Enhanced Floating Isogeometric Analysis

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

DOI
10.1016/j.cma.2023.116346

Publication date
2023

Document Version
Final published version

Published in
Computer Methods in Applied Mechanics and Engineering

Available online at www.sciencedirect.com
ScienceDirect
Comput. Methods Appl. Mech. Engrg. 417 (2023) 116346
Computer methods in applied mechanics and engineering
www.elsevier.com/locate/cma

Enhanced 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

Available online 26 August 2023

Abstract

The numerical simulation of additive manufacturing techniques promises the acceleration of costly experimental procedures to identify suitable process parameters. We recently proposed Floating Isogeometric Analysis (FLIGA), a new computational solid mechanics approach, which is mesh distortion-free in one characteristic spatial direction. FLIGA emanates from Isogeometric Analysis and its key novel aspect is the concept of deformation-dependent "floating" of individual B-spline basis functions along one parametric axis of the mesh. Our previous work showed that FLIGA not only overcomes the problem of mesh distortion associated to this direction, but is also ideally compatible with material point integration and enjoys a stability similar to that of conventional Lagrangian mesh-based methods. These features make the method applicable to the simulation of large deformation problems with history-dependent constitutive behavior, such as additive manufacturing based on polymer extrusion. In this work, we enhance the first version of FLIGA by *(i)* a novel quadrature scheme which further improves the robustness against mesh distortion, *(ii)* a procedure to automatically regulate floating of the basis functions (as opposed to the manual procedure of the first version), and *(iii)* an adaptive refinement strategy. We demonstrate the performance of enhanced FLIGA on relevant numerical examples including a selection of viscoelastic extrusion problems.

Keywords: Floating Isogeometric Analysis; Large deformations; Viscoelasticity; Adaptive refinement; Mesh distortion

Dedication

We dedicate this paper to Tom Hughes, a giant of computational mechanics and a continuous source of inspiration, for his incredible lifetime achievements. He shaped the computational mechanics field as well as the scientific path of generations of scholars, who owe him a great debt of gratitude. The senior author of this paper is one of them.

1. Introduction

Additive manufacturing (AM) is a wide-ranging class of modern fabrication technologies in which material is successively added to build the final part, see [1] for an introduction to the field. Among others, a prominent concept for the controlled addition of material is polymer extrusion. Polymer extrusion has long been a state-of-the-art technique also for the manufacturing of plastic profiles, where molten polymer is forced through a die to create a

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

https://doi.org/10.1016/j.cma.2023.116346

continuous strand. In its application to AM, on a smaller scale, geometrically simple strands are deposited under movement of a heated robotic nozzle, thereby building products by adding strand to strand, and then layer to layer. The technique is especially attractive as it facilitates the customized fabrication of parts with complex topology and/or multi-material arrangement. However, the quality of the solidified end product is heavily dependent on the suitable choice of the process parameters, which is often specific to the individual printing task. This motivates the need for process simulations, to avoid costly and time-consuming trial-and-error experimental procedures.

The Lagrangian and Eulerian viewpoints of motion in continuum mechanics are usually preferred to describe solid and fluid behavior, respectively. When it comes to the numerical approximation, the finite element method (FEM) and Isogeometric Analysis (IGA) [2] are often combined with the Lagrangian viewpoint for the discretization of solid mechanics problems. Lagrangian FEM is highly sensitive to mesh distortion for very large deformations and therefore inapplicable to the simulation of fluid behavior. While showing a higher degree of robustness [3], standard (Lagrangian) IGA also suffers from the same issue. Eulerian formulations of FEM or IGA are inherently free from mesh distortion, but typically come along with additional efforts since a stabilization of the advective terms as well as a special treatment of moving boundaries are required, see e.g. [4]. A challenge arises for problems involving both very large (fluid-like) deformations *and* solid behavior. Polymer extrusion is one of such cases, as the viscoelastic deformation response of most polymers lies somewhere in between fluid and solid behavior. In our previous paper [5] we discussed the available computational techniques to handle such cases; as follows, we only mention them very briefly along with a few areas requiring further improvements. The re-generation of undistorted meshes, also known as remeshing, is computationally expensive and its success is highly dependent on suitable techniques to control the errors of history data projection [6]. Adoption of remeshing in IGA is complex and a subject of ongoing research [7–9]. Though avoiding costly mesh generations, most meshless methods [10–12] are still considered less efficient than their mesh-based competitors [13–15] and stabilization plays an important role in case of extreme deformations [16,17]. In many cases, accurate quadrature of meshless basis functions is challenging; moreover, for incompressible material behavior which is relevant for polymers, the construction of mixed discretizations to avoid locking is not trivial [18]. The well-founded theory of mesh-based analysis and the flexibility of meshless methods have motivated some hybrid approaches, aiming to benefit from the complementing advantages of both settings, see e.g. [19–23]. Finally, mesh-based approaches combining Lagrangian and Eulerian concepts have also been devised, e.g. the Arbitrary Lagrangian Eulerian method (ALE) [24,25]. Unfortunately, the increase in methodical complexity remains again a limitation to its further establishment [26].

Beyond the original goal to circumvent the expensive mesh generation in FEM, the adoption of basis functions stemming from Computer Aided Design (CAD) in IGA [2] proved beneficial for the analysis itself in many applications. Mostly due to the higher and tunable continuity of isogeometric basis functions at element boundaries, IGA was shown to lead to advantages in e.g., structural dynamics [27], the solution of higher-order partial differential equations [28], and contact mechanics [29], among many others. As mentioned, like FEM, Lagrangian IGA also suffers from mesh distortion; the basis functions are constructed first on a reference domain and a distorted mapping to the physical space in case of large deformations negatively affects the numerical solution [3]. Yet, there is one significant difference between FEM and IGA. While the reference (parent) domain in FEM collects a few basis functions which are mapped many times to different *elements*, the reference (parametric) domain in IGA collects many basis functions and is mapped only a few times to different *patches*. In IGA, a patch is again subdivided into (so-called Bézier) elements, so that the construction of the basis functions can be alternatively viewed as *local* or *elementwise* (like in FEM) or *global* or *patchwise* (an alternative typical of IGA). While the first perspective was adopted in the isogeometric counterpart of FEM remeshing techniques [7,8], the alternative one can be considered the underlying viewpoint of the new strategy to overcome mesh distortion that we recently proposed and denoted as Floating IGA (FLIGA) [5].

FLIGA approaches the numerical solution of a Lagrangian model from a global (patchwise) perspective. An important starting assumption is that, in the problem at hand, extreme deformations occur along one (possibly curved) physical axis, which can be mapped to one corresponding parametric direction, denoted as the *characteristic direction*. The key idea is to comprise all Bézier elements along the characteristic direction in a so-called *characteristic element*, as illustrated in Fig. 1 on the exemplary geometry of a material extrusion problem, while preserving local support of its basis functions. The abolition of Bézier element boundaries in such direction confers a meshless character to the basis functions limited to this direction, while the mesh is preserved in the other parametric direction (also denoted as *normal direction*). The problem of mesh distortion along the characteristic elements is

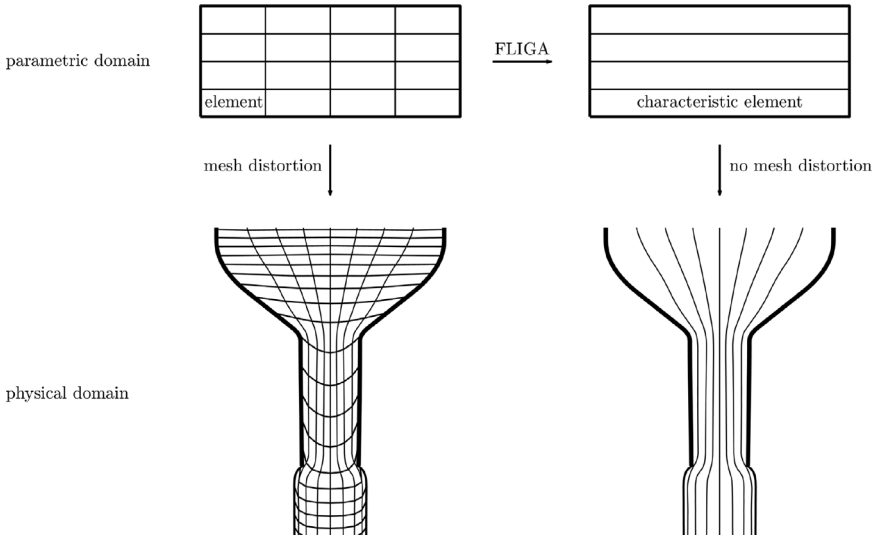


Fig. 1. Element concepts in IGA and FLIGA.

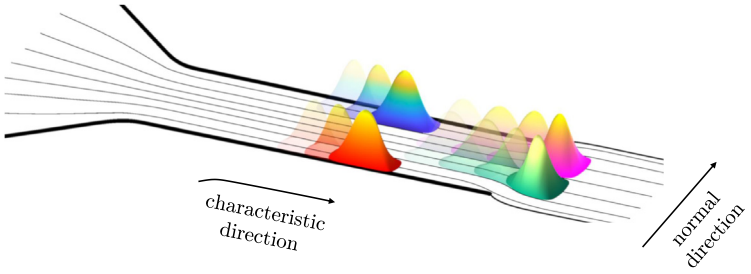


Fig. 2. Exemplary "floating" movement of four basis functions along the characteristic elements in the physical space.

then solved by the mere update of the corresponding univariate B-Spline basis functions. The movement of basis functions that we denote as *floating* is schematically illustrated in Fig. 2 for the extrusion example and provides the basis for a new class of boundary-fitted spline approximation spaces, which are distortion-free along one parametric direction.

In our recent paper [5], we developed FLIGA following the isoparametric concept and demonstrated its performance on the Taylor–Couette flow problem and on a polymer extrusion problem, both featuring viscoelastic material behavior and extreme deformations. In this paper, we aim at improving upon three aspects of the initial formulation. Firstly, in [5] we could solve the Taylor–Couette flow problem up to over five full turns of the cylinder, after which we observed a rapid increase of the numerical error. We suspected the reason to be related to the quadrature scheme, i.e. to the accumulation of quadrature error. Secondly, floating of the basis functions was regulated by the movement of the so-called floating regulation points, which was connected to the control point displacements through a manually designed level-set function. This manual design was based on a qualitatively reasonable *a priori* guess about the displacements, which was not too difficult to obtain for the considered problems but may easily become impractical or even impossible in more general cases. Finally, while FLIGA solved the issue of shear-driven distortion, the floating of one basis function was still associated to its elongational deformation. To prevent issues stemming from an excessive dilation for severe floating, we used very fine meshes, which led to a high computational cost.

In this paper, we address these three points and propose three corresponding enhancements. For the first problem, a new numerical quadrature concept is introduced to improve the numerical stability. The second issue is solved via an automated procedure for floating regulation based on the *actual* (and not on an *a priori* guessed) deformation. To

address the third issue, we devise a simple adaptive refinement strategy for FLIGA, which reduces the computational cost. A number of numerical examples demonstrate the effect of these enhancements.

The remainder of this paper is organized as follows: In Section 2, after reviewing the concept of floating basis functions, we illustrate the three proposed enhancements. In Section 3 we focus on the Lagrangian modeling of viscoelasticity and its standard IGA discretization, and eventually propose the enhanced version of FLIGA incorporating the new approaches of the previous section. Finally, Section 4 discusses the numerical examples and conclusions are drawn in Section 5.

2. Floating B-Splines

At the heart of FLIGA are *floating* B-Splines, which we proposed in [5] as a generalization of conventional B-Splines [30,31]. In this section, we illustrate the design of floating B-Splines in a further generalized fashion compared to Hille et al. [5] and we introduce some new operations on them. These concepts are later applied to analysis in Section 3.3, which outlines the enhanced version of FLIGA.

2.1. Basis construction

A univariate standard B-Spline basis is constructed on a parametric domain with coordinate ξ, here $\hat{\Omega}_\xi = [0, 1]$, by means of the Cox-de Boor recursion formula [32,33]

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

where we define $\frac{0}{0} = 0$. The non-decreasing sequence of real numbers denoted as knots, ξ_i, constitutes the knot vector Ξ and if the first (last) $p + 1$ knots are all 0 (1) we call the knot vector "open". Here p is the polynomial order of the B-Spline basis, which tunes its continuity at the knots. At a unique (not repeated) knot, continuity is $\mathcal{C}^{p-1}$; each repetition of a knot decreases the order of continuity at the knot by one; between the knots, continuity is $\mathcal{C}^\infty$. In this paper, we always use open knot vectors and unique inner knots. Each knot span is also denoted as an *element*.

Let us now define two different types of bivariate B-spline bases (trivariate bases are not treated in this paper). To this end, a patch is spanned on a square parametric domain with coordinates ξ and η, here $\hat{\Omega} = \hat{\Omega}_\xi \times \hat{\Omega}_\eta = [0, 1] \times [0, 1]$. The first type of bivariate B-Spline basis has the classical tensor product structure (the one used in IGA); this is obtained by combining one univariate basis in η with only *one* univariate basis in ξ, i.e.

$$\hat{B}^{TP}_{ij}(\xi, \eta) = \hat{N}_i(\xi) \hat{M}_j(\eta) \qquad i = 1, \ldots, I;\ j = 1, \ldots, J, \tag{2}$$

with I and J as the total number of basis functions in the two parametric directions. The basis $\left\{\hat{M}_j\right\}_{j=1,\ldots,J}$ is constructed similarly to basis $\left\{\hat{N}_i\right\}_{i=1,\ldots,I}$ knowing the order q and the knot vector H along coordinate η. The second type of bivariate B-Spline basis has the *floating* tensor product structure (the one used in FLIGA); here we associate a different univariate basis in ξ to each j

$$\hat{B}_{ij}(\xi, \eta) = \hat{N}_{ij}(\xi) \hat{M}_j(\eta) \qquad i = 1, \ldots, I_j;\ j = 1, \ldots, J. \tag{3}$$

Note that no summation over repeated indices is implied in Eq. (3) (throughout the paper, to avoid any confusion we will write all summation symbols explicitly). We denote $\left\{\hat{N}_{ij}\right\}_{i=1,\ldots,I_j}$ for each fixed $j = 1, \ldots, J$ as *characteristic basis* and $\left\{\hat{M}_j\right\}_{j=1,\ldots,J}$ as *normal basis*. This floating tensor product structure allows us to construct bases as depicted in Fig. 3. In the figure, starting from a classical bivariate tensor product basis we only modify the characteristic bases for $j = A, B$, i.e. $\left\{\hat{N}_{iA}\right\}_{i=1,\ldots,I_A}$ and $\left\{\hat{N}_{iB}\right\}_{i=1,\ldots,I_B}$ for better clarity (in practice, we will typically use different characteristic bases for all j). Note that, unlike in our first FLIGA paper [5], here the number of basis functions along the characteristic direction, I_j, can vary for different j, a property crucial for

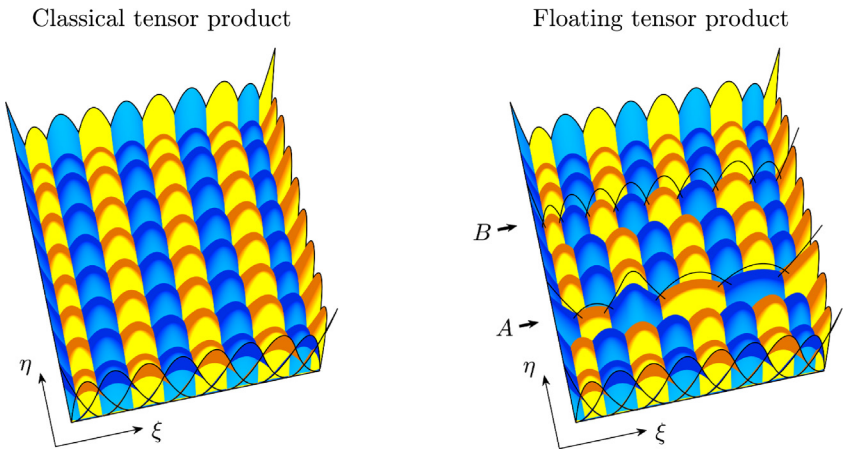


Fig. 3. Examples of the two tensor product structures in parametric space.

local refinement (see Section 2.4.3). E.g. in Fig. 3 it is $I_A = 6$ and $I_B = 10$. Fig. 3 also shows that the element boundaries of the characteristic bases are not required to stay aligned for the floating tensor product structure. In fact, the element concept has been relaxed and element division is only kept in the η direction, associated to the normal basis. This corresponds to the concept of characteristic elements schematized in Fig. 1.

