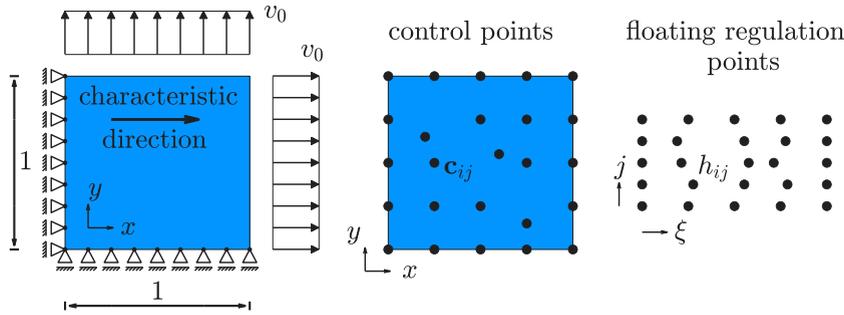


Fig. 8. Patch test problem for FLIGA.

Table 1

Logarithmic relative L^2 errors of velocities in the patch test.

Polynomial order		r=1	$\rightarrow \infty$	r=2	r=3		
No. of quadrature points		$\approx 1e2$		$\approx 1e2$	$\approx 1e6$	$\approx 1e2$	$\approx 1e6$
IGA	x direction	-16.28		-15.67	-14.80	-15.73	-14.47
	y direction	-16.02		-15.85	-13.69	-15.63	-14.26
FLIGA	x direction (characteristic)	-1.93	lin. convergence	-2.46	-10.76	-2.88	-11.73
	y direction (normal)	-2.86		-3.78	-12.18	-3.94	-12.19

illustrated in Fig. 8, with $v_0 = 1$. We solve (21), where we set the pressure to zero and thus do not need the incompressibility constraint (22). The analytical velocity solutions are given by the monomials

$$\begin{pmatrix} v_x \\ v_y \end{pmatrix} = \begin{pmatrix} x \\ y \end{pmatrix}. \tag{49}$$

We construct FLIGA discretizations where the characteristic direction is associated to the x -axis. We distribute arbitrary 5×5 floating regulation points on $\hat{\Omega}_\xi$ and associate an equivalent number of 5×5 control points. The control points are likewise positioned arbitrarily, where their irregular distribution is the same for all studied polynomial orders. According to the constant control point number, we derive for each of these polynomial orders equal-sized element divisions for parent and normal domain. Quadrature is carried out by material point integration with either a moderate or an overkill number of points. It is not possible to define a classical Gauss point set for the floating meshes, however, we choose for the material points only those positions that are Gauss points of at least one supported basis function. For comparison, we also test classical IGA with Gauss integration, for which the floating regulation points are all equidistantly aligned in order not to introduce a floating of basis functions.

Results of the patch test are given in Table 1 and show the logarithm of the L^2 norm of the relative error for the approximate velocity solutions. Classical IGA passes the test for all polynomial orders r , all quadrature sets and both velocity components. On the other hand, we find FLIGA to fulfill the patch test only when overkill integration is applied. This indicates that FLIGA B-Splines fulfill first-order consistency up to the error induced by the numerical quadrature.

For overkill integration of linear FLIGA B-Splines we do not provide specific values but the observation that the L^2 norm of the relative error converges towards zero with errors being $\mathcal{O}(Q^{-1})$ in the point number Q . For instance, $\{10^2, 10^3, 10^4, \dots\}$ quadrature points yield logarithmic errors of approximately $\{-2, -3, -4, \dots\}$. We checked this behavior up to errors of -8.5 , a further increase of quadrature points being exceedingly expensive.

4.3. Taylor–Couette flow: general remarks

We now consider the benchmark problem of inertialess Taylor–Couette flow, see the setup in Fig. 9. We study the flow behavior of a fluid comprised between two concentric cylinders, with inner and outer radii R_I and R_O , respectively, see Table 2. We let the outer cylinder rotate clockwise with an angular velocity Ω_O , whereas the inner cylinder is clamped. Assuming an infinite length in the out-of-plane direction, we employ a two-dimensional model

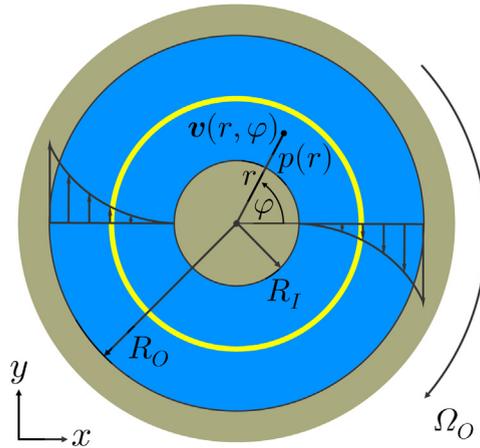


Fig. 9. Taylor–Couette problem.

Table 2

Parameters for the Taylor–Couette setup.

Parameter	Notation	Value
Inner radius	R_I	≈ 100 mm
Outer radius	R_O	≈ 200 mm
Angular velocity	Ω_O	7.5 s $^{-1}$

where velocities in the out-of-plane direction vanish. Each fluid particle is then characterized by an in-plane velocity vector and a pressure which depend only on its radial position. We assume the material to have either Newtonian or Oldroyd-B constitutive behavior. For these two material models, a stability analysis of the continuous Taylor–Couette model predicts instabilities in some ranges of problem parameters, see Bai [37] and references therein. Here we choose the parameters such that stability is preserved.

We choose a characteristic parent domain $\tilde{\Omega} = [0, 10]$ and a parametric domain $\hat{\Omega}^n = \hat{\Omega} = [0, 10] \times [0, 2]$. The specific nature of Taylor–Couette flow requires the application of periodic boundary conditions to the parent domain. These act in characteristic direction and the modeling details are provided in Appendix J. At the initial time step, the boundary control points are distributed point symmetrically between inner radius $R_I^{CP} = 100$ mm and outer radius $R_O^{CP} = 200$ mm. Different refinements are studied.

As level function we employ

$$\mathcal{L}(\mathbf{x}) = -10 \cdot \frac{\varphi}{2\pi} = -\frac{10}{2\pi} \arctan(y/x), \quad \nabla_{\mathbf{x}} \mathcal{L}(\mathbf{x}) = \frac{10}{2\pi} \begin{pmatrix} \frac{y}{x^2 + y^2} \\ -\frac{x}{x^2 + y^2} \end{pmatrix}. \quad (50)$$

This choice associates points along each radial line in the physical space to the same parametric level. A blending to ensure a rectangular parametric domain structure is not required due to the applied periodic boundary condition, hence $\gamma^n(\xi) = 1$.

The imposition of boundary conditions is carried out first on all outer boundary control points of pressure and velocity, with index sets \mathcal{A}_{outer} and \mathcal{B}_{outer} , respectively. Dirichlet boundary conditions are homogeneous for pressure, i.e. $q_a^n = 0$ Pa $\forall a \in \mathcal{A}_{outer}$, and inhomogeneous for velocity

$$\mathbf{d}_b^n = \begin{pmatrix} d_{xb}^n \\ d_{yb}^n \end{pmatrix} = \Omega_O \begin{pmatrix} c_{yb}^n \\ -c_{xb}^n \end{pmatrix} \quad \forall b \in \mathcal{B}_{outer}. \quad (51)$$

Inner boundary velocity control points with index set \mathcal{B}_{inner} are subjected only to homogeneous Dirichlet boundary conditions on velocity, $\mathbf{d}_b^n = \mathbf{0}$ mm/s $\forall b \in \mathcal{B}_{inner}$.