The next question which arises is how to construct the different characteristic bases in the floating tensor product structure. To this end, for each $j = 1, \dots, J$ we construct a so-called *parent* basis $\left\{\tilde{N}_{ij}\left(\tilde{\xi}\right)\right\}_{i=1,\dots,I_j}$ with order p from parent knot vector $\tilde{\Xi}_j$ on the parent domain $\tilde{\Omega} = [0, 1]$ along coordinate $\tilde{\xi}$. Then, for each $j = 1, \dots, J$ we define a scalar mapping $\mathcal{G}_j : \tilde{\Omega} \to \hat{\Omega}_\xi$, denoted as *floating map*, which transforms a given $\tilde{\xi}$ to

$$\xi_j\left(\bullet; \mathcal{H}\right) = \mathcal{G}_j\left(\tilde{\xi}; \mathcal{H}\right) = \sum_{i=1}^{I_j} h_{ij} \tilde{N}_{ij}\left(\tilde{\xi}\right), \tag{4}$$

where the scalar linear coefficients h_{ij} are termed *floating regulation points* and $\mathcal{H}$ denotes the set in which they are grouped:

$$\mathcal{H} = \left\{h_{ij}\right\}_{i=1,\dots,I_j;\, j=1,\dots,J}. \tag{5}$$

The inverse mapping $\mathcal{G}_j^{-1} : \hat{\Omega}_\xi \to \tilde{\Omega}$ transforms back a given ξ to

$$\tilde{\xi}_j\left(\bullet; \mathcal{H}\right) = \mathcal{G}_j^{-1}\left(\xi; \mathcal{H}\right). \tag{6}$$

Next, the characteristic basis functions are derived as the push-forwards of the respective parent basis functions

$$\hat{N}_{ij}\left(\xi; \mathcal{H}\right) = \tilde{N}_{ij}\left(\mathcal{G}_j^{-1}\left(\xi; \mathcal{H}\right)\right) = \tilde{N}_{ij}\left(\tilde{\xi}_j\left(\bullet; \mathcal{H}\right)\right), \tag{7}$$

which corresponds to a pull back of the characteristic coordinate ξ to the parent space, see Fig. 4, where $i = 5$ and $i^* = 8$. In this sense, the modification of the floating regulation points allows to control the decoupled floating of the characteristic bases. Let us highlight that, by construction of a parent basis function, its evaluation $\tilde{N}_{ij}\left(\tilde{\xi}\right)$ only indirectly depends on $\mathcal{H}$ in case its argument $\tilde{\xi}$ does. In contrast, a characteristic basis function evaluation $\hat{N}_{ij}\left(\xi; \mathcal{H}\right)$ explicitly involves such parameter dependency even for a constant argument ξ. With the above parent concept, Eq. (3) now reads

$$\hat{B}_{ij}\left(\boldsymbol{\xi}; \mathcal{H}\right) = \hat{N}_{ij}\left(\xi; \mathcal{H}\right) \hat{M}_j\left(\eta\right) \qquad i = 1, \dots, I_j;\; j = 1, \dots, J, \tag{8}$$

where $\boldsymbol{\xi} = (\xi, \eta)^T$. In the following equations we continue to indicate if a quantity or function has a parameter dependency on the floating regulation points.

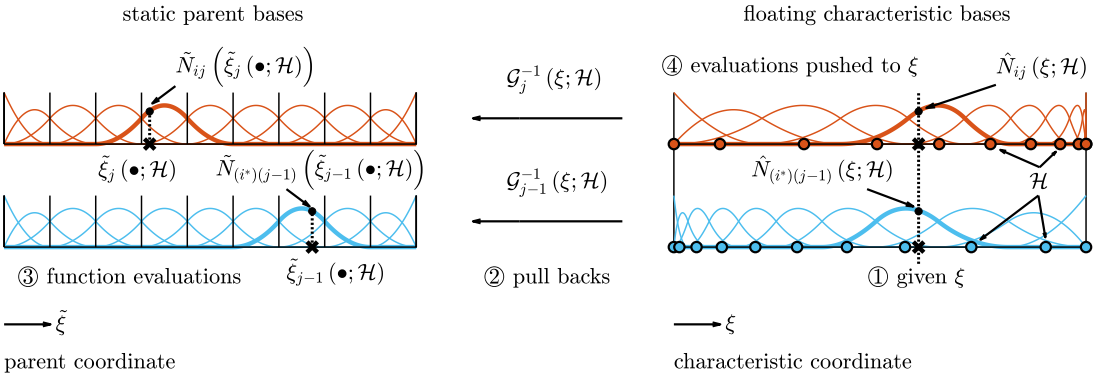


Fig. 4. Function computations for two characteristic bases given ξ, $\mathcal{H}$, and the parent bases. Numbers in circles indicate four consecutive operational steps.

2.2. Physical map

Standard B-Spline free-form surfaces are constructed from a linear combination of a function basis and so-called control points, mapping the parametric to the physical space, $\mathcal{F}^{TP}:\hat{\Omega}\rightarrow\Omega$

$$\boldsymbol{x}=\mathcal{F}^{TP}(\boldsymbol{\xi})=\sum_{j=1}^{J}\sum_{i=1}^{I}\boldsymbol{c}_{ij}\hat{B}_{ij}^{TP}(\boldsymbol{\xi})\,, \tag{9}$$

where $\boldsymbol{c}_{ij}$ are the position vectors of the control points. The basis functions can be pushed-forward to a given $\boldsymbol{x}$ in physical space by

$$B_{ij}^{TP}(\boldsymbol{x})=\hat{B}_{ij}^{TP}\left(\left(\mathcal{F}^{TP}\right)^{-1}(\boldsymbol{x})\right)=\hat{B}_{ij}^{TP}(\boldsymbol{\xi})\,. \tag{10}$$

Similarly, for a floating B-Spline basis we employ the following mapping, $\mathcal{F}:\hat{\Omega}\rightarrow\Omega$, from the parametric to the physical space

$$\boldsymbol{x}(\bullet;\mathcal{H})=\mathcal{F}(\boldsymbol{\xi};\mathcal{H})=\sum_{j=1}^{J}\sum_{i=1}^{I_j}\boldsymbol{c}_{ij}\hat{B}_{ij}(\boldsymbol{\xi};\mathcal{H})\,. \tag{11}$$

The floating basis functions are pushed-forward to a given $\boldsymbol{x}$ in physical space by

$$B_{ij}(\boldsymbol{x};\mathcal{H})=\hat{B}_{ij}\left(\mathcal{F}^{-1}(\boldsymbol{x};\mathcal{H});\mathcal{H}\right)=\hat{B}_{ij}(\boldsymbol{\xi}(\bullet;\mathcal{H});\mathcal{H})\,. \tag{12}$$

Note that in general, continuity in the mapping is limited at $\boldsymbol{\xi}$ if just *one* of all supported basis functions has limited continuity. However, continuity of the map is always at least $\mathcal{C}^{p-1}$ when traveling along lines of constant η, and $\mathcal{C}^{q-1}$ when traveling along lines of constant ξ (recall that we assumed that inner knots are not repeated).

We summarize the entire construction concept of classical and floating tensor product B-Splines in Figs. 5 and 6, respectively, where two exemplary bivariate bases with $p=2$, $q=1$ are mapped to a free-form geometry in physical space each. The colored circles in the physical domain indicate the control points (in Fig. 6, the colored circles in the parametric domain denote the floating regulation points) connected to the control point $\mathbf{c}_{ij}$ (floating regulation point h_{ij}), for $i=4$ and $j=3$, which is highlighted with a star symbol. For the floating tensor product, different j indices are referred to by a different color and the color legend of Fig. 6 equally applies to the following figures on this tensor product. Until now, the exemplary surfaces and their parametrizations are the same for both concepts, as all floating regulation points are aligned. However, the floating B-Spline representation allows for a richer modification as will be seen next.

2.3. Effect of floating regulation

When using classical B-Spline representations for Lagrangian large deformation analysis, the extreme displacement of control points can result in a distorted mesh. Fig. 7 visualizes again the tensor product mesh from

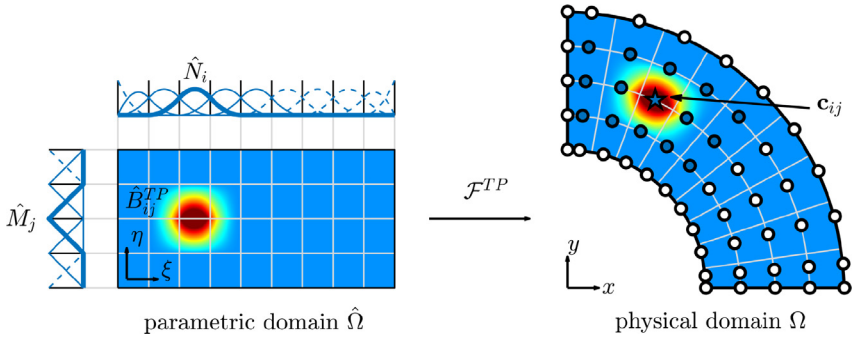


Fig. 5. Summary of the construction procedure for classical B-Spline surfaces.

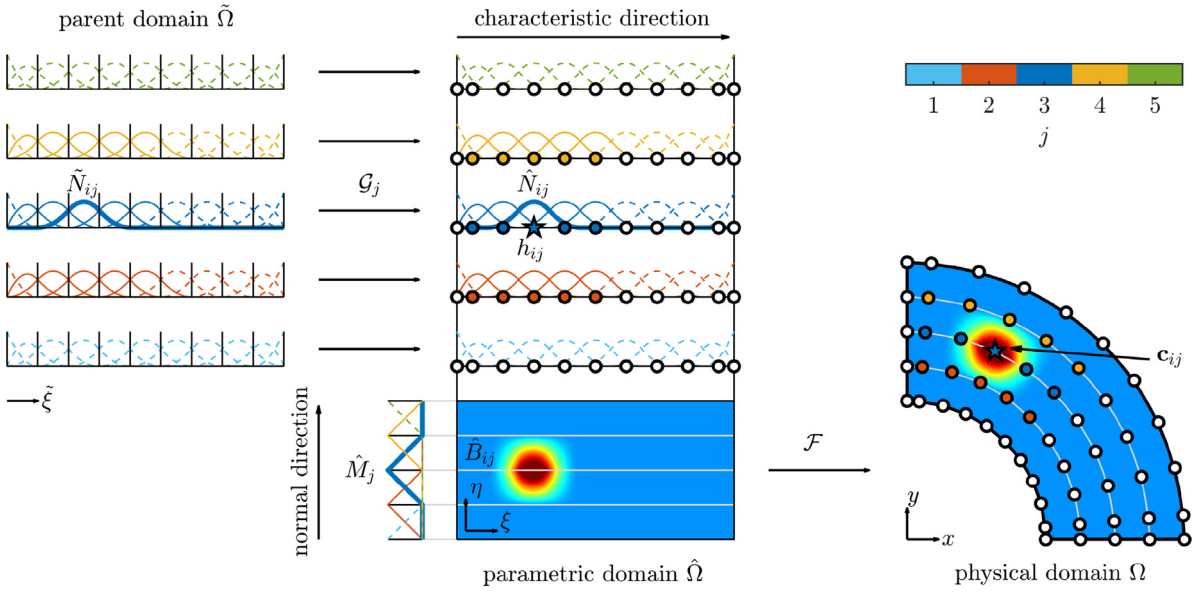


Fig. 6. Summary of the construction procedure for floating B-Spline surfaces. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

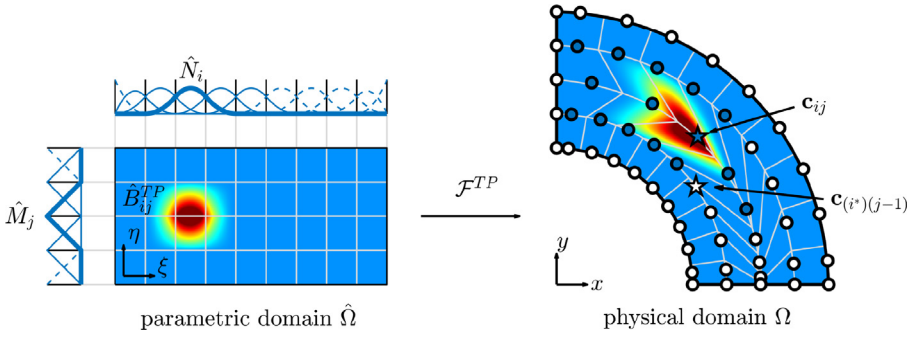


Fig. 7. Mesh distortion in IGA.

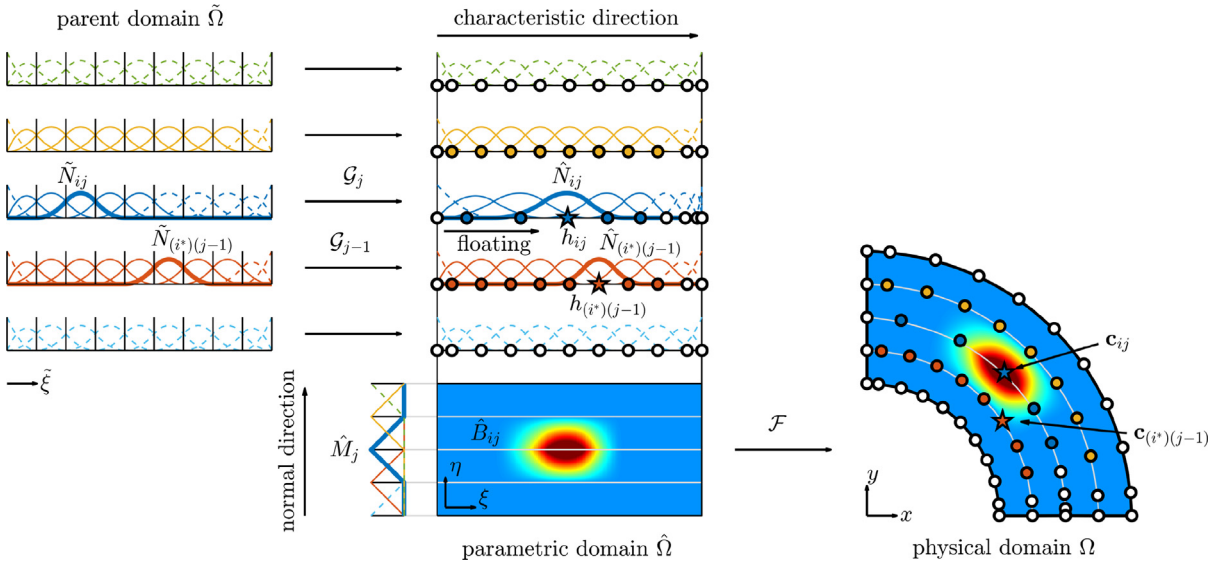


Fig. 8. Overcoming shear mesh distortion in FLIGA.

Fig. 5, but for a deformed control point set, resulting in significant mesh distortion. Note that here control point connectivities are rigid; e.g., the control point $\boldsymbol{c}_{ij}$ still connects to the same (now displaced) control points as in the previous configuration. Even though $\boldsymbol{c}_{ij}$ has moved close to $\boldsymbol{c}_{(i^*)(j-1)}$ ($i^* = 7$, white star symbol), these two remain unconnected.

For floating B-Splines, two control points $\boldsymbol{c}_{ij}$ and $\boldsymbol{c}_{(i^*)(j-1)}$ with a given $i^* \in \{1, 2, \ldots, I_{j-1}\}$ are connected if the associated $\hat{N}_{ij}$ and $\hat{N}_{(i^*)(j-1)}$ are both supported within a sub-range of $\hat{\Omega}_\xi$. Hence, changing the floating maps (which is possible by tuning the floating regulation points, recall Eq. (4)) allows to unconnect control points that are far from each other in physical space and/or connect those which are close neighbors. Such adjustment reparametrizes the physical domain and reduces the shear distortion in the physical mapping $\mathcal{F}$. As a consequence we observe the "floating" movement of the basis functions as initially illustrated in Fig. 2. In Fig. 8 we demonstrate this effect on the same displaced control point set as in Fig. 7, but with a suitable deformation-dependent choice of the floating maps, resulting in an update of connectivity. E.g., the control points $\boldsymbol{c}_{ij}$ and $\boldsymbol{c}_{(i^*)(j-1)}$ ($i^* = 7$, red star symbol) are now connected. The question of how to automatically obtain robust choices of floating regulation points is addressed in Section 2.4.2.

2.4. Enhanced features

The construction of the basis functions has been presented with a level of generality that now allows us to introduce three new features of floating B-Splines with respect to Hille et al. [5]:

- a modified quadrature (Section 2.4.1);
- the automated choice of floating regulation points (Section 2.4.2);
- local h-refinement (Section 2.4.3).

At this point, we also add one restriction, namely, the choice of the polynomial order of the *normal* basis as $q = 1$. This choice may be surprising, but will be shown in the next subsection to lead to important advantages for numerical quadrature in case of history-dependent material behavior. The B-Spline order p of the *characteristic* bases remains tunable and hence we preserve the smooth floating character along the characteristic direction, see Fig. 8.

2.4.1. Numerical quadrature

Conventional quadrature in IGA makes use of the tensor product structure and combines quadrature coordinates of the univariate bases to obtain quadrature points in parametric space. In the Lagrangian setting, the quadrature points

are mapped to consistent material points in undeformed and deformed physical configurations. The renowned Gauss–Legendre rule was the first quadrature coordinate stencil suggested in IGA [2] and, owing to its high robustness and accuracy for regular integrands, established itself as one of the standards in the field. Other rules have gained noteworthy attention as well, e.g. for cost critical applications and the quadrature of trimmed elements [34–39].

Conversely, the situation for floating tensor product B-Splines is more challenging, as the parametrization of a patch changes during large deformation analysis in order to remove mesh distortion. In [5], we proposed to advect the Gauss–Legendre points of the initial mesh, i.e. to treat them as material points in physical space (a procedure borrowed from meshless analysis). This strategy enjoys the advantages of the fully Lagrangian viewpoint of motion, however it also has a few shortcomings, most notably:

- The physical material points must be mapped back to the parametric domain to compute basis function values.
- The quadrature locations in parent/parametric space change over time following the deformation by an *a priori* unpredictable relation and thereby lose Gauss point character.
- The parent/parametric quadrature weights are also lost and the physical quadrature weights must be updated explicitly under the assumption of affine deformations.

Ultimately, this may result in accumulation of quadrature error and thus numerical instability in the long run. To address these shortcomings, after some preliminary considerations, we propose an improved Lagrangian quadrature scheme for floating B-Splines exploiting the underlying floating tensor product structure.

Preliminaries. Let us assume that a quadrature point stays fixed in the parametric space, $\boldsymbol{\xi}^* = (\xi^*, \eta^*)^T$, like in IGA. For a floating B-Spline basis, the physical coordinate of this point reads Eq. (11)

$$\boldsymbol{x}^* (\bullet; \mathcal{H}) = \sum_{j=1}^{J} \sum_{i=1}^{I_j} \boldsymbol{c}_{ij} \hat{B}_{ij} \left(\boldsymbol{\xi}^*; \mathcal{H} \right) = \sum_{j=1}^{J} \sum_{i=1}^{I_j} \boldsymbol{c}_{ij} \hat{N}_{ij} \left(\xi^*; \mathcal{H} \right) \hat{M}_j \left(\eta^* \right), \tag{13}$$

where we used Eq. (8). Pulling the characteristic bases back to the parent space with Eq. (7), we obtain

$$\boldsymbol{x}^* (\bullet; \mathcal{H}) = \sum_{j=1}^{J} \sum_{i=1}^{I_j} \boldsymbol{c}_{ij} \tilde{N}_{ij} \left(\tilde{\xi}_j^* (\bullet; \mathcal{H}) \right) \hat{M}_j \left(\eta^* \right). \tag{14}$$

As the evaluation locations of the static parent bases depend on the floating regulation points $\mathcal{H}$, upon floating, we change the physical position corresponding to $\boldsymbol{\xi}^*$ and hence lose the material point character of the quadrature point.

Let us now assume that a quadrature point stays fixed in the *parent* space, at coordinate $\left(\tilde{\xi}^*, \eta^* \right)^T$, and that it is mapped to the parametric space by a certain $\mathcal{G}_s$ with $s \in \{1, 2, \ldots, J\}$, such that with Eq. (4)

$$\boldsymbol{\xi}_s^* (\bullet; \mathcal{H}) = \left(\xi_s^* (\bullet; \mathcal{H}), \eta^* \right)^T = \left(\mathcal{G}_s \left(\tilde{\xi}^*; \mathcal{H} \right), \eta^* \right)^T. \tag{15}$$

Mapping to the physical space through Eq. (11) and using Eq. (8) gives

$$\boldsymbol{x}_s^* (\bullet; \mathcal{H}) = \mathcal{F} \left(\boldsymbol{\xi}_s^* (\bullet; \mathcal{H}); \mathcal{H} \right) = \sum_{j=1}^{J} \sum_{i=1}^{I_j} \boldsymbol{c}_{ij} \hat{B}_{ij} \left(\boldsymbol{\xi}_s^* (\bullet; \mathcal{H}); \mathcal{H} \right) = \sum_{j=1}^{J} \sum_{i=1}^{I_j} \boldsymbol{c}_{ij} \hat{N}_{ij} \left(\xi_s^* (\bullet; \mathcal{H}); \mathcal{H} \right) \hat{M}_j \left(\eta^* \right). \tag{16}$$

After pulling the characteristic bases back to the parent space with Eq. (7), the physical coordinate is alternatively expressed as

$$\begin{aligned} \boldsymbol{x}_s^* (\bullet; \mathcal{H}) &= \sum_{j=1}^{J} \sum_{i=1}^{I_j} \boldsymbol{c}_{ij} \tilde{N}_{ij} \left(\tilde{\xi}_{sj}^* (\bullet; \mathcal{H}) \right) \hat{M}_j \left(\eta^* \right) \\ &= \sum_{i=1}^{I_s} \boldsymbol{c}_{is} \tilde{N}_{is} \left(\tilde{\xi}^* \right) \hat{M}_s \left(\eta^* \right) + \sum_{\substack{j=1 \\ j \neq s}}^{J} \sum_{i=1}^{I_j} \boldsymbol{c}_{ij} \tilde{N}_{ij} \left(\tilde{\xi}_{sj}^* (\bullet; \mathcal{H}) \right) \hat{M}_j \left(\eta^* \right), \end{aligned} \tag{17}$$

with $\tilde{\xi}^*_{sj}(\bullet;\mathcal{H}) = \mathcal{G}_j^{-1}\left(\xi^*_s(\bullet;\mathcal{H});\mathcal{H}\right)$ and, in particular, $\tilde{\xi}^*_{ss} = \mathcal{G}_s^{-1}\left(\xi^*_s(\bullet;\mathcal{H});\mathcal{H}\right) = \tilde{\xi}^*$, see Eq. (6) and Eq. (15). From this equation we can see that the physical quadrature point coordinate becomes independent of the floating regulation points if a Kronecker-delta property w.r.t. s holds for $\hat{M}_j$ at η^*, in which case

$$\boldsymbol{x}^*_s = \sum_{i=1}^{I_s} \boldsymbol{c}_{is}\tilde{N}_{is}\left(\tilde{\xi}^*\right). \tag{18}$$

For B-Splines we can easily realize a Kronecker-delta property of $\hat{M}_j$ by setting $q = 1$ and adopting a Gauss–Lobatto quadrature scheme in the normal direction.

A new quadrature scheme. Let us now apply these concepts to construct a quadrature scheme in which the physical location of quadrature points is independent of the floating regulation, i.e. quadrature points have material point character. We set $q = 1$ (which is always assumed in the following) and choose the Gauss–Lobatto quadrature scheme in η, as it places the quadrature points η^l, $l = 1,\dots,2J-2$, at the Kronecker-delta positions of the normal basis. At inner knots in H, this 1D Gauss–Lobatto scheme gives *two* quadrature points (one from each adjacent knot span) and the total number of Gauss–Lobatto points is therefore larger than the number of characteristic bases. Hence, we formulate an index map deriving the index s of the supported characteristic basis at η^l from the Gauss–Lobatto point index l

$$s = \mathcal{S}(l) = 1 + \frac{l - (l\%2)}{2}, \tag{19}$$

where % is the modulo operator.

As a result of the above choices, all quadrature points are positioned at the boundaries of the characteristic elements, along which we perform quadrature at Gauss–Legendre[1] positions of the respectively supported parent basis functions $\tilde{N}_{is}$ (which are static apart from refinement in Section 2.4.3). However, instead of working directly with the Gauss–Legendre points of the parent knot spans, we evaluate the Gauss–Legendre points of potentially refined quadrature knot vectors $\tilde{\Xi}^l \supseteq \tilde{\Xi}_{\mathcal{S}(l)}$ as supersets of the parent knot vectors. This strategy grants us flexibility to adjust the point density of the quadrature set, which will be crucial in Section 2.4.3 within the context of adaptive refinement. In that context we will need the dependency of the quadrature knot vectors on l (and not just j) in order to perform manipulations dependent on which of the two adjacent characteristic elements a quadrature point belongs to.

Thus, we obtain the quadrature locations starting from combining the G-point Gauss–Legendre schemes $\mathcal{Q}^{Leg,G,l} = \mathcal{Q}^{Leg,G}\left(\tilde{\Xi}^l\right)$ for the quadrature knot vectors $\tilde{\Xi}^l$ (leading to a number of quadrature points $n^{QP,l}$ for each l) with the 2-point Gauss–Lobatto scheme $\mathcal{Q}^{Lob,2} = \mathcal{Q}^{Lob,2}(H)$ for the normal knot vector H, i.e.

$$\tilde{\boldsymbol{\xi}}^{gl} = \left\{\tilde{\xi}^{gl}, \eta^l\right\}, \tag{20}$$

with fixed $\tilde{\xi}^{gl} \in \mathcal{Q}^{Leg,G,l}$, $g = 1,\dots,n^{QP,l}$, and fixed $\eta^l \in \mathcal{Q}^{Lob,2}$, $l = 1,\dots,2J-2$. Note that $\tilde{\xi}^{gl}$ corresponds to $\tilde{\xi}^*$ in the above preliminary considerations.

The map of $\tilde{\xi}^{gl}$ to the parametric space reads

$$\xi^{gl}_s(\bullet;\mathcal{H}) = \mathcal{G}_s\left(\tilde{\xi}^{gl};\mathcal{H}\right), \tag{21}$$

with $s = \mathcal{S}(l)$ given by Eq. (19), and the quadrature point vector reads Eq. (15)

$$\boldsymbol{\xi}^{gl}_s(\bullet;\mathcal{H}) = \left(\xi^{gl}_s(\bullet;\mathcal{H}), \eta^l\right)^T. \tag{22}$$

Fig. 9 illustrates the novel concepts which we have introduced so far. We start the quadrature set construction on the left of the figure, where we highlight the quadrature point $\tilde{\xi}^{gl}$ ($l = 4$) that is the $g = 5$-th member of the Gauss–Legendre point set $\mathcal{Q}^{Leg,G}\left(\tilde{\Xi}^l\right)$ (using a $G = 3$-point rule). Here, $\tilde{\Xi}^l$ is twice as dense as $\tilde{\Xi}_s$ ($s = 3$); as we do not have adaptive refinement yet, all other quadrature knot vectors are identical. Note that the quadrature point is marked by a $\times$ symbol to emphasize its association to an *even* l, whereas for quadrature points associated with an *odd* l we use a $+$ symbol; in this way we distinguish between points with different l but the same s. We

[1] In principle, other quadrature rules for isogeometric analysis might as well be used, see e.g. the introductory references of the section.

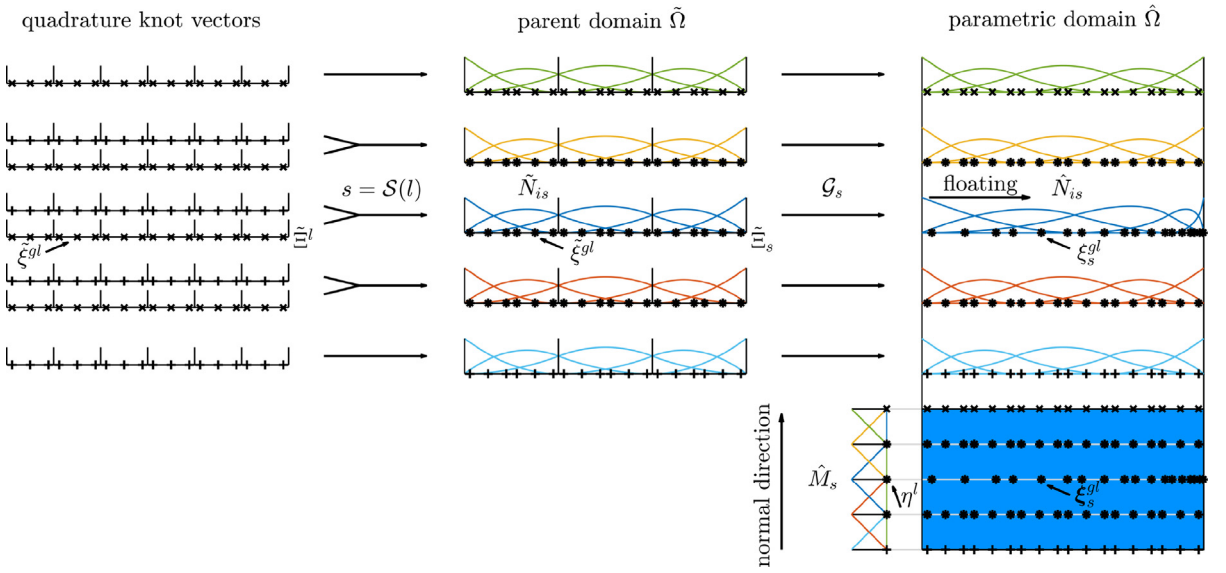


Fig. 9. A novel quadrature stencil for FLIGA.

push forward the quadrature parent coordinate by $\mathcal{G}_s$, obtaining the characteristic parametric coordinate ξ_s^{gl} which is floating-dependent. Lastly, the combination of ξ_s^{gl} with η^l follows the idea of a *floating* tensor product (i.e., ξ_s^{gl} depends on l, cf. Eq. (3)) and gives the parametric quadrature point vector $\boldsymbol{\xi}_s^{gl}$.

Let us now discuss the pull back of $\boldsymbol{\xi}_s^{gl}$ to parent coordinates for two cases, cf. Eq. (17). The pull back with the floating map in the first case, $j = s$, is trivial, as due to Eq. (21)

$$\tilde{\xi}_{ss}^{gl} = \mathcal{G}_s^{-1}\left(\xi_s^{gl}\left(\bullet;\mathcal{H}\right);\mathcal{H}\right) = \tilde{\xi}^{gl}, \tag{23}$$

and in the following, we will always simply write $\tilde{\xi}^{gl}$. For the second case, $j \neq s$, let us first introduce n as the index of the other normal basis function contained in the knot span of η^l (than the one with index s)

$$n = \mathcal{N}(l) = \mathcal{S}(l) + \beta, \qquad \begin{cases} \beta = 1, & \text{if } l \text{ is odd,} \\ \beta = -1, & \text{if } l \text{ is even,} \end{cases} \tag{24}$$

i.e., the index of the *neighbor* characteristic basis. Note that, due to compact support in normal direction ($\hat{M}_j\left(\eta^l\right) = \frac{\partial}{\partial\eta}\hat{M}_j\left(\eta^l\right) = 0$ if $(j \neq s \wedge j \neq n)$), it suffices for the second case to compute the pull back using only the neighbor floating map $\mathcal{G}_n^{-1}$,

$$\tilde{\xi}_{sn}^{gl}\left(\bullet;\mathcal{H}\right) = \mathcal{G}_n^{-1}\left(\xi_s^{gl}\left(\bullet;\mathcal{H}\right);\mathcal{H}\right). \tag{25}$$

In [5] we formulated a Newton method for the solution of scalar nonlinear equations of the type of Eq. (25).