The accuracy is quantified with the logarithm of the L^2 norm of the relative error in velocity and pressure with respect to the analytical reference solution. Computation of the reference results requires evaluation of the current

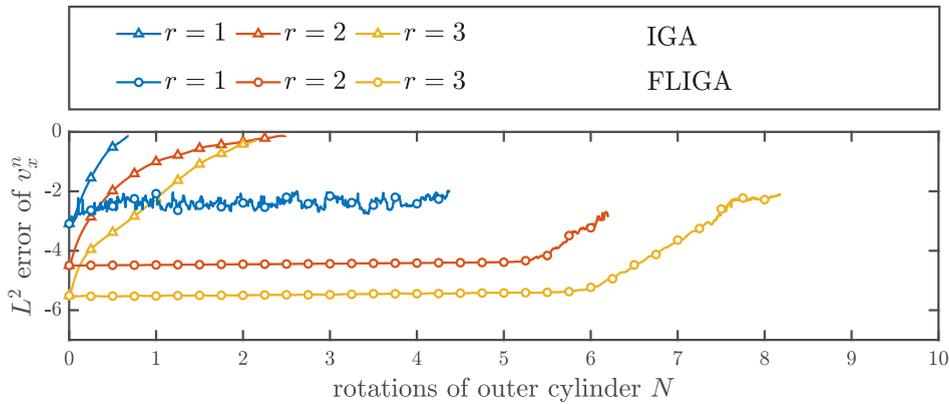


Fig. 10. Error over deformation in the Newtonian Taylor–Couette benchmark for polynomial degrees $r = 1, 2, 3$.

radii R_I and R_O . We evaluate these quantities at each time step from the current mesh boundary, as the radius of the outer boundary increases slightly during the simulation due to the incremental time stepping procedure. This effect is not due to FLIGA but to time integration and hence we compensate for it in our post-processing by recomputing the reference solution.

4.4. Taylor–Couette flow: Newtonian fluid

For this constitutive model, we adopt a solvent viscosity $\eta_s = 50$ Pa s. Bai [37] provides the following analytical solution for the horizontal velocity profile

$$v_{Newt,x}(r, \varphi) = \sin(\varphi) \left(\alpha r + \frac{\beta}{r} \right), \tag{52}$$

in which

$$\alpha = \frac{\Omega_O R_O^2 - \Omega_I R_I^2}{R_O^2 - R_I^2}, \tag{53}$$

$$\beta = \frac{(\Omega_I - \Omega_O) R_I^2 R_O^2}{R_O^2 - R_I^2}. \tag{54}$$

In our non-inertial Newtonian case, we do not expect pressure variations to occur and thus obtain for boundary condition $p_{Newt}(R_O) = 0$ Pa

$$p_{Newt}(r) = 0 \text{ Pa}. \tag{55}$$

We adopt $\Delta t = 5.0e - 5$ s, however the time step does not have a significant influence on the results for the present setup.

As follows, we illustrate the numerical results obtained using FLIGA as well as classical IGA (Section 2) for reference. We first evaluate the L^2 norm of the relative error as a function of the deformation. We investigate linear, quadratic and cubic polynomial B-Spline orders, i.e. $r = 1, 2, 3$, where the initial mesh consists of 18×6 pressure and 36×12 velocity elements. We initialize both IGA and FLIGA mesh with Gauss integration using $(r+1) \times (r+1)$ points.

The results are reported in Fig. 10. It can be seen that the accuracy of IGA deteriorates at relatively small deformations, which is expected due to distortion of the classical mesh. Instead, with FLIGA the error stays nearly constant throughout the floating of the mesh for a remarkably large range of deformations. As the polynomial order r increases, the accuracy of the results improves, however the computational cost also increases. The lack of smoothness is probably the reason for the lower accuracy obtained with linear basis functions ($r = 1$) in combination with material point integration (note that quadrature points have to pass discontinuities in FLIGA). At present, the

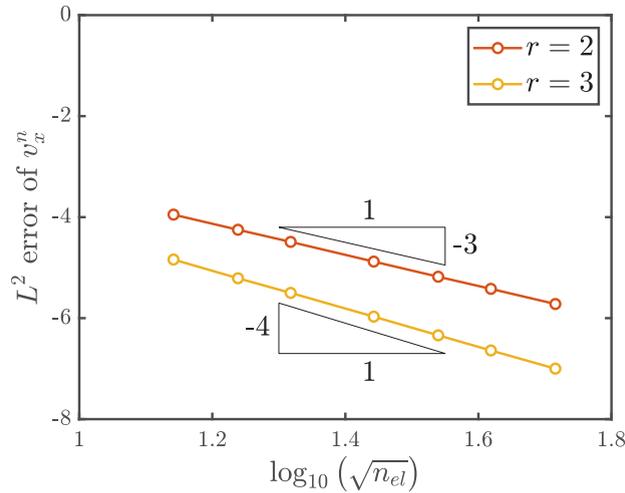


Fig. 11. Error plateau over spatial refinement in the Newtonian Taylor–Couette benchmark.

exact reason for the observed divergence after 5 – 6 rotations ($r = 2, 3$) is unclear. We suspect it to be related to loss of quadrature accuracy and are going to investigate and improve this aspect further in the future.

The formation of an error plateau allows for an easy evaluation of convergence properties under mesh refinement. Fig. 11 illustrates the L^2 norm of the relative velocity error plateau for polynomial orders $r = 2, 3$ over spatial refinement. These results agree well with what is expected from classical IGA, as the floating mesh initially conforms with such a discretization, the convergence rate being close to $r + 1$.

4.5. Taylor–Couette flow: Oldroyd-B fluid

For the viscoelastic Taylor–Couette problem with the Oldroyd-B constitutive model, we choose a polymeric and a solvent viscosity of $\eta_p = 150$ Pa s and $\eta_s = 50$ Pa s, respectively. The initial condition for the polymeric stress reads $\pi^0 = \mathbf{0}$ Pa and the relaxation time is chosen as $\lambda = 0.1$ s. Hence, we obtain a Weissenberg number of

$$Wi = \lambda \cdot \dot{\gamma} = \lambda \cdot \frac{\Omega_O R_O}{R_O - R_I} \approx 1.5, \tag{56}$$

characterizing the ratio of elastic to viscous forces in our benchmark flow. It is well known that most standard numerical schemes are only applicable to viscoelastic simulations up to $Wi \approx 1$ [38]. Here, we maximize the elastic component by entering this regime, but do not investigate a further increase which is generally critical in the absence of stabilization.

As before, we compare FLIGA and IGA results with the analytical reference solution using the L^2 norm of the relative error. Note that in the viscoelastic case we have a transient stress response, which starts from the initial polymeric stress condition and approaches a steady state. We investigate only the steady state and use the reference solution in Bai [37]. The analytical steady state velocity field equals that of the Newtonian case, $v_{OldB,x}(r, \phi) = v_{Newt,x}(r, \phi)$. However, for the pressure boundary condition $p_{OldB}(R_O) = 0$ Pa, we now expect a different pressure profile at steady state

$$p_{OldB}(r) = 2\beta^2\lambda\pi \cdot \left(\frac{1}{r^4} - \frac{1}{R_O^4} \right), \tag{57}$$

with β given by (54).

Once again we evaluate the L^2 norm of the relative error as a function of the deformation, where we now decrease the time increment to $\Delta t = 1.0e - 5$ s for $r = 1, 2$, and $\Delta t = 5.0e - 6$ s for $r = 3$ due to the transient character of the problem. We adopt again 18×6 pressure and 36×12 velocity elements for the initial mesh and initialize material point integration on this mesh by distributing $(r + 1) \times (r + 1)$ Gauss points per element.

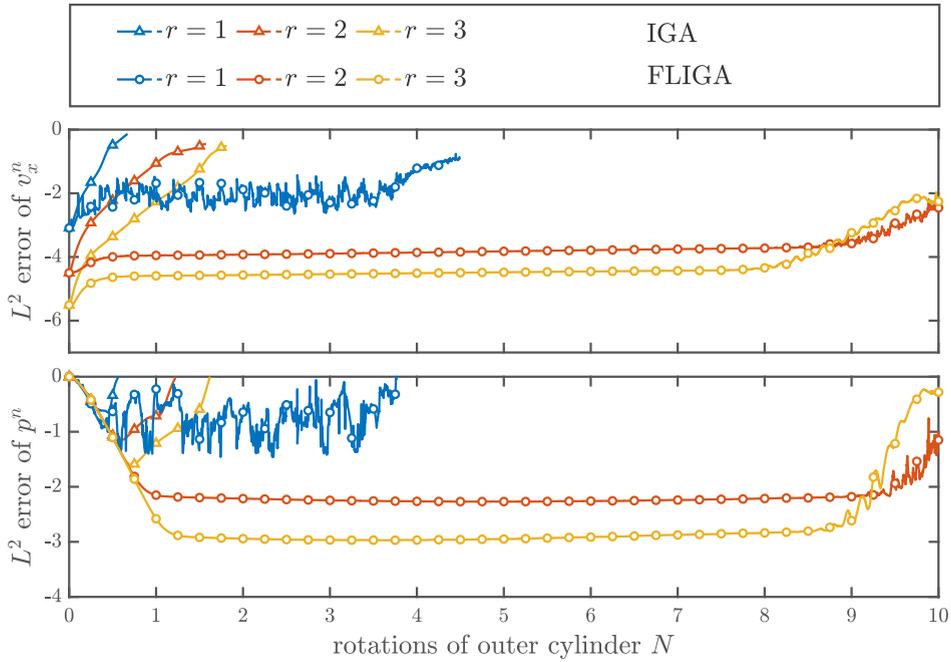


Fig. 12. Error over deformation in the viscoelastic Taylor–Couette benchmark for polynomial degrees $r = 1, 2, 3$.