Let us denote

$$\tilde{\boldsymbol{\xi}}_{sn}^{gl}\left(\bullet;\mathcal{H}\right) := \left\{\tilde{\xi}^{gl}, \tilde{\xi}_{sn}^{gl}\left(\bullet;\mathcal{H}\right), \eta^l\right\}, \tag{26}$$

as the set of all quadrature point coordinates, and highlight that the neighbor parent coordinate $\tilde{\xi}_{sn}^{gl}$, hence the set, depends on the floating regulation points. Writing $\tilde{\boldsymbol{\xi}}^{gl}$ without subscript sn, we exclude $\tilde{\xi}_{sn}^{gl}$ Eq. (20).

Computing basis functions and their gradients. Let us now compute the basis functions at quadrature point $\boldsymbol{\xi}_s^{gl}$ following Section 2.1 as

$$\begin{aligned}\hat{B}_{ij}\left(\boldsymbol{\xi}_s^{gl}\left(\bullet;\mathcal{H}\right);\mathcal{H}\right) &= \hat{N}_{ij}\left(\xi_s^{gl}\left(\bullet;\mathcal{H}\right);\mathcal{H}\right)\ \hat{M}_j\left(\eta^l\right) = \tilde{N}_{ij}\left(\tilde{\xi}_{sj}^{gl}\left(\bullet;\mathcal{H}\right)\right)\ \hat{M}_j\left(\eta^l\right)\\ &= \begin{cases}\tilde{N}_{is}\left(\tilde{\xi}^{gl}\right)\hat{M}_s\left(\eta^l\right), & \text{for } j=s,\\ \tilde{N}_{in}\left(\tilde{\xi}_{sn}^{gl}\left(\bullet;\mathcal{H}\right)\right)\hat{M}_n\left(\eta^l\right), & \text{for } j=n,\\ 0, & \text{otherwise,}\end{cases}\\ &= \begin{cases}\tilde{N}_{is}\left(\tilde{\xi}^{gl}\right), & \text{for } j=s,\\ 0, & \text{otherwise,}\end{cases}\\ &= \hat{B}_{ij}\left(\tilde{\boldsymbol{\xi}}^{gl}\right),\end{aligned} \tag{27}$$

considering the Kronecker-delta property of $\hat{M}_j$. Note that, in the final expression, all univariate basis function evaluations are of Gauss–Legendre character. Basis function gradients with respect to $\boldsymbol{\xi}$ at the quadrature point are

$$\begin{aligned}\frac{\partial}{\partial\boldsymbol{\xi}}\hat{B}_{ij}\left(\boldsymbol{\xi}_s^{gl}\left(\bullet;\mathcal{H}\right);\mathcal{H}\right) &= \begin{pmatrix}\dfrac{\partial}{\partial\xi}\hat{N}_{ij}(\xi_s^{gl}\left(\bullet;\mathcal{H}\right);\mathcal{H})\ \hat{M}_j(\eta^l)\\ \hat{N}_{ij}(\xi_s^{gl}\left(\bullet;\mathcal{H}\right);\mathcal{H})\ \dfrac{\partial}{\partial\eta}\hat{M}_j(\eta^l)\end{pmatrix}\\ &= \begin{pmatrix}J_j\left(\tilde{\xi}_{sj}^{gl}\left(\bullet;\mathcal{H}\right);\mathcal{H}\right)^{-1}\ \dfrac{\partial}{\partial\tilde{\xi}}\tilde{N}_{ij}\left(\tilde{\xi}_{sj}^{gl}\left(\bullet;\mathcal{H}\right)\right)\ \hat{M}_j(\eta^l)\\ \tilde{N}_{ij}\left(\tilde{\xi}_{sj}^{gl}\left(\bullet;\mathcal{H}\right)\right)\ \dfrac{\partial}{\partial\eta}\hat{M}_j\left(\eta^l\right)\end{pmatrix}.\\ &= \begin{cases}\begin{pmatrix}J_s\left(\tilde{\xi}^{gl};\mathcal{H}\right)^{-1}\ \dfrac{\partial}{\partial\tilde{\xi}}\tilde{N}_{is}\left(\tilde{\xi}^{gl}\right)\\ \tilde{N}_{is}\left(\tilde{\xi}^{gl}\right)\ \dfrac{\partial}{\partial\eta}\hat{M}_s\left(\eta^l\right)\end{pmatrix}, & \text{for } j=s,\\ \begin{pmatrix}0\\ \tilde{N}_{in}\left(\tilde{\xi}_{sn}^{gl}\left(\bullet;\mathcal{H}\right)\right)\ \dfrac{\partial}{\partial\eta}\hat{M}_n\left(\eta^l\right)\end{pmatrix}, & \text{for } j=n,\\ \begin{pmatrix}0\\0\end{pmatrix}, & \text{otherwise,}\end{cases}\\ &= \frac{\partial}{\partial\boldsymbol{\xi}}\hat{B}_{ij}\left(\tilde{\boldsymbol{\xi}}_{sn}^{gl}\left(\bullet;\mathcal{H}\right);\mathcal{H}\right),\end{aligned} \tag{28}$$

with the scalar Jacobian J_s of the supported floating map Eq. (4) derived from

$$J_j\left(\tilde{\xi};\mathcal{H}\right) = \frac{\partial}{\partial\tilde{\xi}}\mathcal{G}_j\left(\tilde{\xi};\mathcal{H}\right) = \sum_{i=1}^{I_j} h_{ij}\frac{\partial}{\partial\tilde{\xi}}\tilde{N}_{ij}\left(\tilde{\xi}\right). \tag{29}$$

Let us clarify that, while the neighbor normal basis function is not supported in terms of function value, $\hat{M}_n\left(\eta^l\right)=0$, it is supported in terms of the gradient $\dfrac{\partial}{\partial\eta}\hat{M}_n\left(\eta^l\right)\neq 0$. In the final expression of Eq. (28), including Eq. (29), all univariate basis function evaluations are of either Gauss–Legendre or Gauss–Lobatto character, except for the evaluation of $\tilde{N}_{in}\left(\tilde{\xi}_{sn}^{gl}\left(\bullet;\mathcal{H}\right)\right)$ appearing for $j=n$.

Basis function gradients with respect to $\boldsymbol{x}$ are

$$\frac{\partial}{\partial\boldsymbol{x}}\hat{B}_{ij}\left(\tilde{\boldsymbol{\xi}}_{sn}^{gl}\left(\bullet;\mathcal{H}\right);\mathcal{H}\right) = \boldsymbol{J}\left(\tilde{\boldsymbol{\xi}}_{sn}^{gl}\left(\bullet;\mathcal{H}\right);\mathcal{H}\right)^{-T}\ \frac{\partial}{\partial\boldsymbol{\xi}}\hat{B}_{ij}\left(\tilde{\boldsymbol{\xi}}_{sn}^{gl}\left(\bullet;\mathcal{H}\right);\mathcal{H}\right), \tag{30}$$

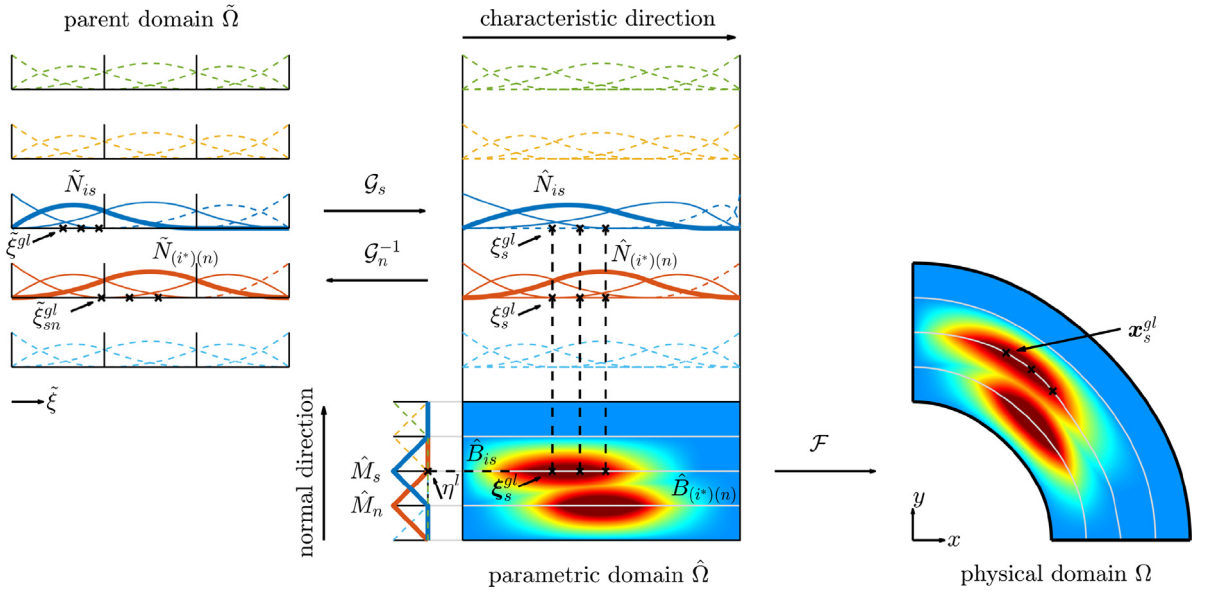


Fig. 10. Evaluation point characteristics for the novel quadrature scheme. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

with the Jacobian of the physical map Eq. (11)

$$\boldsymbol{J}\left(\tilde{\boldsymbol{\xi}}_{sn}^{gl}\left(\bullet;\mathcal{H}\right);\mathcal{H}\right)=\sum_{j=1}^{J}\sum_{i=1}^{I_j}\boldsymbol{c}_{ij}\left(\frac{\partial}{\partial\boldsymbol{\xi}}\hat{B}_{ij}\left(\tilde{\boldsymbol{\xi}}_{sn}^{gl}\left(\bullet;\mathcal{H}\right);\mathcal{H}\right)^{T}\right). \tag{31}$$

With the basis functions at hand, the physical quadrature point position is derived Eq. (11) and Eq. (27) as

$$\begin{aligned}\boldsymbol{x}_s^{gl}\left(\bullet;\mathcal{H}\right)&=\mathcal{F}\left(\boldsymbol{\xi}_s^{gl}\left(\bullet;\mathcal{H}\right);\mathcal{H}\right)=\sum_{j=1}^{J}\sum_{i=1}^{I_j}\boldsymbol{c}_{ij}\hat{B}_{ij}\left(\boldsymbol{\xi}_s^{gl}\left(\bullet;\mathcal{H}\right);\mathcal{H}\right)\\&=\sum_{j=1}^{J}\sum_{i=1}^{I_j}\boldsymbol{c}_{ij}\hat{B}_{ij}\left(\tilde{\boldsymbol{\xi}}^{gl}\right)=\sum_{i=1}^{I_s}\boldsymbol{c}_{is}\tilde{N}_{is}\left(\tilde{\xi}^{gl}\right)=\boldsymbol{x}_s^{gl},\end{aligned} \tag{32}$$

which shows dependency only on the control points (not the floating regulation points) confirming the Lagrangian character.

The evaluation of basis functions is demonstrated in Fig. 10 for two exemplary functions with $i=2, s=3$ (blue) and $i^*=3, n=2$ (red), respectively, proceeding the example of Fig. 9. Most notably the figure shows how the parent coordinates $\tilde{\xi}^{gl}$ and $\tilde{\xi}_{sn}^{gl}$ are different due to floating regulation (cf. Fig. 4). On a qualitative basis, we find that all function evaluations of $\tilde{N}_{is}$, $\hat{M}_s$, $\hat{M}_n$ at $G=3$ exemplary points of a given quadrature knot span are indeed of Gauss–Legendre or Gauss–Lobatto character; however, not those of $\tilde{N}_{(i^*)(n)}$.

Connectivities. Due to the local support of the basis functions, the implementation needs the connectivities at a quadrature point. With $\tilde{\xi}^{gl}\in\left[\tilde{\xi}_k\ ,\ \tilde{\xi}_{k+1}\right]$ for $\tilde{\xi}_k,\tilde{\xi}_{k+1}\in\tilde{\Xi}_s$ and $\tilde{\xi}_k\neq\tilde{\xi}_{k+1}$, the connectivities of $\tilde{\xi}^{gl}$ (alternatively expressed as $\tilde{\xi}_{ss}^{gl}$, Eq. (23)) to the supported parent basis $\left\{\tilde{N}_{is}\right\}_{i=1,\ldots,I_s}$ are

$$\mathcal{N}_{ss}^{gl}=\{k-p,\ k-p+1,\ \ldots,\ k\}. \tag{33}$$

Note that these are independent of the floating regulation points. Instead, with

$$\tilde{\xi}_{sn}^{gl}\left(\bullet;\mathcal{H}\right)\in\left[\tilde{\xi}_k\ ,\ \tilde{\xi}_{k+1}\right], \tag{34}$$

for $\tilde{\xi}_k, \tilde{\xi}_{k+1} \in \tilde{\Xi}_n$ and $\tilde{\xi}_k \neq \tilde{\xi}_{k+1}$, the connectivities of $\tilde{\xi}_{sn}^{gl}$ to the neighbor parent basis $\left\{\tilde{N}_{in}\right\}_{i=1,\dots,I_n}$ are

$$\mathcal{N}_{sn}^{gl}(\bullet;\mathcal{H}) = \{k(\bullet;\mathcal{H}) - p,\ k(\bullet;\mathcal{H}) - p + 1,\ \dots,\ k(\bullet;\mathcal{H})\}, \tag{35}$$

and update upon floating. Then, with Eq. (19) and Eq. (24) the constant connectivities of η^l to the normal basis $\left\{\hat{M}_j\right\}_{j=1,\dots,J}$, sorted in ascending order, are

$$\mathcal{M}^l = \text{sort}(\{s,\ n\}). \tag{36}$$

Hence, the updating connectivities of $\tilde{\boldsymbol{\xi}}_{sn}^{gl}(\bullet;\mathcal{H})$ to the bivariate basis $\left\{\hat{B}_{ij}\right\}_{i=1,\dots,I_j;\,j=1,\dots,J}$ are easily derived from the floating tensor product structure as the set of tuples

$$\mathcal{B}_{sn}^{gl}(\bullet;\mathcal{H}) = \left\{(i,j) \;\middle|\; i \in \mathcal{N}_{sj}^{gl}(\bullet;\mathcal{H}),\ j \in \mathcal{M}^l\right\}. \tag{37}$$

As with the classical tensor product structure, the total number of supported basis functions at $\tilde{\boldsymbol{\xi}}_{sn}^{gl}$ is $(p+1)\cdot(q+1)$, i.e., $(p+1)\cdot 2$. For implementation purposes, the translation of a tuple $(i,\ j)$ into a single running index $m = 1, \dots, M$ is most convenient by

$$m = \left(\sum_{\gamma=1}^{j-1} I_\gamma\right) + i, \tag{38}$$

where $\sum_{\gamma=1}^{0} I_\gamma := 0$ and I_γ is the number of basis functions in the characteristic basis with index γ. This expression allows to fill two arrays for associating $m \leftarrow (i,\ j)$ and $(i,\ j) \leftarrow m$.