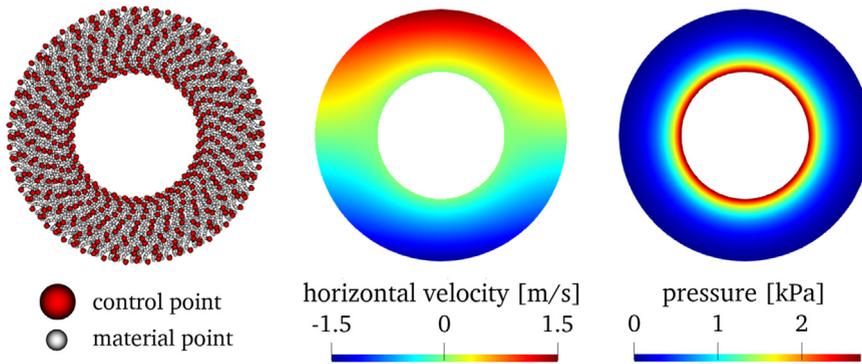


Fig. 13. Approximate horizontal velocity and pressure fields after $N = 2$ rotations of the outer cylinder.

Fig. 12 shows at the top the error versus deformation for the horizontal velocity component. The error norm starts at a low level which corresponds to the Newtonian result since viscoelastic behavior is not yet active. As elastic contributions come into play, the accuracy slightly decreases and the error reaches a plateau at steady state. The effect of the polynomial degree on the accuracy agrees well with the previous observations in the Newtonian case. Surprisingly, for $r = 2, 3$ the extent of stable deformation is even significantly larger than in the Newtonian case. At the bottom, Fig. 12 displays the error versus deformation for the pressure field. As the analytical pressure given by (57) is valid only at steady state, the error is large in the initial state where, correctly, the numerical pressure is zero. Afterwards, again a plateau in the error is reached at steady state. Similar observations as for the velocity error hold in this case.

The approximate velocity and pressure fields for $r = 3$ after $N = 2$ rotations of the outer cylinder are shown in Fig. 13. Both fields are free from oscillations, which is an indication of the absence of locking effects.

We conduct another mesh refinement study for $r = 2, 3$, see Fig. 14. The parameters are the same as before, but the time step is adjusted for fine floating meshes to guarantee that the error is controlled by the spatial discretization.

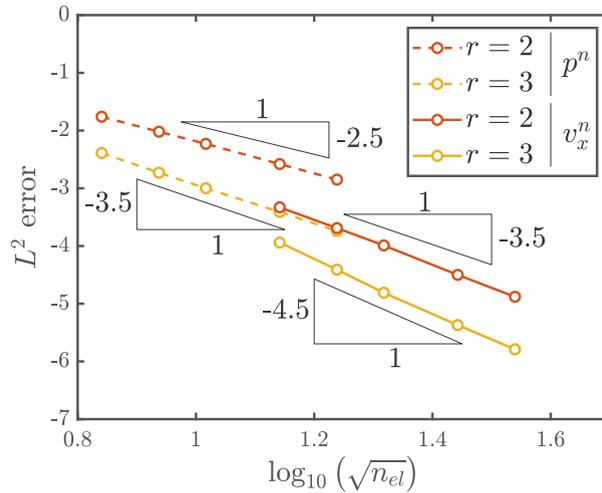


Fig. 14. Error plateau for the steady state of pressure and velocity over spatial refinement in the Oldroyd-B Taylor–Couette benchmark.

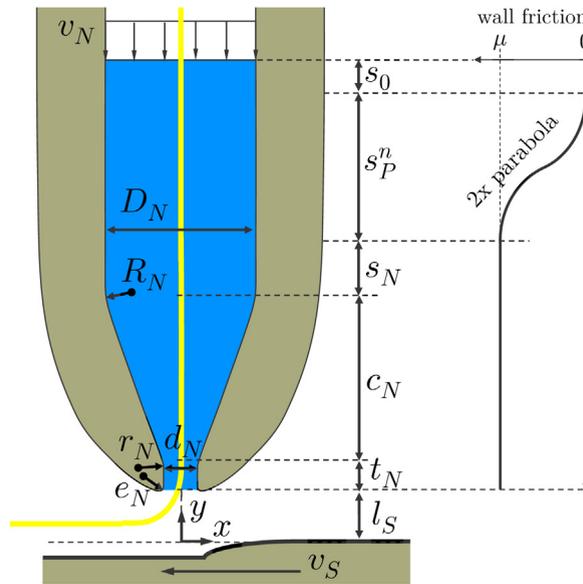


Fig. 15. Schematic extrusion setup.

We obtain for both velocity and pressure a convergence rate that even exceeds the expected order of $r + 1$ and r , respectively.

4.6. Extrusion-based AM

In this third numerical example, we demonstrate the suitability of FLIGA for the simulation of polymer extrusion processes in AM, which was our original motivation for FLIGA. The problem setup consists of an extrusion nozzle, that is initially filled with material, see Fig. 15 for details. The contact between material and nozzle walls is described by a node-to-surface frictional contact formulation (Coulomb friction), where the contact constraints are enforced with the penalty method. To facilitate the employment of a plug flow boundary condition driving the extrusion, the friction factor fades in from no wall friction at the upper domain area (s_0) to a plateau. This is modeled by two equal-sized sections of parabolic increase, see again the figure for more details. Note that we keep all measures

describing the nozzle constant throughout the simulation except the transition area for wall friction, s_p^n , which is determined by the current coordinate level of the top boundary. Values for wall penalty factor, plateau friction coefficient as well as the other parameters are given in Table 3.

Additionally, we impose on the two top control point rows ($i = 1, 2$) the condition

$$d_{x(i)(j)}^n = d_{x(\alpha)(j)}^{n-1}, \tag{58}$$

with $\alpha = 11$, which has a stabilizing effect on the horizontal deformation in the top region and only negligibly affects the overall simulation result.

We employ the Oldroyd-B viscoelastic model with initial condition for the polymeric stress $\boldsymbol{\pi}^0 = \mathbf{0}$ Pa. Once the extrudate reaches below a threshold vertical coordinate, it attaches irreversibly to a substrate modeled by Dirichlet boundary conditions. The substrate level is adapted so as to avoid a singularity in the physical mapping at points associated to the corners of the parametric domain (see Fig. 15). As inertia forces are absent from the model, instead of moving the nozzle, we can impose the printing movement on the substrate [25].

A time step of $\Delta t = 2.0e - 5$ s is selected. Spatial discretization starts with the definition of the polynomial order $r = 2$ for both velocity and pressure. In the initial configuration of a classical mesh, we employ 4 pressure and 8 velocity elements per cross-section and use a large number of elements in extrusion direction (480 for pressure and 960 for velocity). The parent and initial parametric domain read $\tilde{\Omega} = [0, 9]$ and $\hat{\Omega}^0 = [0, 9] \times [-1, 1]$, respectively. We choose material point integration with 8×3 Gauss points per parametric element in the initial configuration.

We employ the following blending function

$$\gamma(\xi) = \begin{cases} 0 & \xi = a^n \\ \text{lin. ramp} & a^n < \xi \leq a^n + 2d \\ 1 & a^n + 2d < \xi \leq b^n - 2d \\ \text{lin. ramp} & b^n - 2d < \xi \leq b^n - d \\ 0 & b^n - d < \xi \end{cases}, \tag{59}$$

with $d = 1.0$ mm. The level function is chosen such that

$$\begin{aligned} \text{before attachment : } \nabla_x \mathcal{L}(\mathbf{x}) &= \begin{pmatrix} 0 \\ -1 \end{pmatrix} \\ \text{once attached : } \nabla_x \mathcal{L}(\mathbf{x}) &= \begin{cases} \begin{pmatrix} 0 \\ -1 \end{pmatrix} & P_{Ry} \leq y \\ \begin{pmatrix} r_y/\sqrt{\mathbf{r} \cdot \mathbf{r}} \\ -r_x/\sqrt{\mathbf{r} \cdot \mathbf{r}} \end{pmatrix} & y < P_{Ry}, P_{Rx} \leq x \\ \begin{pmatrix} -1 \\ 0 \end{pmatrix} & y < P_{Ry}, x < P_{Rx} \end{cases}, \end{aligned} \tag{60}$$

where $\mathbf{r} = \mathbf{x} - \mathbf{P}_R$, with $\mathbf{P}_R = \left(-\frac{d_N}{2} - e_N - 0.1, l_s + e_N + 0.1\right)^T$ for the coordinate system defined in Fig. 15. This choice defines a level function as qualitatively illustrated in Fig. 6.

Fig. 16 shows the obtained solution fields prior to deformation and during extrusion at time $T \approx 0.4$ s. The upper part of the figure shows the vertical velocity field. At a given horizontal level in the deformed state, the absolute velocity at the nozzle walls is always observed to be lower than at the center of the nozzle, which is due to wall friction. The significant increase of vertical velocity along the nozzle contraction is due to the incompressibility constraint. Also, we observe viscoelastic swelling after the polymer leaves the confinement of the nozzle. In the bottom part of the figure, the pressure field is shown. This is free from oscillations, indicating the absence of locking. Further, the pressure in the deformed state decreases along the contraction zone and finally becomes close to zero at the nozzle exit. This behavior is consistent with observations in the AM literature, see e.g. [39].

The position of a selection of white dots aligned horizontally at the initial time step is tracked graphically to qualitatively illustrate the extreme deformations occurring during extrusion, which are successfully handled by FLIGA.

Table 3
Parameters for the extrusion setup.

Parameter	Notation	Value
Inlet area	s_0	0.25 mm
Parabolic friction area at $t = 0$	s_P^0	5.15 mm
Straight section	s_N	1.1 mm
Contraction area	c_N	2.0 mm
Nozzle tip	t_N	0.8 mm
Substrate distance	l_S	0.525 mm
Inlet diameter	D_N	2.0 mm
Outlet diameter	d_N	0.5 mm
Convergent radius	R_N	2.5 mm
Divergent radius	r_N	1.2 mm
Nozzle exit radius	e_N	0.3 mm
Inlet velocity	v_N	10.0 mm/s
Substrate speed	v_S	38.0 mm/s
Wall penalty factor	k	1.0e8
Plateau friction coefficient	μ	0.075
Relaxation time	λ	0.04 s
Polymeric viscosity	η_P	1.5 kPa · s
Solvent viscosity	η_S	0.05 kPa · s

4.7. Computational cost

We conclude this section with some brief considerations on the computational cost of FLIGA in comparison with existing techniques. A comparison with IGA may make little practical sense, since IGA simply cannot handle the extreme deformations that become possible with FLIGA; on the other hand, such comparison may still provide a useful term of reference for IGA researchers. While a rigorous and fair cost comparison would require code optimization for both and is deemed out of scope for the present investigation, we generally observe that the computational cost of FLIGA is of the same order of magnitude but higher than that of IGA, which can be attributed mainly to the following additional tasks in FLIGA:

- determination of parametric material point positions,
- recomputation of basis functions and derivatives,
- recomputation of floating regulation points for velocity and pressure.

The condition number of the final global linear system may also be larger in FLIGA. In our experience with a non-optimized code for the Newtonian Taylor–Couette flow with $r = 2$, the simulation of the first rotation with FLIGA costs slightly less than twice the same simulation with IGA. The cost overhead becomes less significant as the polynomial order increases.

A perhaps more significant cost comparison would involve, rather than IGA, the CFD computational technologies which are currently able to tackle problems with extreme deformations. We have not performed any of such comparisons for the time being.

5. Conclusions

We propose Floating Isogeometric Analysis (FLIGA) as a novel extension of classical IGA. FLIGA enables the solution of problems featuring extreme deformations which occur predominantly along one characteristic axis. The basic idea is a modification of the classical tensor-product structure of B-Spline basis functions, such that multiple B-Spline chains in the characteristic direction are able to independently float against each other in the parametric domain. Following this strategy, distortion introduced by the physical mapping is overcome without the need for classical remeshing. FLIGA enjoys several benefits of isogeometric and meshless modeling: (i) the initial geometry representation is a classical IGA B-Spline mesh preserving the isogeometric principle, (ii) several properties of B-Splines favorable for analysis are preserved e.g. partition of unity, weak Kronecker-delta property and first-order consistency, (iii) moving domain boundaries are naturally dealt with, (iv) mesh distortion is overcome, (v) we

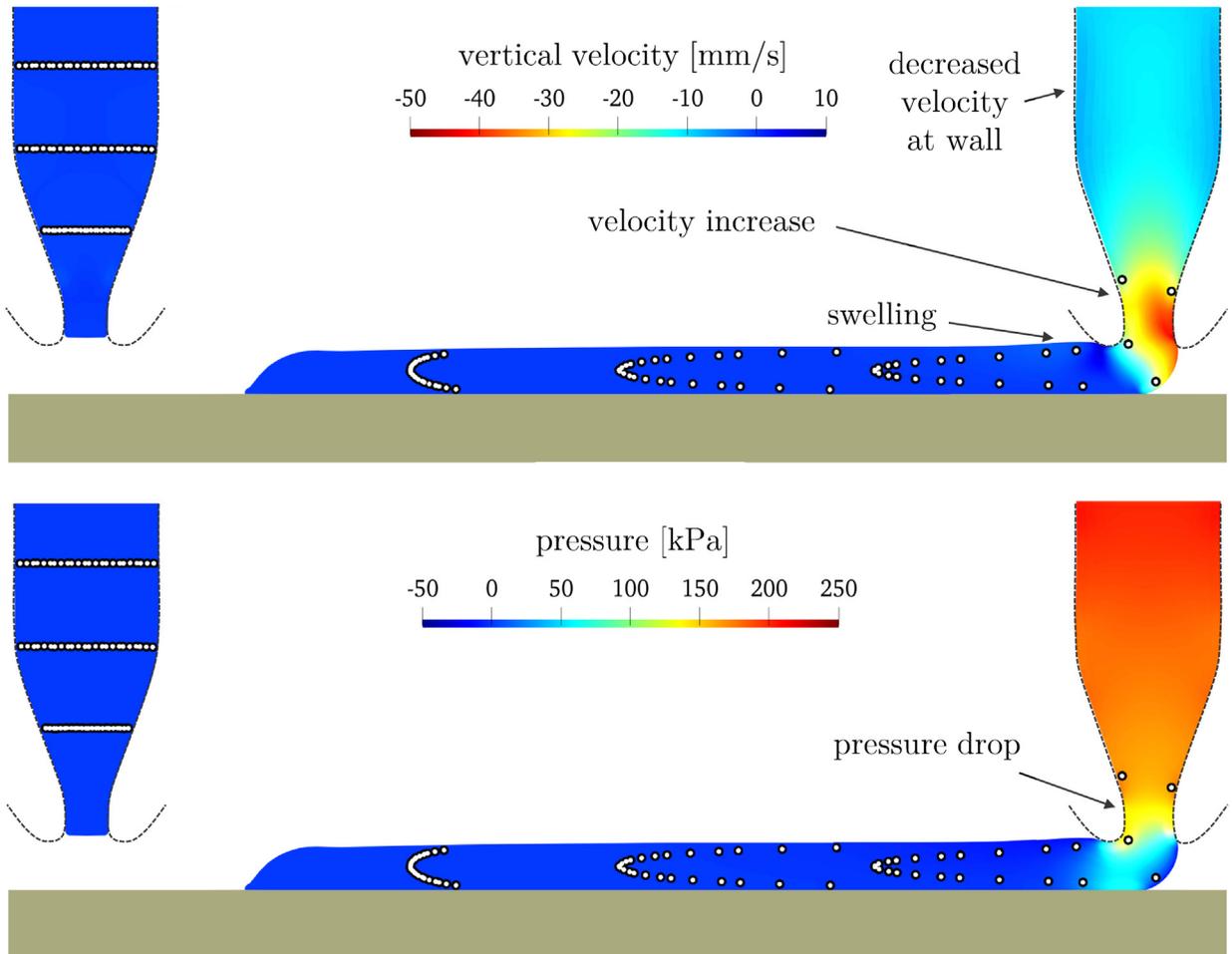


Fig. 16. Extrusion simulation with FLIGA. The white dots highlight material points which were horizontally aligned at the initial step.

can easily construct mixed approximation spaces to avoid locking due to e.g. incompressibility, (vi) there is no need to stabilize advective terms due to the Lagrangian quadrature point character. The effectiveness and accuracy of FLIGA were demonstrated and quantified with the benchmark problem of Taylor–Couette flow with Newtonian and viscoelastic Oldroyd-B material models. Accurate results were obtained up to extreme deformation levels while keeping the spatial convergence rate expected for IGA. Finally, FLIGA was applied to the viscoelastic simulation of extrusion-based additive manufacturing, showing stable and qualitatively accurate results.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Appendix A. Newton–Raphson scheme for solving the governing system of equations

To compactly formulate the global Newton–Raphson scheme, we summarize at the current time step all unknowns in a single vector \mathbf{u} and all residual components in a single vector \mathbf{R} . We introduce \mathbf{K} as tangent stiffness matrix, i.e. the linearization of the residual vector w.r.t. to the unknowns. For brevity, we omit here the time step index n .