Weights. The parametric weights w_s^{gl} are obtained from the weights of the Gauss–Legendre schemes $\mathcal{W}^{Leg,G,l} = \mathcal{W}^{Leg,G}\left(\tilde{\Xi}^l\right)$ and of the Gauss–Lobatto scheme $\mathcal{W}^{Lob,2} = \mathcal{W}^{Lob,2}(H)$ with Eq. (29) by

$$w_s^{gl}(\bullet;\mathcal{H}) = J_s\left(\tilde{\xi}^{gl};\mathcal{H}\right)\ \tilde{w}^{gl}\ w^l, \tag{39}$$

where $\tilde{w}^{gl} = \tilde{w}^g \in \mathcal{W}^{Leg,G,l}$ and $w^l \in \mathcal{W}^{Lob,2}$. The physical weight is obtained as

$$W_s^{gl}(\bullet;\mathcal{H}) = \det\left(\boldsymbol{J}\left(\tilde{\boldsymbol{\xi}}_{sn}^{gl}(\bullet;\mathcal{H});\mathcal{H}\right)\right)\ w_s^{gl}(\bullet;\mathcal{H}). \tag{40}$$

Finally, numerical approximation of a physical integral follows as

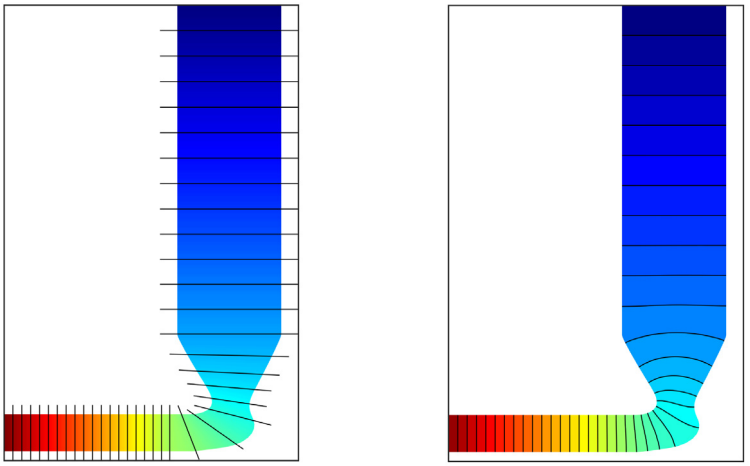
$$\int_\Omega [\bullet]\,\mathrm{d}\Omega \approx \sum_{l=1}^{2J-2}\sum_{g=1}^{n^{QP,l}} [\bullet]_{\boldsymbol{x}=\boldsymbol{x}_s^{gl}}\ W_s^{gl}(\bullet;\mathcal{H}). \tag{41}$$

Summary. Summarizing the new strategy, we highlight that the quadrature point $\tilde{\boldsymbol{\xi}}_{sn}^{gl}$

- is Lagrangian, i.e. moves as material point;
- has a constant parent coordinate $\tilde{\xi}^{gl}$ hence is part of a G-point Gauss–Legendre scheme for the associated pth order supported characteristic basis $\left\{\tilde{N}_{is}\right\}_{i=1,\dots,I_s}$;
- has a floating-dependent neighbor parent coordinate $\tilde{\xi}_{sn}^{gl}$ hence is *not* part of a Gauss–Legendre scheme for the associated pth order neighbor characteristic basis $\left\{\tilde{N}_{in}\right\}_{i=1,\dots,I_n}$;
- has a constant normal coordinate η^l hence is part of a 2-point Gauss–Lobatto scheme for the 1-st order normal basis $\left\{\hat{M}_j\right\}_{j=1,\dots,J}$;
- has constant parent and normal weights $\tilde{w}^{gl}$, w^l.

2.4.2. Automated floating regulation

In [5] we proposed the concept of a problem-specific level function to determine suitable floating regulation point positions for the reduction of mesh distortion. This function was designed so as to express the desired association of characteristic and physical coordinates, ξ and $\boldsymbol{x}$, respectively, and floating regulation points were then chosen so



(a) Manual design of static levels. (b) Automated PDE-based parametrization.

Fig. 11. Schematics of the old (a) and new (b) concept for the undistorted parametrization of physical space w.r.t. ξ. Colors indicate values of ξ. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

as to establish this relation. The level function construction was manual and based on an *a priori* guess about the deformations which is generally difficult to obtain. The significant construction effort suggested the use of only one level function to be applied to all load steps (and a local blending to force the levels in the rectangular shape of the parametric domain). We graphically summarize the old procedure in Fig. 11(a) on a simple example of extrusion, where the qualitative path of material flow is easy to predict.

In this paper, we replace this manual method with an automated approach, where the characteristic coordinate field is adjusted by recurrently solving a partial differential equation (PDE) on different deformed configurations; an exemplary time step result is illustrated in Fig. 11(b). In fact, the concept of PDE-based parametrization has been investigated before to determine analysis-suitable *control* point positions in IGA, especially in the context of ALE methods, see [8,40,41] and references therein. In our case however, an important requirement on the reparametrization procedure is to preserve the Lagrangian character of the quadrature points introduced in Section 2.4.1 which are dependent on the control points Eq. (32). For this reason, the control points are kept constant throughout the following procedure.

We start with introducing a floating B-Spline ansatz (indicated by superscript h) which expresses the parametric coordinate ξ at a physical point $\boldsymbol{x}$ in terms of the control and floating regulation points,

$$\xi^h\left(\boldsymbol{x};\mathcal{H}\right)=\sum_{m=1}^{M}h_m B_m\left(\boldsymbol{x};\mathcal{H}\right)=\xi\left(\bullet;\mathcal{H}\right), \tag{42}$$

see Appendix A for a proof. Here we consider the running index convention $m=1,\ldots,M$ from Eq. (38) and we omit denoting parameter dependencies on the control points. Let us highlight that also the basis functions in this ansatz depend on the floating regulation point coordinates which are unknown variables.

In the PDE-based reparametrization concept the differential-type Laplace operator is applied to this ansatz in physical space

$$\frac{\partial}{\partial\boldsymbol{x}}\cdot\frac{\partial}{\partial\boldsymbol{x}}\xi^h\left(\boldsymbol{x};\mathcal{H}\right)=0\qquad\text{in }\Omega, \tag{43}$$

complemented by the Dirichlet boundary conditions

$$\begin{aligned}\xi^h&=0\qquad\text{on }\tilde{\xi}=0,\\ \xi^h&=1\qquad\text{on }\tilde{\xi}=1,\end{aligned} \tag{44}$$

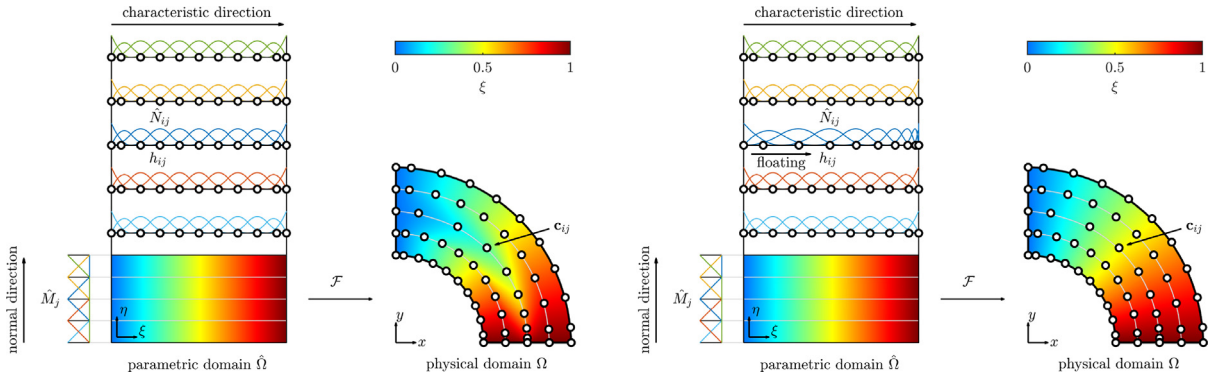


Fig. 12. Restoring an analysis-suitable parametrization by automated adjustment of floating regulation points.

and by the Neumann boundary conditions

$$\begin{aligned} \frac{\partial}{\partial \boldsymbol{x}} \xi^h(\boldsymbol{x}; \mathcal{H}) \cdot \boldsymbol{n} = 0 \qquad & \text{on } \eta = 0, \\ \frac{\partial}{\partial \boldsymbol{x}} \xi^h(\boldsymbol{x}; \mathcal{H}) \cdot \boldsymbol{n} = 0 \qquad & \text{on } \eta = 1, \end{aligned} \tag{45}$$

where $\boldsymbol{n}$ is the outward normal unit vector at the boundary. Let us now express Eq. (43) in weak form as

$$R_i = \sum_{l=1}^{2J-2} \sum_{g=1}^{n^{QP,l}} \sum_{m=1}^{M} \frac{\partial}{\partial \boldsymbol{x}} \hat{B}_i \left(\tilde{\boldsymbol{\xi}}_{sn}^{gl}(\bullet; \mathcal{H}); \mathcal{H} \right) \cdot \frac{\partial}{\partial \boldsymbol{x}} \hat{B}_m \left(\tilde{\boldsymbol{\xi}}_{sn}^{gl}(\bullet; \mathcal{H}); \mathcal{H} \right) \; h_m \; W_s^{gl}(\bullet; \mathcal{H}) \stackrel{!}{=} 0 \qquad \forall\, i, \tag{46}$$

where Eq. (44) is incorporated in the solution space (by setting $h_{1j} = 0,\ h_{(I_j)(j)} = 1\ \forall j = 1, \ldots, J$) and Eq. (45) is satisfied naturally. Here we make use of numerical integration by the concept introduced in Section 2.4.1. Obviously, the obtained problem is nonlinear in h_m (recall that $\mathcal{H}$ also contains the floating regulation points) and requires a solution e.g. by the iterative Newton–Raphson procedure (with tangent stiffness provided in Appendix B). We directly propose the solution $\mathcal{H}$ as suitable floating regulation point coordinates.

Note that Eq. (43) acts as a smoothing operator on the parametrization, while Eq. (44) ensures that the new parametrization fits the parametric domain. Eq. (45) forces the levels of ξ^h perpendicular to the physical domain boundaries along ξ, cf. Fig. 11(b). Hence we expect the procedure to determine floating regulation point positions such as to restore a regular ascent of ξ along the characteristic elements in physical space.[2] Importantly, the physical positions of the novel quadrature points along the characteristic element interfaces Eq. (32) are independent of the floating regulation point positions, such that no history data projection is required upon reparametrization.

In Fig. 12, we illustrate the restoration of an undistorted parametrization by the above procedure. The distorted (left) and reparametrized (right) B-Spline discretizations only differ in terms of the floating regulation points, whose appropriate choice resolves the problem of shear distortion of the basis functions along the characteristic direction. However, a distortion of the basis functions in terms of elongation or compression (seen e.g. from the control point distances) remains. This is addressed in the next section.

2.4.3. Local h-refinement

In the following, we describe an adaptive procedure for knot insertion and/or removal, which solves the problem of distortion of the basis functions in terms of elongation or compression. We finally outline the effect of this procedure on the quadrature scheme described in Section 2.4.1.

Adaptive knot insertion. We propose the standard *knot insertion* strategy [31] for h-refinement along the characteristic direction of a floating B-Spline geometry:

[2] Note that we here assume that the characteristic element interfaces inside the domain always reasonably align to the outer physical domain boundaries, and have spared the imposition of additional constraints on ξ^h at such interfaces.

- Choose a $j \in \{1, \dots, J\}$ specifying the characteristic basis to be refined;
- choose a knot $\tilde{\xi}^+$ to insert which lies within knot span $\left(\tilde{\xi}_k, \tilde{\xi}_{k+1}\right)$ of parent knot vector $\tilde{\Xi}_j$;
- for each $k-p+1 \leq i \leq k$: Compute the ratio $a_i = \dfrac{\tilde{\xi}^+ - \tilde{\xi}_i}{\tilde{\xi}_{i+p} - \tilde{\xi}_i}$ with $\tilde{\xi}_i, \tilde{\xi}_{i+p} \in \tilde{\Xi}_j$ as well as the

  new floating regulation point $h_{ij}^+ = (1-a_i)h_{(i-1)j} + a_i h_{ij}$
  new control point $\boldsymbol{c}_{ij}^+ = (1-a_i)\boldsymbol{c}_{(i-1)j} + a_i \boldsymbol{c}_{ij}$
  new control variable for each field approximation, e.g. $\boldsymbol{d}_{ij}^+ = (1-a_i)\boldsymbol{d}_{(i-1)j} + a_i \boldsymbol{d}_{ij}$;

- for each $k-p+1 \leq i \leq k-1$: remove the

  old floating regulation point h_{ij}
  old control point $\boldsymbol{c}_{ij}$
  old control variable for each field approximation, e.g. $\boldsymbol{d}_{ij}$;
  and insert in the gap the respective p new quantities;

- insert $\tilde{\xi}^+$ in $\tilde{\Xi}_j$ obtaining the refined parent knot vector $\tilde{\Xi}_j^+ = \left\{..., \ \tilde{\xi}_k, \ \tilde{\xi}^+, \ \tilde{\xi}_{k+1}, \ ...\right\}$.

Note that the number of characteristic basis functions associated to the specific j as well as the global number of basis functions hereby increase by one. This refinement has *local* nature since we construct the other characteristic bases from different parent knot vectors (recall Section 2.1). For spline order $q = 1$ in η direction all parametrizations $\mathcal{G}_j$ and $\mathcal{F}$ are preserved by the proposed refinement concept, hence also the material point character of the quadrature points.

We determine the knots to be inserted adaptively during the analysis by evaluating for each (non-zero) parent knot span $\tilde{\Omega}_{kj} = \left[\tilde{\xi}_k, \ \tilde{\xi}_{k+1}\right]$ with $\tilde{\xi}_k, \tilde{\xi}_{k+1} \in \tilde{\Xi}_j$ the length of its physical representation as curve segment

$$L_{kj} = \int_{\Omega_{kj}} 1 \cdot \mathrm{d}s \approx \sum_g L^{gl} = \sum_g \tilde{w}^{gl} \left\| \sum_{i=1}^{I_j} \boldsymbol{c}_{ij} \frac{\partial}{\partial \tilde{\xi}} \tilde{N}_{ij}\left(\tilde{\xi}^{gl}\right) \right\|_2, \tag{47}$$

where $\Omega_{kj} = \mathcal{F}\left(\mathcal{G}_j\left(\tilde{\Omega}_{kj}\right)\right)$; we choose $l : \mathcal{S}(l) = j$ (inner interfaces are associated to two quadrature knot vectors, we can choose either one); and summation is over all $g : \tilde{\xi}^{gl} \in \mathcal{Q}^{Leg,G,l} \cap \tilde{\Omega}_{kj}$. We set a length threshold T_{ins}, and if

$$L_{kj} > T_{ins}, \tag{48}$$

we insert a new knot

$$\tilde{\xi}^+ = 0.5 \cdot \left(\tilde{\xi}_k + \tilde{\xi}_{k+1}\right), \tag{49}$$

at the center of $\tilde{\Omega}_{kj}$.

Adaptive knot removal. The inverse process of knot insertion is called *knot removal*. Note that unlike knot insertion, removal of knots conserves the exact parametrization only under specific circumstances [31]. For this reason, we follow one of the strategies proposed in [42] for standard B-Spline curves, minimizing the change of geometry and parametrization in the sense of a certain L^∞ norm. Its transfer to the floating tensor product structure yields an algorithm as follows:

- Choose a $j \in \{1, \dots, J\}$ specifying the characteristic basis to be coarsened;
- choose a knot $\tilde{\xi}^- = \tilde{\xi}_k$ to remove which bounds the knot span $\left[\tilde{\xi}_k, \tilde{\xi}_{k+1}\right)$ of parent knot vector $\tilde{\Xi}_j$;
- if $p > 1$:

  compute or set:

$$l_r = \frac{\tilde{\xi}^- - \tilde{\xi}_{k-p+r-1}}{\tilde{\xi}_{k+r} - \tilde{\xi}_{k-p+r-1}} \qquad \text{for } r = 1, \dots, p+1,$$

$$\gamma_\infty = \sum_{t=1}^{p} \begin{bmatrix} p \\ t+1 \end{bmatrix},$$

$$\mu_r = \frac{1}{\gamma_\infty} \sum_{t=1}^{r-1} \begin{bmatrix} p \\ t+1 \end{bmatrix} \qquad \text{for } r = 1, \dots, p-1,$$

with bracket operator

$$\begin{bmatrix} b \\ a \end{bmatrix} := \begin{cases} \frac{1}{l_a} \prod_{r=a}^{b} \frac{1-l_{r+1}}{l_{r+1}}, & \text{for } a \leq b \\ \frac{1}{l_a}, & \text{for } a = b+1 \\ \frac{1}{1-l_a}, & \text{for } a = b+2, \end{cases}$$

$$\boldsymbol{c}_1^{\mathrm{I}} = \boldsymbol{c}_{(k-p-1)(j)}; \qquad \boldsymbol{c}_r^{\mathrm{I}} = \frac{1}{l_r} \boldsymbol{c}_{(k+r-2)(j)} + \left(1 - \frac{1}{l_r}\right) \boldsymbol{c}_{r-1}^{\mathrm{I}} \quad \text{for } r = 2, \dots, p,$$

$$\boldsymbol{c}_{p+1}^{\mathrm{II}} = \boldsymbol{c}_{(k)(j)}; \qquad \boldsymbol{c}_r^{\mathrm{II}} = \frac{1}{1-l_{r+1}} \boldsymbol{c}_{(k+p-r-1)(j)} + \left(1 - \frac{1}{l_{r+1}}\right) \boldsymbol{c}_{r+1}^{\mathrm{II}} \quad \text{for } r = p, p-1, \dots, 2$$

;

for $r = 1, \dots, p-1$: replace the old control point $\boldsymbol{c}_{(k-p+r-1)(j)} = (1-\mu_r)\, \boldsymbol{c}_{r+1}^{\mathrm{I}} + \mu_r \boldsymbol{c}_{r+1}^{\mathrm{II}}$;

- remove the old control point $\boldsymbol{c}_{(k-1)(j)}$;
- remove $\tilde{\xi}^-$ from $\tilde{\Xi}_j$ obtaining the coarsened parent knot vector $\tilde{\Xi}_j^- = \left\{ ..., \ \tilde{\xi}_{k-1}, \ \tilde{\xi}_{k+1}, \ ... \right\}$.

The procedure for deriving the new floating regulation points and control variables of each field approximation is fully equivalent. Like knot insertion, knot removal along the characteristic direction is a local process for floating B-Splines, initiated for any parent knot span with physical span length

$$L_{kj} < T_{rem}, \tag{50}$$

where T_{rem} is the removal threshold. Fig. 13 demonstrates how the elongational/compressive mesh distortion that remained in Fig. 8 is cured by adaptive knot insertion and removal in $\tilde{\Xi}_j$ (with $j = 3$) at the locations indicated by arrows.

Effect on quadrature. The quadrature knot vectors $\tilde{\Xi}^l$ introduced in Section 2.4.1 are unaffected by knot insertion or removal in the parent knot vectors along the course of analysis. While we cannot precisely predict in advance which parent knot spans will be split how often, we simply initialize the quadrature point set by sufficiently dense quadrature knot vectors $\tilde{\Xi}^l$. The insertion of new quadrature points along with the mapping of history data is thereby avoided even upon adaptive refinement. To this end, before the analysis, we estimate (in terms of knot span density) the maximum refinement of the parent knot vectors occurring during the analysis, and accordingly construct all $\tilde{\Xi}^l$. With the convention to insert new parent knots only at *centers* of parent knot spans Eq. (49) the one-time construction of $\tilde{\Xi}^l$ is straightforward and preserves all beneficial properties of the novel quadrature concept. To build the quadrature vectors, we propose a simple split of each knot span of the (assumed uniform[3]) initial parent knot vectors into a certain number of subspans, according to

$$\tilde{\Xi}^l = \mathcal{K}\left(\max\left(\rho_{\mathcal{S}(l)} n^S_{\mathcal{S}(l)}, \ 2\rho_{\mathcal{N}(l)} n^S_{\mathcal{N}(l)}\right), p\right), \tag{51}$$

where $\mathcal{K}(n^S, p)$ denotes an open uniform knot vector with n^S (non-zero) knot spans for polynomial order p. Further, n_j^S denotes the number of knot spans in the initial (unrefined) $\tilde{\Xi}_j$, and $\rho_j \geq 1$ (chosen as a power of 2) is the assumed maximum parent knot span relative shrinkage due to (possibly nested) refinement in $\tilde{\Xi}_j$. We recommend to multiply a factor of two to $\rho_{\mathcal{N}(l)}$ in Eq. (51), in order to improve quadrature accuracy for integration of the *neighbor* characteristic basis (as its quadrature is not of Gauss character, recall Section 2.4.1). Note that an overestimation

[3] The idea is readily generalized to the non-uniform case as well, but $\tilde{\Xi}^l$ must be a superset of $\tilde{\Xi}_j$.

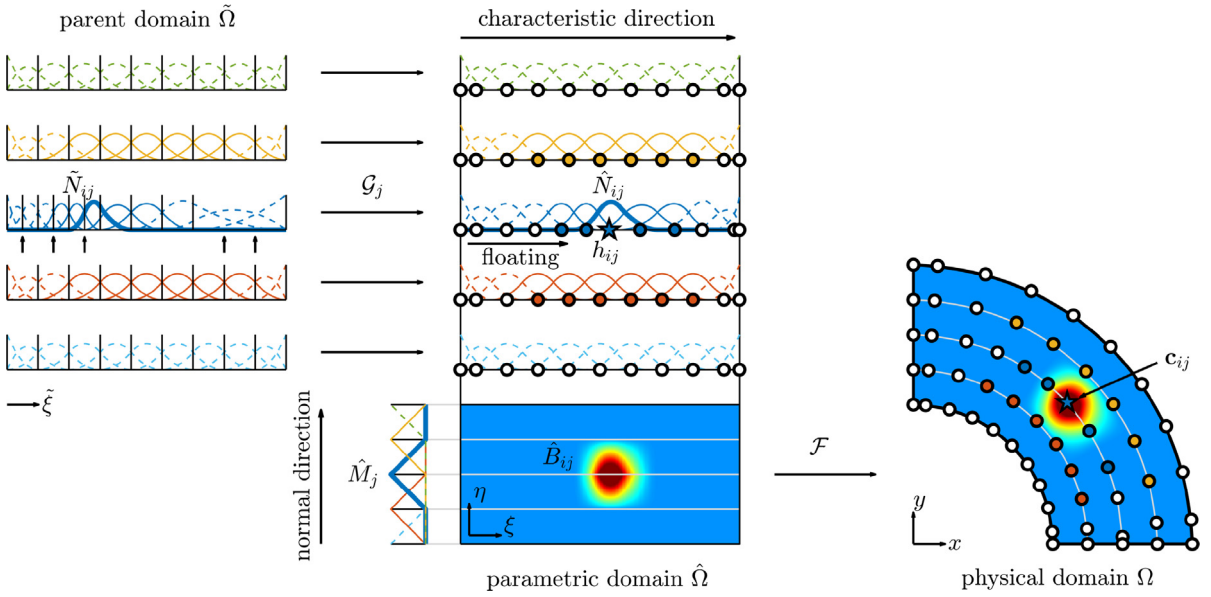


Fig. 13. Removed shear *and* dilatational mesh distortion in FLIGA.

of the quadrature knot span density does not lead to any difficulties (except for an increase in computational cost), whereas an underestimation may cause problems due to under-integration after repeated knot insertion in the parent knot vectors.

To make this clearer, let us give an example of how to construct the quadrature knot vectors for adaptive refinement, assuming that the course of refinement is known in advance. To this end, we consider once more the example of Fig. 9, for simplicity in an "unfloated" configuration without loss of generality. In the figure, as refinement was not yet introduced in Section 2.4.1 ($\rho_j = 1$), we had equal and uniform parent knot vectors $\tilde{\Xi}_j = \tilde{\Xi} = \mathcal{K}(n^S, p)$ and equal and uniform quadrature knot vectors $\tilde{\Xi}^l = \mathcal{K}\left(2n^S, p\right)$, with $n^S = 3$. Now we assume that the knot spans of parent knot vector $\tilde{\Xi}_3$ are expected to undergo a split to half the size, i.e. $\rho_3 = 2$, while for $j = 1, 2, 4, 5$ no refinement is expected, $\rho_{j\neq 3} = 1$, see Fig. 14. In this setting, according to Eq. (51), we construct the quadrature knot vectors:

$$
\begin{aligned}
&\bullet\ l = 1, 2, 7, 8: && \tilde{\Xi}^l = \mathcal{K}\left(2\rho_{j\neq 3} n^S, p\right) && = \mathcal{K}\left(6, p\right)\\
&\bullet\ l = 3, 6: && \tilde{\Xi}^l = \mathcal{K}\left(2\rho_{j=3} n^S, p\right) && = \mathcal{K}\left(12, p\right)\\
&\bullet\ l = 4, 5: && \tilde{\Xi}^l = \mathcal{K}\left(2\rho_{j\neq 3} n^S, p\right) = \mathcal{K}\left(\rho_{j=3} n^S, p\right) && = \mathcal{K}\left(6, p\right).
\end{aligned}
$$

We now see the motivation for the introduction of quadrature knot vectors related to l and not to j in Section 2.4.1, in that superfluous points are avoided in $\tilde{\Xi}^2$ and $\tilde{\Xi}^7$. These are associated to either one of the two outer characteristic elements in which the refined basis functions of $\left\{\hat{B}_{i3}\right\}_{i=1,\dots,I_3}$ are not supported.

In many cases, the exact refinement is *a priori* unknown, hence ρ_j is just estimated. We highlight once more that the quadrature stencil is designed for the most refined state and employed at all load steps. Mapping of history data is not required.[4]

A remark on p-refinement. In this paper we assume that all characteristic bases are of the same polynomial order p. In principle, it is as well possible to assign to them different p_j, enabling a p-refinement along ξ which is local in the normal direction. This could be beneficial for the adjustment of approximation spaces to the smoothness of the solution. E.g., in the context of extrusion problems, boundary conditions may suddenly change (most notably at the exit of the nozzle, see Sections 4.3 and 4.4), while a highly smooth deformation behavior is expected inside the strand. Hence, the selection of lower polynomial orders at the characteristic strand boundaries (with locally small

[4] For history-independent behavior the quadrature knot vectors can be adapted to the current parent knot vectors for cost reduction.

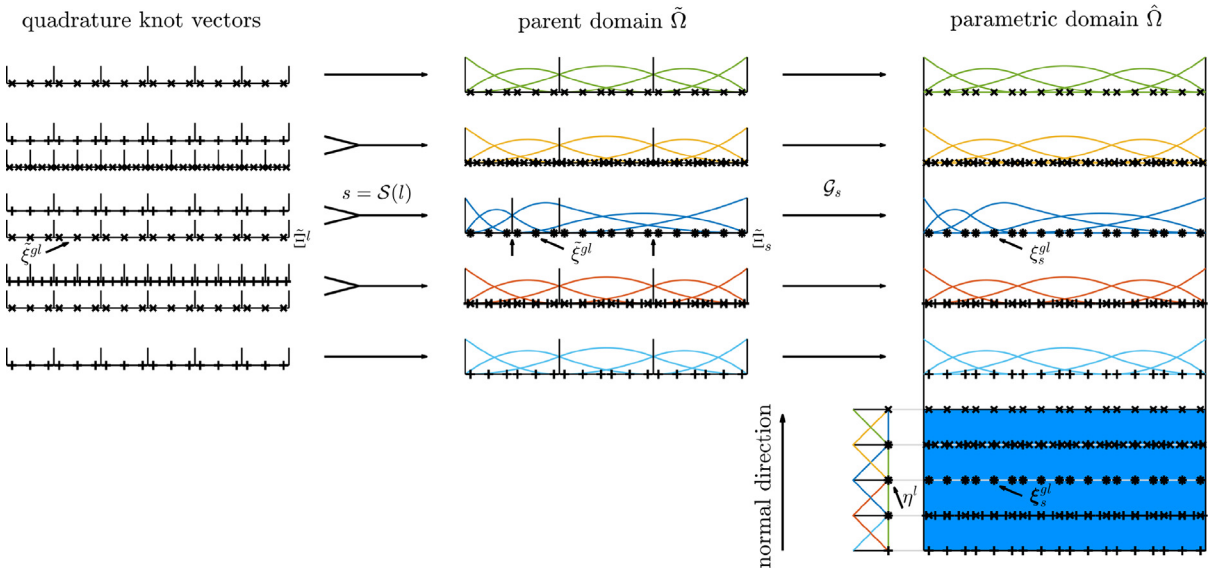


Fig. 14. Quadrature and refinement.

h-refinement thresholds) and higher polynomial orders in the interior (with large thresholds) could be an attractive alternative.

2.4.4. Function properties

Among others, enhanced floating B-Splines possess the following analysis-suitable properties inherited from conventional B-Splines:

- partition of unity
- weak Kronecker-delta property at the boundaries
- first-order consistency
- boundary preservation
- interface preservation

In [5] we provided proofs for the first four of these properties whose extensions to the enhanced splines are trivial and therefore omitted here. The interface preservation property follows immediately from the Lagrangian character of the interface points which was shown in Section 2.4.1.

2.4.5. Enhancements summary

Let us conclude the overview of the enhancements with a brief summary. Firstly, the novel quadrature concept resolves all the drawbacks of the previous advection procedure in [5], however, it solves the problem of loss of Gauss point character only to a large extent. Secondly, we have proposed an automated procedure for floating regulation to comfortably overcome shear distortion of the basis functions along the characteristic direction. Thirdly, adaptive knot insertion and removal prevent excessive basis function elongation/compression in the same direction. The combination of distortion-free approximation spaces with fully Lagrangian, yet highly systematic quadrature provides an ideal basis for numerical analysis of extreme, history-dependent deformations. As a limitation, the effectiveness of the proposed concepts relies on the orientation of the extreme deformations along one characteristic direction. Hence, the application to viscoelastic extrusion-based processes (Section 1) is a natural choice. The related analysis concepts are introduced next.

2.4.6. Comparison with ALE

The ALE method that was mentioned in the introduction shares some fundamental concepts with FLIGA. Let us briefly compare the two. Firstly, both techniques rely on boundary-fitted approximation spaces, with the behavior

of the basis functions lying in between the Lagrangian and Eulerian viewpoints to compensate for mesh distortion. Similar is also the practice to construct the analysis-suitable meshes by application of a PDE-based parametrization operation. However, in (IGA-based) ALE this operation modifies the control point positions, whereas in FLIGA it only affects the positions of the floating regulation points (see Section 2.4.2). Therefore, in ALE (unlike in FLIGA) the movement of the control points is decoupled from the material deformation. This leads to different choices of the viewpoint of motion at the quadrature points in the two approaches: ALE typically employs Gauss point quadrature, which brings the benefit of accurate integral evaluations. Unfortunately, the loss of Lagrangian character at the control points immediately translates to the Gauss points and relative advection terms must be added to the model equations. Such terms do not appear in FLIGA, where the control points are updated according to the deformations and, hence, material points are a favorable choice for quadrature. The general downside of material point quadrature is that the locations of the points at which the basis functions are evaluated are fully determined by the course of deformation, making the accurate computation of integrals more challenging (see Section 2.4.1). Finally, ALE (unlike FLIGA) does not require the presence of a characteristic direction in the problem to be solved.

3. Continuum model, IGA and enhanced FLIGA

In this section we first present the continuum mechanical model for viscoelastic finite deformation problems following the Lagrangian viewpoint of motion, and then illustrate the conventional IGA and enhanced FLIGA discretizations.