With

$$\mathbf{u} = \begin{bmatrix} q_a \\ d_{bi} \end{bmatrix}, \quad \mathbf{R} = \begin{bmatrix} Q_a \\ S_{bi} \end{bmatrix}, \quad \mathbf{K} = \frac{\partial \mathbf{R}}{\partial \mathbf{u}} = \begin{bmatrix} \frac{\partial Q_a}{\partial q_\alpha} & \frac{\partial Q_a}{\partial d_{\beta k}} \\ \frac{\partial S_{bi}}{\partial q_\alpha} & \frac{\partial S_{bi}}{\partial d_{\beta k}} \end{bmatrix}, \quad (\text{A.1})$$

the contributions to the tangent stiffness matrix read as follows:

$$\frac{\partial Q_a}{\partial q_\alpha} = 0, \quad \frac{\partial Q_a}{\partial d_{\beta k}} = \int_{\Omega} \frac{\partial B_\beta}{\partial x_k} A_a \, d\Omega, \quad (\text{A.2})$$

$$\frac{\partial S_{bi}}{\partial q_\alpha} = \int_{\Omega} \left(-A_\alpha \delta_{ij} + \frac{\partial \tau_{ij}}{\partial q_\alpha} \right) \frac{\partial B_b}{\partial x_j} \, d\Omega, \quad \frac{\partial S_{bi}}{\partial d_{\beta k}} = \int_{\Omega} \frac{\partial \tau_{ij}}{\partial d_{\beta k}} \frac{\partial B_b}{\partial x_j} \, d\Omega. \quad (\text{A.3})$$

The linearization of the deviatoric stress is given in [Appendix H](#).

The Newton–Raphson scheme solves the nonlinear vector equation

$$\mathbf{R}(\mathbf{u}) = \mathbf{0}, \quad (\text{A.4})$$

by the iterative formula

$$\mathbf{u}^{(r+1)} = \mathbf{u}^{(r)} - \mathbf{K}^{-1} \mathbf{R}(\mathbf{u}^{(r)}). \quad (\text{A.5})$$

Starting from the initial guess $\mathbf{u}^{(0)}$ (that can be obtained from the previous time step) we repeat application of this formula until $\|\mathbf{R}(\mathbf{u}^{(r^*)})\| < \text{tol}$ and thus identify $\mathbf{u} \leftarrow \mathbf{u}^{(r^*)}$. We choose $\text{tol} = 1.0e - 9$.

Appendix B. Newton–Raphson scheme for the determination of the parent quadrature point position

In order to determine $\tilde{\xi}_{qj}$ (time step index n omitted) as defined by Eq. (34), we introduce the nonlinear scalar equation

$$R_j(\tilde{\xi}) = \mathcal{G}_j(\tilde{\xi}) - \xi_q, \quad (\text{B.1})$$

based on Eq. (27) with the unknown parent coordinate being the root $\tilde{\xi}_{qj} : R_j(\tilde{\xi}_{qj}) = 0$. Given the parametric quadrature point coordinate ξ_q and J_j as the Jacobian of \mathcal{G}_j Eq. (29), we now apply the Newton–Raphson scheme

$$\tilde{\xi}_{qj}^{(r+1)} = \tilde{\xi}_{qj}^{(r)} - J_j(\tilde{\xi}_{qj}^{(r)})^{-1} R_j(\tilde{\xi}_{qj}^{(r)}), \quad (\text{B.2})$$

by starting from the initial guess $\tilde{\xi}_{qj}^{(0)}$ (obtained from the previous time step) until at iteration $r^* : \|R_j(\tilde{\xi}_{qj}^{(r^*)})\| < \text{tol} \approx 1e - 10$. Finally, we obtain $\tilde{\xi}_{qj} \leftarrow \tilde{\xi}_{qj}^{(r^*)}$.

Note that an initial guess might not be available when a material point has just crossed an element boundary in normal direction. In this case, we can solve Eq. (B.1) once by the bisection method and then apply the proposed Newton scheme again for the subsequent time steps.

Appendix C. Continuity

Basis function continuity in FLIGA is exemplarily illustrated on the parametric domain for the arbitrary floating of an IGA mesh with $r = 2$ and 3×3 elements in [Fig. C.17](#). For both IGA and FLIGA, the continuity of B-Spline approximations along the depicted colored lines is C^{r-1} , assuming equal polynomial order of parent and normal bases and no repetition of inner knots in the respective knot vectors. In the interior blue areas continuity is C^∞ . It is observed that less discontinuity lines of the single B-Splines are aligned in the case of FLIGA.

Appendix D. Partition of unity

We show as follows that FLIGA B-Splines satisfy partition of unity at any point $\mathbf{x}_q \in \Omega$, i.e.

$$\sum_i \sum_j B_{ij}(\mathbf{x}_q) = 1. \quad (\text{D.1})$$

For brevity, we omit the time step index n .

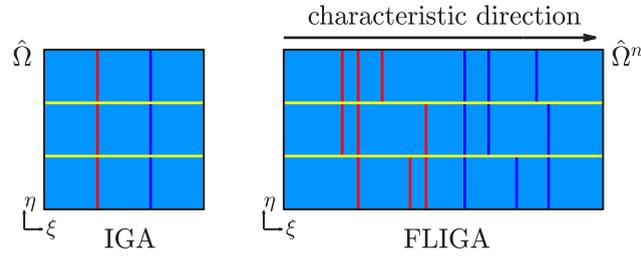


Fig. C.17. Schematic visualization of parametric lines with limited continuity of B-Spline approximations by IGA and FLIGA.

Proof. Let us reformulate the condition employing (37) and (35):

$$1 = \sum_i \sum_j \tilde{N}_i(\tilde{\xi}_{qj}) \hat{M}_j(\eta_q). \tag{D.2}$$

The latter requirement can also be expressed as

$$1 = \sum_j \left(\hat{M}_j(\eta_q) \sum_i \tilde{N}_i(\tilde{\xi}_{qj}) \right). \tag{D.3}$$

Note that, since $\tilde{\xi}_{qj}$ is invariant w.r.t. i and since the parent and normal bases fulfill partition of unity [40], it holds

$$\begin{aligned} \sum_i \tilde{N}_i(\tilde{\xi}_{qj}) &= 1, & j &= \text{const}, \\ \sum_j \hat{M}_j(\eta_q) &= 1. \end{aligned} \tag{D.4}$$

This finally leads to

$$1 = \sum_j \left(\hat{M}_j(\eta_q) \sum_i \tilde{N}_i(\tilde{\xi}_{qj}) \right) = \sum_j (\hat{M}_j(\eta_q) \cdot 1) = 1, \tag{D.5}$$

proving partition of unity for FLIGA B-Spline bases.

Appendix E. Weak Kronecker-delta property

FLIGA B-Spline bases constructed on the analysis domain Ω possess another important feature, namely weak Kronecker-delta property

$$B_{ij}(\mathbf{x}_q) = 0 \quad \forall \mathbf{x}_q \in \partial\Omega, (i, j) \notin \mathcal{B}_{bound}, \tag{E.1}$$

where \mathcal{B}_{bound} denotes the set of basis function index tuples associated to boundary control points and where again we drop index n .