3.1. Continuum problem

Consider the quasi-static balance of linear momentum

$$\frac{\partial}{\partial \boldsymbol{x}} \cdot \boldsymbol{\sigma} + \boldsymbol{f} = \boldsymbol{0} \qquad \text{in } \Omega, \tag{52}$$

where $\boldsymbol{\sigma}$ is the Cauchy stress and $\boldsymbol{f}$ the external body force, with Dirichlet boundary conditions on the velocity $\boldsymbol{v}$

$$\boldsymbol{v} = \boldsymbol{v}_D \qquad \text{on } \partial\Omega_D, \tag{53}$$

as well as Neumann boundary conditions

$$\boldsymbol{\sigma}\boldsymbol{n} = \boldsymbol{h}_N \qquad \text{on } \partial\Omega_N. \tag{54}$$

Here, $\boldsymbol{h}_N$ is the surface traction vector. Further, the velocity is derived as $\boldsymbol{v} = \dot{\boldsymbol{u}}$ from the displacement $\boldsymbol{u}$. We also enforce the incompressibility condition

$$\frac{\partial}{\partial \boldsymbol{x}} \cdot \boldsymbol{v} = 0 \qquad \text{in } \Omega, \tag{55}$$

and in preparation for viscoelastic material models, we split the Cauchy stress in a hydrostatic, a solvent, and a polymeric extra stress

$$\boldsymbol{\sigma} = -p\mathbf{I} + 2\eta_s \nabla^s \boldsymbol{v} + \boldsymbol{\tau}_p. \tag{56}$$

Here, p is the pressure, $\nabla^s(\bullet) = \frac{1}{2}\left(\frac{\partial(\bullet)}{\partial \boldsymbol{x}}\right) + \frac{1}{2}\left(\frac{\partial(\bullet)}{\partial \boldsymbol{x}}\right)^T$ the symmetric gradient of vector $(\bullet)$, and η_s the solvent viscosity. The polymeric extra stress $\boldsymbol{\tau}_p$ evolves according to an ordinary differential equation in time

$$\dot{\boldsymbol{\tau}}_p = \dot{\boldsymbol{\tau}}_p\left(\frac{\partial \boldsymbol{v}}{\partial \boldsymbol{x}}, \boldsymbol{\tau}_p; \{C_k\}\right), \tag{57}$$

starting from the initial value $\boldsymbol{\tau}_p^0$. The set of material parameters $\{C_k\}$ may include parameters such as relaxation time λ and polymeric viscosity η_p. A detailed introduction to viscoelastic modeling is given in [43]. In our work, we focus only on the classical Oldroyd-B model where

$$\dot{\boldsymbol{\tau}}_p = \frac{\partial \boldsymbol{v}}{\partial \boldsymbol{x}}\,\boldsymbol{\tau}_p + \boldsymbol{\tau}_p\left(\frac{\partial \boldsymbol{v}}{\partial \boldsymbol{x}}\right)^T - \frac{1}{\lambda}\left(\boldsymbol{\tau}_p - 2\eta_p \nabla^s \boldsymbol{v}\right). \tag{58}$$

Upon loading the body deforms over time, so that

$$\boldsymbol{x}(\boldsymbol{X},t) = \boldsymbol{X} + \boldsymbol{u}(\boldsymbol{X},t) = \boldsymbol{X} + \int_0^t \boldsymbol{v}(\boldsymbol{X},\tau)\, d\tau, \tag{59}$$

where $\boldsymbol{X}$ is a generic coordinate in the initial configuration. By referring the description of motion to the initial configuration, we apply the Lagrangian viewpoint of motion. Note that the (material) time derivative $\dot{(\,)}$ in Eq. (57) and Eq. (58) describes the change of stress over time focusing on a material point $\boldsymbol{X}$.

For problems involving contact, we devise a penalty formulation with respect to wall penetration and penetration rate, augmented by another penalty formulation to regularize the no-slip constraint, $\boldsymbol{v} = 0$. We arrive at the contact boundary condition at the wall interface $\partial\Omega_C$

$$\boldsymbol{\sigma}\boldsymbol{n} = -\kappa_P^* P\boldsymbol{n} - \kappa_R^* \dot{P}\boldsymbol{n} - \kappa_S^* \boldsymbol{v} \qquad \text{on } \partial\Omega_C, \tag{60}$$

where P is the wall penetration, $\dot{P}$ the wall penetration rate and $\boldsymbol{n}$ the outward normal unit vector. Finally, the active penalty parameters, $\kappa_P^*,\ \kappa_R^*,\ \kappa_S^*$ are,

$$\kappa_P^* = \begin{cases} \kappa_P, & \text{for } P > 0, \\ 0, & \text{otherwise,} \end{cases} \qquad \kappa_R^* = \begin{cases} \kappa_R, & \text{for } \dot{P} > 0, \\ 0, & \text{otherwise,} \end{cases} \qquad \kappa_S^* = \begin{cases} \kappa_S, & \text{for } P > 0, \\ 0, & \text{otherwise,} \end{cases} \tag{61}$$

based on the constant penalty coefficients, $\kappa_P,\ \kappa_R,\ \kappa_S$, respectively.

3.2. Isogeometric discretization

For classical IGA discretization of the problem presented above, we describe the geometry of the undeformed domain based on isogeometric basis functions, e.g. standard B-Splines (see Section 2 and references therein). With the geometry representation available, discretization in space (indicated by index h) and in time (indicated by time step index n for time step t^n) of the unknown velocity and pressure fields follows in case of a mixed method, using Eq. (10), as

$$\boldsymbol{v}^{h,n}(\boldsymbol{x}) = \sum_{m=1}^{M} B_m^{TP}(\boldsymbol{x})\, \boldsymbol{d}_m^n = \sum_{m=1}^{M} \hat{B}_m^{TP}(\boldsymbol{\xi})\, \boldsymbol{d}_m^n = \boldsymbol{v}^{h,n}(\boldsymbol{\xi}), \tag{62}$$

$$p^{h,n}(\boldsymbol{x}) = \sum_{z=1}^{Z} A_z^{TP}(\boldsymbol{x})\, q_z^n = \sum_{z=1}^{Z} \hat{A}_z^{TP}(\boldsymbol{\xi})\, q_z^n = p^{h,n}(\boldsymbol{\xi}), \tag{63}$$

where $\left\{\hat{B}_m^{TP}\right\}_{m=1,\dots,M}$ and $\left\{\hat{A}_z^{TP}\right\}_{z=1,\dots,Z}$ are classical B-Spline bases and $\boldsymbol{d}_m^n$ and q_z^n are the current velocity and pressure control variables, respectively. Further, the isoparametric concept is applied, meaning that $\left\{\hat{B}_m^{TP}\right\}_{m=1,\dots,M}$ here is the same basis that describes the mapping between parametric and physical space, Eq. (9). Inserting the velocity ansatz in the weak form of Eq. (52) leads to the equilibrium of internal and external force vectors at each control point $\boldsymbol{c}_m^n$ and each time step n

$$\boldsymbol{F}_{int,m}^n = \sum_{g=1}^{n^{QP}} \left[\left(\boldsymbol{\sigma}^{h,n} \frac{\partial}{\partial \boldsymbol{x}} \hat{B}_m^{TP} \right)\Big|_{\boldsymbol{\xi}^g} W^{g,n} \right] \overset{!}{=} \sum_{g=1}^{n^{QP}} \left[\left(\boldsymbol{f}^n \hat{B}_m^{TP} \right)\Big|_{\boldsymbol{\xi}^g} W^{g,n} \right] = \boldsymbol{F}_{ext,m}^n. \tag{64}$$

Here we assume $\boldsymbol{h}_N^n = \boldsymbol{0}$ on the Neumann boundary, which is satisfied naturally in the weak formulation. Dirichlet boundary conditions can be incorporated *a priori* in the solution space. Numerical integration is applied over Gauss–Legendre quadrature points with coordinates $\boldsymbol{\xi}^g$ and physical weights $W^{g,n}$, where $g = 1, \dots, n^{QP}$. Note that each $\boldsymbol{\xi}^g$ is constant over time in standard IGA.

Likewise, we insert the velocity ansatz in the weighted residual form of Eq. (55), obtaining one equation for each pressure control variable at each time step

$$Q_z^n = -\sum_{g=1}^{n^{QP}} \left[\left(\frac{\partial}{\partial \boldsymbol{x}} \cdot \boldsymbol{v}^{h,n}\, \hat{A}_z^{TP} \right)\Big|_{\boldsymbol{\xi}^g} W^{g,n} \right] \overset{!}{=} 0. \tag{65}$$

We multiply here by -1 to obtain a symmetric stiffness matrix later. The discretized Cauchy stress evaluated at $\boldsymbol{\xi}^g$ is given by

$$\boldsymbol{\sigma}^{h,g,n} = -p^{h,g,n}\mathbf{I} + 2\eta_s \nabla^s \boldsymbol{v}^{h,g,n} + \boldsymbol{\tau}_p^{h,n,g}\left(\frac{\partial \boldsymbol{v}^{h,g,n-1}}{\partial \boldsymbol{x}}, \boldsymbol{\tau}_p^{h,g,n-1}; \{C_k\}; \Delta t\right), \tag{66}$$

where Δt is the time step size. This expression makes use of the time discretization of Eq. (57) by the forward Euler method, so that the polymeric extra stress is independent of the current unknown solution. In the Lagrangian framework, history data at each Gauss–Legendre point are available from the previous time step $n-1$. For the Oldroyd-B model in Eq. (58), we obtain the current polymeric extra stress by

$$\boldsymbol{\tau}_p^{h,g,n} = \boldsymbol{\tau}_p^{h,g,n-1} + \Delta t\left[\frac{\partial \boldsymbol{v}^{h,g,n-1}}{\partial \boldsymbol{x}}\ \boldsymbol{\tau}_p^{h,g,n-1} + \boldsymbol{\tau}_p^{h,g,n-1}\ \left(\frac{\partial \boldsymbol{v}^{h,g,n-1}}{\partial \boldsymbol{x}}\right)^T - \frac{1}{\lambda}\left(\boldsymbol{\tau}_p^{h,g,n-1} - 2\eta_p \nabla^s \boldsymbol{v}^{h,g,n-1}\right)\right]. \tag{67}$$

Let us also discretize the rate form of Eq. (59) in space and time

$$\dot{\boldsymbol{x}}^{h,n}\left(\mathcal{F}^0(\boldsymbol{\xi})\right) = \boldsymbol{v}^{h,n}\left(\mathcal{F}^0(\boldsymbol{\xi})\right), \tag{68}$$

where $\mathcal{F}^0$ is the physical mapping at the initial time step $n=0$, or equivalently

$$\dot{\boldsymbol{x}}^{h,n}(\boldsymbol{\xi}) = \boldsymbol{v}^{h,n}(\boldsymbol{\xi}). \tag{69}$$

Inserting the basis functions Eq. (9) and Eq. (62) yields

$$\sum_{m=1}^{M} \dot{\boldsymbol{c}}_m^n \hat{B}_m^{TP}(\boldsymbol{\xi}) = \sum_{m=1}^{M} \boldsymbol{d}_m^n \hat{B}_m^{TP}(\boldsymbol{\xi}), \tag{70}$$

which is then discretized again by the forward Euler method, resulting in

$$\boldsymbol{c}_m^n = \boldsymbol{c}_m^{n-1} + \Delta t\ \boldsymbol{d}_m^{n-1}. \tag{71}$$

Hence the construction of basis functions at the current configuration is independent of the current velocity solution. Collecting the individual equations from Eq. (64) and Eq. (65) for each time step, we obtain the global residual vectors

$$\boldsymbol{R}^n(\boldsymbol{U}^n) = \boldsymbol{F}_{int}^n(\boldsymbol{U}^n) - \boldsymbol{F}_{ext}^n \overset{!}{=} \mathbf{0}, \tag{72}$$

and

$$\boldsymbol{Q}^n(\boldsymbol{U}^n) \overset{!}{=} \mathbf{0}, \tag{73}$$

respectively. $\boldsymbol{U}^n$ contains the control variables $\boldsymbol{d}_m^n$ and q_z^n. Here, the linearization of $\left[\boldsymbol{R}^{nT}, \boldsymbol{Q}^{nT}\right]^T$ w.r.t. $\boldsymbol{U}^n$ gives a symmetric global stiffness matrix $\boldsymbol{K}^n$ (Appendix C) which is independent of $\boldsymbol{U}^n$ and involves only current physical basis function values $\frac{\partial}{\partial \boldsymbol{x}}\hat{B}_m^{TP}(\boldsymbol{\xi}^g)$ and $\hat{A}_z^{TP}(\boldsymbol{\xi}^g)$ and physical quadrature weights $W^{g,n}$. Note that linearity of the discrete system is due to the explicit Lagrangian treatment of Eq. (67) and Eq. (71).

For contact problems the translation of Eq. (60) into weighted residual form and discretization gives the nodal contact force at a boundary $b = 1, 2, 3, 4$ of the patch

$$\boldsymbol{F}_{C,m}^n = -\sum_{g=1}^{n_b^{QP}}\left[\left(\kappa_P^{h,*} P^{h,n}\boldsymbol{n}^{h,n} + \kappa_R^{h,*}\dot{P}^{h,n}\boldsymbol{n}^{h,n} + \kappa_S^{h,*}\boldsymbol{v}^{h,n}\right)\hat{B}_m^{TP}\right]_{\boldsymbol{\xi}=\boldsymbol{\xi}^g} L^{g,n}, \tag{74}$$

where $P^{h,n}$ and $\dot{P}^{h,n} = \boldsymbol{v}^{h,n}\cdot\boldsymbol{n}^{h,n}$ are the discrete penetration and penetration rate; $\boldsymbol{n}^{h,n}$ is the discrete current outward normal unit vector; and numerical quadrature is applied over $g = 1, \ldots, n_b^{QP}$ boundary Gauss–Legendre points with weight $L^{g,n}$ along the respective physical boundary. We define the active penalty parameters, $\kappa_P^{h,*}$, $\kappa_R^{h,*}$, $\kappa_S^{h,*}$ in the discrete setting as

$$\kappa_P^{h,*} = \begin{cases} \kappa_P, & \text{for } P^h > 0, \\ 0, & \text{otherwise,} \end{cases} \qquad \kappa_R^{h,*} = \begin{cases} \kappa_R, & \text{for } \dot{P}^h > 0, \\ 0, & \text{otherwise,} \end{cases} \qquad \kappa_S^{h,*} = \begin{cases} \kappa_S, & \text{for } P^h > 0, \\ 0, & \text{otherwise.} \end{cases} \tag{75}$$

The nodal contact forces are added to the external force vector

$$\boldsymbol{F}^n_{ext,C,m} = \boldsymbol{F}^n_{ext,m} + \boldsymbol{F}^n_{C,m}, \tag{76}$$

leading to the nodal balance

$$\boldsymbol{F}^n_{int,m} = \boldsymbol{F}^n_{ext,C,m}. \tag{77}$$

The final global system becomes

$$\boldsymbol{R}^n_C(\boldsymbol{U}^n) = \boldsymbol{F}^n_{int}(\boldsymbol{U}^n) - \boldsymbol{F}^n_{ext,C}(\boldsymbol{U}^n) \overset{!}{=} \boldsymbol{0}, \tag{78}$$

supplemented with Eq. (73). It is nonlinear in $\boldsymbol{U}^n$ due to contact and we solve it with the Newton–Raphson method. The again symmetric global tangent matrix $\boldsymbol{K}^n_C(\boldsymbol{U}^n)$ is assembled from the nodal tangent matrix contributions as formulated in Appendix D.

Let us add the remark that, since we assume incompressible material behavior as per Eq. (65), it is important to properly select the discretization spaces for velocity and pressure to preclude effects of volumetric locking. Two important candidates are *(i)* the so-called *Taylor–Hood* pair where the pressure basis $\left\{\hat{A}\right\}_{z=1,\dots,Z}$ is of one polynomial degree less than the velocity basis (see [44]), and *(ii)* the construction of the velocity basis $\left\{\hat{B}\right\}_{m=1,\dots,M}$ as a subdivision of the pressure basis (see [45]). Due to the tensor product structure of the B-Spline bases, these concepts are applicable at the level of the constituent univariate bases.