Proof. Let us apply (37) and (35) such that proving

$$\tilde{N}_i(\tilde{\xi}_{qj}) \hat{M}_j(\eta_q) = 0 \quad \forall \xi_q \in \partial\Omega, (i, j) \notin \mathcal{B}_{bound}, \tag{E.2}$$

is equivalent to prove (E.1). The (floating) tensor-product structure allows index-based determination of interior control points by

$$(i, j) \notin \mathcal{B}_{bound} \Leftrightarrow (i \neq 1) \wedge (i \neq I) \wedge (j \neq 1) \wedge (j \neq J), \tag{E.3}$$

which is similar to classical IGA. Further, due to the rectangular structure of the parametric domain, we can guarantee

$$\begin{aligned} \xi_q \in \partial\Omega &\Leftrightarrow (\xi_q = a) \vee (\xi_q = b) \vee (\eta_q = \eta_1) \vee (\eta_q = \eta_{J+r+1}), \text{ while} \\ (\xi_q = a) &\Leftrightarrow (\tilde{\xi}_{qj} = \tilde{\xi}_1 \quad \forall j = 1, \dots, J) \\ (\xi_q = b) &\Leftrightarrow (\tilde{\xi}_{qj} = \tilde{\xi}_{I+r+1} \quad \forall j = 1, \dots, J). \end{aligned} \tag{E.4}$$

Evaluation of (E.2) can now be carried out separately for the four possible cases represented in (E.4). Let us assume that we are in the first case, i.e. $\xi_q = a$ or $\tilde{\xi}_{qj} = \tilde{\xi}_1 \quad \forall j = 1, \dots, J$ while due to (E.3) $i \neq 1$. This implies an assessment of weak Kronecker-delta property for the parent basis, which is a classical 1D B-Spline basis. Fulfillment was shown in [40]. Satisfying $i \neq 1$, we thus obtain $\tilde{N}_i(\tilde{\xi}_{qj}) = \tilde{N}_i(\tilde{\xi}_1) = 0$ resulting in $\hat{B}_{ij}(\xi_q) = 0$ for the case at hand, see (E.2). With this argumentation, we demonstrated weak Kronecker-delta property of FLIGA bases at the left parametric domain boundary $\xi = a$.

The other three cases of (E.4) correspond to the remaining three edges and can be handled similarly. As each case satisfies (E.2) (condition (E.3) fulfilled) there is no option left for which the weak Kronecker-delta property could be violated. Hence, all four parametric edges exhibit the weak Kronecker-delta property. Indeed, this property can be recognized from Fig. 5 to follow from the weak Kronecker-delta property of the characteristic and normal bases.

Appendix F. Boundary preservation

Another advantageous feature of FLIGA B-Spline bases is the independence of $\partial\Omega$ from the floating regulation points h_{ij} (time step index n skipped). That is, the physical boundary is not altered by the floating of characteristic basis functions, but only by the mechanical evolution of the problem.

Proof. To prove this, let us consider a point $\tilde{\xi}_{q1}^* \in \tilde{\Omega}$, where index $*$ indicates that this point is fixed in the parent domain. Note that this parent point is associated only to the floating map $j = 1$. Let us now map this point to the parametric domain:

$$\xi_q = \mathcal{G}_1(\tilde{\xi}_{q1}^*) = \sum_i h_{i1} \tilde{N}_i(\tilde{\xi}_{q1}^*). \tag{F.1}$$

Obviously, ξ_q depends on the floating regulation points. However, let us further consider the physical mapping of the parametric point $\xi_q = (\xi_q, \eta_1)^T$, where we associate our previously determined ξ_q to the bottom boundary $\eta = \eta_1$. With partition of unity and weak Kronecker-delta property of the normal basis, as well as (14), (35), (28) and (F.1), we have all required information to compute the associated point x_q on the physical boundary:

$$\begin{aligned} x_q &= \sum_i \sum_j c_{ij} \hat{B}_{ij}(\xi_q) = \sum_i \sum_j c_{ij} \hat{N}_{ij}(\xi_q) \hat{M}_j(\eta_1) = \sum_i c_{i1} \hat{N}_{i1}(\xi_q) \\ &= \sum_i c_{i1} \tilde{N}_i(\mathcal{G}_1^{-1}(\xi_q)) = \sum_i c_{i1} \tilde{N}_i(\tilde{\xi}_{q1}^*). \end{aligned} \tag{F.2}$$

Unlike the characteristic ξ_q , the physical x_q proves not to be dependent on h_{ij} . Recall that $\tilde{\xi}_{q1}^*$ is prescribed constant and therefore $\tilde{N}_i(\tilde{\xi}_{q1}^*)$ remains constant as well. Following this demonstration for each $\tilde{\xi}_{q1}^* \in \tilde{\Omega}$ while $\eta = \eta_1$, the related x_q cover the entire physical boundary associated to the bottom edge of the parametric domain. Dependency is only w.r.t. c_{ij} , hence, the physical shape is preserved under variation of h_{ij} . The same argumentation is valid for the upper edge of the parametric domain $\eta = \eta_{J+r+1}$.

Due to the weak Kronecker-delta property, the physical maps of the left and right edges of the parametric domain ($\xi = a$ and $\xi = b$) are fully determined by c_{ij} (with $i = 1$ and $i = I$, respectively) and the normal basis $\{\hat{M}_j\}_{j=1}^J$. The physical shape of these edges is naturally preserved under floating, due to the normal basis remaining static under variation of h_{ij} .

Appendix G. Newton–Raphson scheme for the determination of the parametric material point position

Here, we focus on how to map back a physical point \mathbf{x}_p to parametric space with geometry mapping \mathcal{F} as given in (14). For simplicity we omit superscript n obtaining

$$\xi_q = \mathcal{F}^{-1}(\mathbf{x}_q). \tag{G.1}$$

Analogous to the 1D case in (B.1), we apply a Newton–Raphson scheme on the nonlinear vector equation

$$\mathbf{R}(\xi) = \mathcal{F}(\xi) - \mathbf{x}_q, \tag{G.2}$$

for which we seek the root ξ_q such that $\mathbf{R}(\xi_q) = \mathbf{0}$. Therefore, the linearization of \mathcal{F} w.r.t. ξ is needed, which is known as Jacobian \mathbf{J} by (16).

Starting from an initial guess $\xi_q^{(r=0)}$ (obtained from the previous time step), we apply

$$\xi_q^{(r+1)} = \xi_q^{(r)} - \mathbf{J}(\xi_q^{(r)})^{-1} \mathbf{R}(\xi_q^{(r)}), \tag{G.3}$$

until $\|\mathbf{R}(\xi_q^{(r^*)})\| < \text{tol}$. We choose the same tolerance as for the determination of parent quadrature point positions ($\text{tol} = 1e - 10$) and then set $\xi_q \leftarrow \xi_q^{(r^*)}$.

At each step r of this scheme, we need to provide $\hat{B}_{ij}(\xi_q^{(r)})$ and its gradient $\nabla_{\xi} \hat{B}_{ij}(\xi_q^{(r)})$ where $\xi_q^{(r)} = (\xi_q^{(r)}, \eta_q^{(r)})^T$ is given, therefore we follow the basis function evaluation strategy proposed in Section 3.3.

Appendix H. Deviatoric Cauchy stress linearizations

$$\frac{\partial \tau_{s,ij}}{\partial q_{\alpha}} = \frac{\partial \tau_{OldB,ij}}{\partial q_{\alpha}} = 0, \quad \frac{\partial \tau_{s,ij}}{\partial d_{\beta k}} = \frac{\partial \tau_{OldB,ij}}{\partial d_{\beta k}} = \eta_s \left(\frac{\partial B_{\beta}}{\partial x_j} \delta_{ik} + \frac{\partial B_{\beta}}{\partial x_i} \delta_{jk} \right). \tag{H.1}$$

Appendix I. Tangent stiffness matrix for implementation

The final tangent stiffness matrix according to (A.1) can be implemented as:

$$\frac{\partial Q_{\alpha}}{\partial q_{\alpha}} = 0, \quad \frac{\partial Q_{\alpha}}{\partial d_{\beta k}} = \sum_q W_q \cdot \left(\frac{\partial B_{\beta}}{\partial x_k} \cdot A_{\alpha} \right) \Big|_{x=x_q}, \tag{I.1}$$

$$\frac{\partial S_{bi}}{\partial q_{\alpha}} = - \sum_q W_q \cdot \left(\frac{\partial B_b}{\partial x_i} \cdot A_{\alpha} \right) \Big|_{x=x_q}, \quad \frac{\partial S_{bi}}{\partial d_{\beta k}} = \eta_s \sum_q W_q \cdot \left[\left(\sum_j \frac{\partial B_{\beta}}{\partial x_j} \frac{\partial B_b}{\partial x_j} \right) \delta_{ik} + \frac{\partial B_{\beta}}{\partial x_i} \frac{\partial B_b}{\partial x_k} \right] \Big|_{x=x_q}. \tag{I.2}$$

Appendix J. Periodic bases

For certain problems, periodic boundary conditions in the characteristic direction are required. The parent B-Spline basis of order r is therefore constructed on a closed 1D parent domain $\tilde{\Omega}$ and whenever evaluated, its characteristic connectivities $\mathcal{I}_{qj}^n = \{i \in \{1, \dots, I\} : \tilde{N}_i(\tilde{\xi}_{qj}^n) \neq 0\}$ are derived accordingly. Construction of this basis occurs such that C^{r-1} continuity is preserved, see Fig. J.18.

Assume we have such a periodic parent basis $\{\tilde{N}_i(\tilde{\xi})\}_{i=1}^I$ and ascending positions of the floating regulation points $h_{ij}^n \in [a^n, b^n]$ for given j . (Due to the floating regulation point updates, $h_{ij}^n \notin [a^n, b^n]$ is possible and then we shift h_{ij}^n by a multiple of $b^n - a^n$ so as to enter this interval.) It is important to note that if we evaluate the floating regulation points $h_{ij}^{n,q}$ as connected to a parent quadrature point $\tilde{\xi}_{qj}^n$, we cannot set $h_{ij}^{n,q} = h_{ij}^n$ as this would cause severe spatial jumps of these points at the periodic boundary. Instead we choose $h_{ij}^{n,q} \in \{h_{ij}^n + \lambda_{ij}^{n,q} \cdot (b^n - a^n)\}$ with $\lambda_{ij}^{n,q} \in \{-1, 0, 1\}$ such that

$$0 < h_{(i+1)j}^{n,q} - h_{ij}^{n,q} < b^n - a^n, \tag{J.1}$$

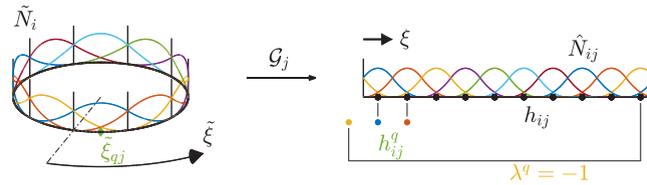


Fig. J.18. Schematic overview of 1D periodic B-Spline basis construction.

while

$$a^n \leq \mathcal{G}_j^n(\tilde{\xi}_{qj}^n) < b^n. \quad (\text{J.2})$$

We then follow the previously introduced B-Splines evaluation concepts for FLIGA.

References

- [1] T. Hughes, J. Cottrell, Y. Bazilevs, Isogeometric analysis: CAD, finite elements, NURBS, exact geometry and mesh refinement, *Comput. Methods Appl. Mech. Eng.* 194 (39) (2005) 4135–4195, <https://doi.org/10.1016/j.cma.2004.10.008>, URL <https://www.sciencedirect.com/science/article/pii/S0045782504005171>.
- [2] S. Lipton, J. Evans, Y. Bazilevs, T. Elguedj, T. Hughes, Robustness of isogeometric structural discretizations under severe mesh distortion, *Comput. Methods Appl. Mech. Eng.* 199 (5) (2010) 357–373, <https://doi.org/10.1016/j.cma.2009.01.022>, URL <https://www.sciencedirect.com/science/article/pii/S0045782509000346> Computational Geometry and Analysis.
- [3] Z. Wiecewski, The material point method in large strain engineering problems, *Comput. Methods Appl. Mech. Eng.* 193 (39) (2004) 4417–4438, <https://doi.org/10.1016/j.cma.2004.01.035>, URL <https://www.sciencedirect.com/science/article/pii/S0045782504002282> The Arbitrary Lagrangian-Eulerian Formulation.
- [4] D. Fußeder, *Isogeometric finite element methods for shape optimization*, 2015.
- [5] A. Shamanskiy, Mesh Deformation in the Context of Isogeometric Analysis, (Ph.D. thesis), Technische Universität Kaiserslautern, 2020, p. VI, 106, URL <http://nbn-resolving.de/urn:nbn:de:hbz:386-kluedo-60820>.
- [6] T. Belytschko, Y. Krongauz, D. Organ, M. Fleming, P. Krysl, Meshless methods: An overview and recent developments, *Comput. Methods Appl. Mech. Eng.* 139 (1) (1996) 3–47, [https://doi.org/10.1016/S0045-7825\(96\)01078-X](https://doi.org/10.1016/S0045-7825(96)01078-X), URL <https://www.sciencedirect.com/science/article/pii/S004578259601078X>.
- [7] J.-S. Chen, M. Hillman, S.-W. Chi, Meshfree methods: Progress made after 20 years, *J. Eng. Mech.* 143 (4) (2017) 04017001, [https://doi.org/10.1061/\(ASCE\)EM.1943-7889.0001176](https://doi.org/10.1061/(ASCE)EM.1943-7889.0001176).
- [8] Y. Xiao, H. Wu, An explicit coupled method of FEM and meshless particle method for simulating transient heat transfer process of friction stir welding, *Math. Probl. Eng.* (2020) <https://doi.org/10.1155/2020/2574127>.
- [9] S. Kumar, A. Vidyasagar, D.M. Kochmann, An assessment of numerical techniques to find energy-minimizing microstructures associated with nonconvex potentials, *Int. J. Num. Methods Eng.* 121 (7) (2020) 1595–1628, <https://doi.org/10.1002/nme.6280>, URL <https://onlinelibrary.wiley.com/doi/abs/10.1002/nme.6280> arXiv:<https://onlinelibrary.wiley.com/doi/pdf/10.1002/nme.6280>.
- [10] T. Belytschko, Y. Guo, W. Kam Liu, S. Ping Xiao, A unified stability analysis of meshless particle methods, *Int. J. Num. Methods Eng.* 48 (9) (2000) 1359–1400, [https://doi.org/10.1002/1097-0207\(20000730\)48:9<1359::AID-NME829>3.0.CO;2-U](https://doi.org/10.1002/1097-0207(20000730)48:9<1359::AID-NME829>3.0.CO;2-U).
- [11] S. Kumar, K. Danas, D.M. Kochmann, Enhanced local maximum-entropy approximation for stable meshfree simulations, *Comput. Methods Appl. Mech. Eng.* 344 (2019) 858–886, <https://doi.org/10.1016/j.cma.2018.10.030>, URL <https://www.sciencedirect.com/science/article/pii/S0045782518305346>.
- [12] C.M. Goh, P.M.F. Nielsen, M.P. Nash, A stabilised mixed meshfree method for incompressible media: Application to linear elasticity and Stokes flow, *Comput. Methods Appl. Mech. Eng.* 329 (2018) 575–598, <https://doi.org/10.1016/j.cma.2017.10.002>, URL <https://www.sciencedirect.com/science/article/pii/S0045782517306710>.
- [13] M. Arroyo, M. Ortiz, Local maximum-entropy approximation schemes: a seamless bridge between finite elements and meshfree methods, *Int. J. Num. Methods Eng.* 65 (13) (2006) 2167–2202, <https://doi.org/10.1002/nme.1534>, URL <https://onlinelibrary.wiley.com/doi/abs/10.1002/nme.1534> arXiv:<https://onlinelibrary.wiley.com/doi/pdf/10.1002/nme.1534>.
- [14] A. Rosolen, M. Arroyo, Blending isogeometric analysis and local maximum entropy meshfree approximants, *Comput. Methods Appl. Mech. Eng.* 264 (2013) 95–107, <https://doi.org/10.1016/j.cma.2013.05.015>, URL <https://www.sciencedirect.com/science/article/pii/S0045782513001321>.
- [15] D. Millán, N. Sukumar, M. Arroyo, Cell-based maximum-entropy approximants, *Comput. Methods Appl. Mech. Eng.* 284 (2015) 712–731, <https://doi.org/10.1016/j.cma.2014.10.012>, URL <https://www.sciencedirect.com/science/article/pii/S004578251400382X> Isogeometric Analysis Special Issue.
- [16] R. Cardoso, J. Cesar de Sa, Blending moving least squares techniques with NURBS basis functions for nonlinear isogeometric analysis, *Comput. Mech.* 53 (2014) 1327–1340, <https://doi.org/10.1007/s00466-014-0977-5>, URL <https://link.springer.com/article/10.1007%2Fs00466-014-0977-5>.

- [17] F. Fathi, L. Chen, R. de Borst, X-IGALME: Isogeometric analysis extended with local maximum entropy for fracture analysis, *Int. J. Num. Methods Eng.* (2021) <https://doi.org/10.1002/nme.6784>, URL <https://onlinelibrary.wiley.com/doi/abs/10.1002/nme.6784> arXiv:<https://onlinelibrary.wiley.com/doi/pdf/10.1002/nme.6784>.
- [18] I. Akkerman, Y. Bazilevs, C. Kees, M. Farthing, Isogeometric analysis of free-surface flow, *J. Comput. Phys.* 230 (11) (2011) 4137–4152, <https://doi.org/10.1016/j.jcp.2010.11.044>, URL <https://www.sciencedirect.com/science/article/pii/S0021999110006595> Special issue High Order Methods for CFD Problems.
- [19] J. Evans, T. Hughes, Isogeometric divergence-conforming B-Splines for the steady navier–stokes equations, *Math. Model. Methods Appl. Sci.* 23 (08) (2013) 1421–1478, <https://doi.org/10.1142/S0218202513500139>.
- [20] B.S. Hosseini, M. Möller, S. Turek, Isogeometric analysis of the Navier–Stokes equations with Taylor–Hood B-spline elements, *Appl. Math. Comput.* 267 (2015) 264–281, <https://doi.org/10.1016/j.amc.2015.03.104>, URL <https://www.sciencedirect.com/science/article/pii/S009630031500418X> The Fourth European Seminar on Computing (ESCO 2014).
- [21] B.S. Hosseini, S. Turek, M. Möller, C. Palmes, Isogeometric analysis of the Navier–Stokes–Cahn–Hilliard equations with application to incompressible two-phase flows, *J. Comput. Phys.* 348 (2017) 171–194, <https://doi.org/10.1016/j.jcp.2017.07.029>, URL <https://www.sciencedirect.com/science/article/pii/S0021999117305375>.
- [22] P.N. Nielsen, A.R. Gersborg, J. Gravesen, N.L. Pedersen, Discretizations in isogeometric analysis of Navier–Stokes flow, *Comput. Methods Appl. Mech. Eng.* 200 (45) (2011) 3242–3253, <https://doi.org/10.1016/j.cma.2011.06.007>, URL <https://www.sciencedirect.com/science/article/pii/S0045782511002258>.
- [23] J. Reinold, V. Nerella, V. Mechtcherine, G. Meschke, Extrusion process simulation and layer shape prediction during 3D-concrete-printing using the particle finite element method, 2020, <https://doi.org/10.20944/preprints202007.0715.v1>, Preprints.
- [24] Z. Ouyang, E. Bertevas, D. Wang, B. Khoo, J. Férec, G. Ausias, N. Phan-Thien, A smoothed particle hydrodynamics study of a non-isothermal and thermally anisotropic fused deposition modeling process for a fiber-filled composite, *Phys. Fluids* 32 (5) (2020) 053106, <https://doi.org/10.1063/5.0004527>.
- [25] R. Comminal, M. Serdeczny, D. Pedersen, J. Spangenberg, Numerical modeling of the strand deposition flow in extrusion-based additive manufacturing, *Addit. Manuf.* 20 (2018) <https://doi.org/10.1016/j.addma.2017.12.013>.
- [26] L. De Lorenzis, P. Wriggers, T.J. Hughes, Isogeometric contact: a review, *GAMM-Mitt.* 37 (1) (2014) 85–123, <https://doi.org/10.1002/gamm.201410005>, URL <https://onlinelibrary.wiley.com/doi/abs/10.1002/gamm.201410005> arXiv:<https://onlinelibrary.wiley.com/doi/pdf/10.1002/gamm.201410005>.
- [27] Y. Bazilevs, V. Calo, J. Cottrell, J. Evans, T. Hughes, S. Lipton, M. Scott, T. Sederberg, Isogeometric analysis using T-splines, *Comput. Methods Appl. Mech. Eng.* 199 (5) (2010) 229–263, <https://doi.org/10.1016/j.cma.2009.02.036>, URL <https://www.sciencedirect.com/science/article/pii/S0045782509000875> Computational Geometry and Analysis.
- [28] J. Cottrell, T. Hughes, Y. Bazilevs, NURBS as a basis for analysis: Linear problems, in: *Isogeometric Analysis*, John Wiley & Sons, Ltd, 2009, pp. 69–107, <https://doi.org/10.1002/9780470749081.ch3>, URL <https://onlinelibrary.wiley.com/doi/abs/10.1002/9780470749081.ch3> arXiv:<https://onlinelibrary.wiley.com/doi/pdf/10.1002/9780470749081.ch3>.
- [29] T. Rübberg, F. Cirak, Subdivision-stabilised immersed b-spline finite elements for moving boundary flows, *Comput. Methods Appl. Mech. Eng.* 209–212 (2012) 266–283, <https://doi.org/10.1016/j.cma.2011.10.007>, URL <https://www.sciencedirect.com/science/article/pii/S0045782511003197>.
- [30] T. Hughes, A. Reali, G. Sangalli, Efficient quadrature for NURBS-based isogeometric analysis, *Comput. Methods Appl. Mech. Eng.* 199 (5) (2010) 301–313, <https://doi.org/10.1016/j.cma.2008.12.004>, URL <https://www.sciencedirect.com/science/article/pii/S0045782508004295> Computational Geometry and Analysis.
- [31] F. Fahrendorf, L. De Lorenzis, H. Gomez, Reduced integration at superconvergent points in isogeometric analysis, *Comput. Methods Appl. Mech. Eng.* 328 (2018) 390–410, <https://doi.org/10.1016/j.cma.2017.08.028>, URL <https://www.sciencedirect.com/science/article/pii/S0045782517303717>.
- [32] F. Auricchio, L.B. Da Veiga, T.J.R. Hughes, A. Reali, G. Sangalli, Isogeometric collocation Methods, *Math. Model. Methods Appl. Sci.* 20 (11) (2010) 2075–2107, <https://doi.org/10.1142/S0218202510004878>.
- [33] G. Fougerson, Contribution to the improvement of meshless methods applied to continuum mechanics, (Ph.D. thesis), Université Paris-Saclay, 2018, URL <https://jeannicod.ccsd.cnrs.fr/STAR/tel-01968070v1>.
- [34] S. Kumar, A.D. Tutcuoglu, Y. Hollenweger, D. Kochmann, A meshless multiscale approach to modeling severe plastic deformation of metals: Application to ECAE of pure copper, *Comput. Mater. Sci.* 173 (2020) 109329, <https://doi.org/10.1016/j.commatsci.2019.109329>, URL <https://www.sciencedirect.com/science/article/pii/S0927025619306287>.
- [35] Y. Yue, B. Smith, C. Batty, C. Zheng, E. Grinspun, Continuum foam: A material point method for shear-dependent flows, *ACM Trans. Graph.* 34 (5) (2015) <https://doi.org/10.1145/2751541>.
- [36] N. Phan-Thien, *Understanding Viscoelasticity*, second ed., Springer, Heidelberg New York Dordrecht London, 2013.
- [37] Y. Bai, Study of the viscoelastic instability in Taylor–Couette system as an analog of the magnetorotational instability, (Ph.D. thesis), L’université du Havre, 2015, URL <https://www.theses.fr/2015LEHA0015.pdf>.
- [38] W.-H. Zhang, J. Li, Q. Wang, Y. Ma, H.-N. Zhang, B. Yu, F. Li, Comparative study on numerical performances of log-conformation representation and standard conformation representation in the simulation of viscoelastic fluid turbulent drag-reducing channel flow, *Phys. Fluids* 33 (2) (2021) 023101, <https://doi.org/10.1063/5.0034589>.
- [39] N. Turner, R. Strong, S. Gold, A review of melt extrusion additive manufacturing processes: I. Process design and modeling, *Rapid Prototyp. J.* (2014) 192–204, <https://doi.org/10.1108/RPJ-01-2013-0012>.
- [40] L. Piegl, W. Tiller, *The NURBS Book*, second ed., Springer-Verlag, New York, NY, USA, 1996.