The reviewed IGA procedure is conceptually straightforward. Unfortunately, due to the movement of the control points, the mapping of the parametrically static basis functions undergoes a distortion, recall Fig. 7. When extreme deformations take place, this may lead to complete failure of the analysis.

3.3. Enhanced FLIGA

We now outline a discretization method following IGA (Section 3.2) but with the floating tensor product B-Splines and the enhancements (improved quadrature, automated floating regulation, adaptive refinement) from Section 2. The most important modifications to the IGA discretization described in Section 3.2 are the following:

- Enrich the approximation bases by floating regulation points (Sections 2.1–2.3).
- Use linear polynomial basis functions in η (Section 2.4).
- Place all quadrature points at the characteristic element boundaries along ξ (Section 2.4.1).
- Update floating regulation points after regular time step increments (Section 2.4.2).
- Introduce an adaptive local refinement procedure along ξ (Section 2.4.3).

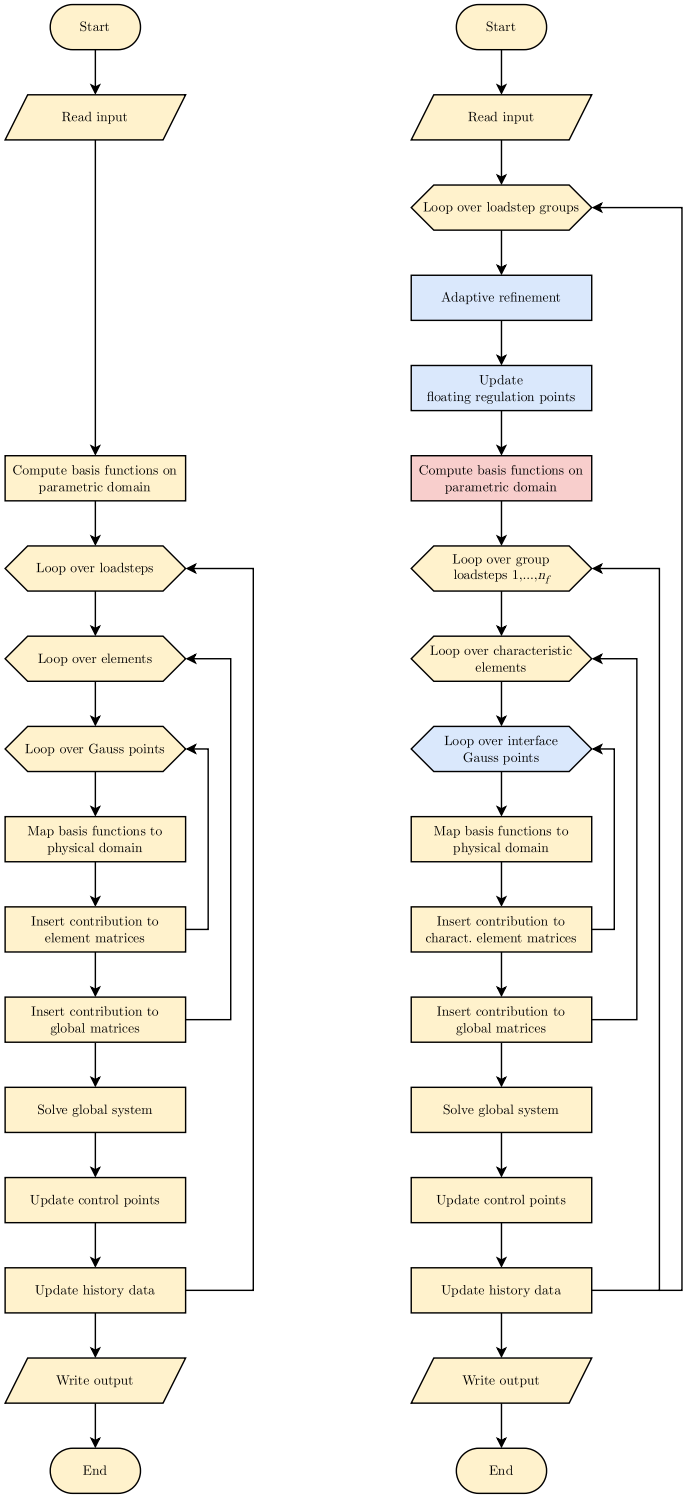
The governing equations are easily transferred to FLIGA by replacing

$$\begin{aligned} \hat{B}^{TP}_m &\leftarrow \hat{B}^n_m \\ \boldsymbol{\xi}^g &\leftarrow \boldsymbol{\xi}^{gl,n}_s \\ W^{g,n} &\leftarrow W^{gl,n}_s, \end{aligned} \tag{79}$$

in Eq. (72) or Eq. (78), respectively, and in Eq. (73). Here we need the current floating B-Splines $\hat{B}^n_m$, novel quadrature points $\boldsymbol{\xi}^{gl,n}_s$ and weights $W^{gl,n}_s$.

Further, we propose the construction of stable mixed approximation spaces in enhanced FLIGA following the subdivision strategy in the normal direction, as well as the Taylor–Hood or subdivision approach along the characteristic direction. Subdivision in normal direction is preferred as we have $q = 1$ and, by nesting two velocity spans within one pressure span, we can conveniently reuse the floating maps $\mathcal{G}_{j=1,3,5,\dots,J}$ applied to the velocity basis for floating regulation of the pressure basis. This facilitates the overall procedure as we avoid the introduction of additional floating regulation points for pressure, and we can even use the same quadrature coordinates to compute all velocity and pressure contributions. The concepts for combining the sets of univariate bases to a bivariate floating basis (Section 2.1) are then correspondingly employed. (As in Section 3.2, we apply the physical map $\mathcal{F}$ for the velocity basis also to the pressure basis.)

In Fig. 15 we schematically overview the procedures followed by IGA (a) and enhanced FLIGA (b). The introduction of floating regulation points in FLIGA mainly leads to modification of the basis computation in



a) Isogeometric analysis b) Enhanced FLIGA

Fig. 15. Algorithm flow chart comparison. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

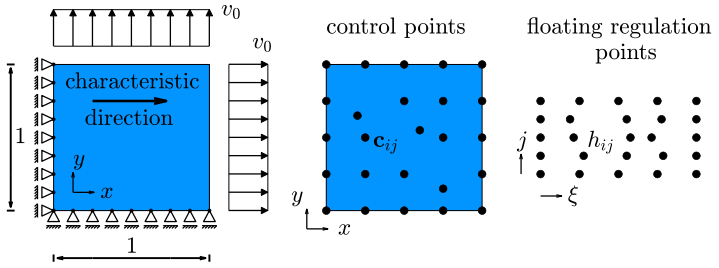


Fig. 16. Patch test problem.

parametric coordinates (red block). Moreover, the operations for the three blocks highlighted in blue correspond to the three enhancements of this work. Let us add a remark on optimizing the computational cost. In the flow chart, we group a certain number of time steps and perform the new "floating" operations on the parametric bases only after each group, i.e., say, every $n_f = 5$ time steps. During the 5 time steps of each group, the mesh distorts (slightly), which is then "relaxed" after the subsequent floating operations. Groups of 5+ time steps are often unproblematic (this obviously also depends on the choice of the time step size) and quadrature points remain fully Lagrangian. Following this strategy the increase of average cost per load step introduced by the floating procedures is quite limited, for more details see Section 4.2.

4. Numerical examples

In this section, we conduct a numerical study of enhanced FLIGA. The patch test, Taylor–Couette flow, planar extrusion and a small set of extrusion-based AM processes serve as examples. Where possible, we directly compare results to those of the first version of FLIGA proposed in [5], denoted as FLIGA 1.0 in the following. Accordingly, we denote enhanced FLIGA also as FLIGA 2.0.

Numerical approximations are assessed in terms of the logarithmic L^2 error norm

$$L^2\left(v^h\right) = \log_{10}\left(\frac{\sqrt{\int_{\Omega^h}\left(v - v^h\right)^2 \,\mathrm{d}\Omega}}{\sqrt{\int_{\Omega^h} v^2 \,\mathrm{d}\Omega}}\right), \tag{80}$$

where v and v^h are components of the analytical and numerical solutions, respectively.

4.1. Patch test

We start with a two-dimensional patch test, similar to the one in [5]. Here we consider a square domain occupied by Newtonian fluid ($\eta_s = 50$, $\eta_p = 0$) subjected to Dirichlet boundary conditions in the normal directions as shown in Fig. 16. The imposed velocity is chosen as $v_0 = 1$. The pressure is set to $p = 0$ and the incompressibility constraint is removed, so as to enable a simple volumetric expansion of the domain with analytical solution

$$\boldsymbol{v} = \begin{pmatrix} v_x \\ v_y \end{pmatrix} = \begin{pmatrix} x \\ y \end{pmatrix}, \tag{81}$$

with a gradient of

$$\frac{\partial \boldsymbol{v}}{\partial \boldsymbol{x}} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}. \tag{82}$$

Hence, the velocity component in x (in y) has a constant gradient pointing in the x (in the y) direction.

In order to enable an easier comparison with Hille et al. [5], we adopt the control and floating regulation points employed there and the automated placement procedure on the floating regulation points introduced in Section 2.4.2 is not applied. We consider three test cases $p = 1, 2, 3$ (while, as always, $q = 1$). Note that due to the fixed number of control points, the open uniform parent knot vectors $\tilde{\Xi}_j = \mathcal{K}\left(n_p^S, p\right)$ are different for each p (as we have the knot span counts $n_1^S = 4, n_2^S = 3, n_3^S = 2$), and we construct the quadrature knot vectors $\tilde{\Xi}^l = \mathcal{K}\left(2n_p^S, p\right)$ to be

Table 1
L^2 error in $\log_{10}$-scale for the patch test. The bracketed values are obtained for quadrature knot spans *eight* times smaller than the parent knot spans.

Polynomial order	Velocity component	$p=1$ $q=1$	$p=2$ $q=2$	$p=3$ $q=3$
IGA	x	−16.28	−15.67	−15.73
	y	−16.02	−15.85	−15.63
FLIGA 1.0	x (characteristic)	−1.93	−2.46	−2.88
	y (normal)	−2.86	−3.78	−3.94
Polynomial order		$p=1$ $q=1$	$p=2$ $q=1$	$p=3$ $q=1$
FLIGA 2.0	x (characteristic)	−3.79 (−4.73)	−4.80 (−6.67)	−7.13 (−8.50)
	y (normal)	−3.04 (−4.32)	−4.45 (−5.95)	−6.18 (−8.11)

twice as dense. Each knot span of any $\tilde{\Xi}^l$ collects $G = p+1$ quadrature points and a total of roughly $n^{QP} = 100$ quadrature points is obtained for each p, allowing for a fair comparison to the FLIGA 1.0 patch test, where we employed a similar number.

An overview of the patch test results for the L^2 errors Eq. (80) w.r.t. the analytical velocity components Eq. (81) is provided in Table 1, where we also reproduce the errors obtained previously for IGA and FLIGA 1.0, however with $q = p$ [5]. As expected, IGA passes the patch test up to machine precision. With FLIGA 1.0 we had observed a limited accuracy for the small number of integration points. Conversely, for FLIGA 2.0 the patch test reveals distinctive reductions of the error by employment of the novel quadrature stencil. The improvements are most pronounced for higher polynomial orders and solution field gradients along the characteristic direction (corresponding to the solution component v_x^h, see Eq. (82)). The numerical error further decreases as the knot span density in $\tilde{\Xi}^l$ is increased. E.g., the values in brackets in Table 1 are obtained for $\tilde{\Xi}^l = \mathcal{K}\left(8n_p^S, p\right)$. This error reduction is possible up to machine precision due to the first-order consistency property of floating B-Splines.

4.2. Taylor-Couette flow

Next, we revisit the two-dimensional problem of Taylor–Couette flow for Newtonian as well as viscoelastic Oldroyd-B fluid behavior, which we investigated with FLIGA 1.0 in [5]. We represent the annulus of fluid by a single patch, treating the circumferential direction as the characteristic one and employing a periodic boundary condition to close the ring structure.[5] While with FLIGA 1.0 we had $q \geq 1$, here $q = 1$, thus we increase the number of elements in normal direction in order to retain a comparable accuracy. Quadrature is performed by the novel procedure constructing $\tilde{\Xi}^l = \mathcal{K}\left(2n^S, p\right)$ twice as dense as $\tilde{\Xi}_j = \mathcal{K}\left(n^S, p\right)$ with $n^S = 36$. For the study of the Oldroyd-B model, we double the time step size w.r.t. Hille et al. [5]. In addition, we reduce the computational cost by applying the floating operations only every $n_f = 20$ time steps. All other parameters are given in Table 2.

We compute the L^2 errors Eq. (80) of horizontal velocity and pressure for different rotation states of the outer cylinder (with the inner one fixed). The analytical (steady-state) solutions are

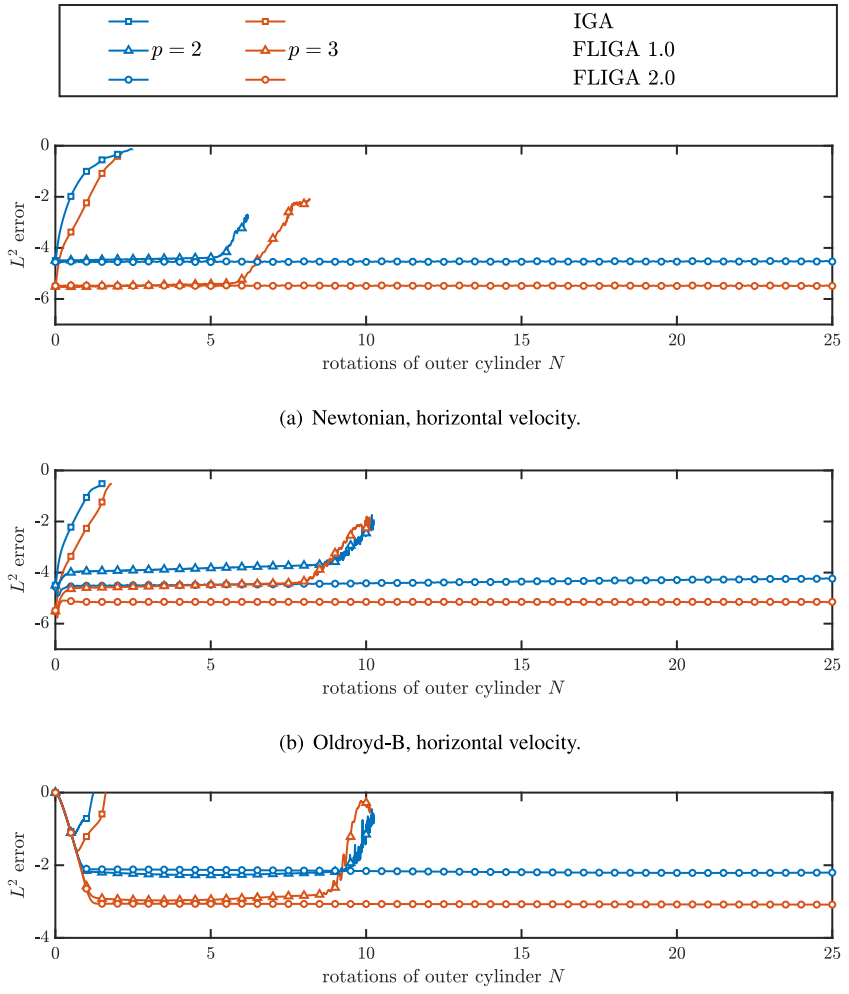
$$v_{x,Newt} = v_{x,Old} = \sin(\varphi)\left(\alpha r + \frac{\beta}{r}\right), \qquad p_{Newt} = 0, \qquad p_{Old} = 2\pi\beta^2\lambda\left(\frac{1}{r^4} - \frac{1}{R_O^4}\right), \tag{83}$$

with

$$\alpha = \Omega_O \cdot \frac{R_O^2}{R_O^2 - R_I^2}, \qquad \beta = -\alpha R_I^2, \tag{84}$$

and the polar coordinates $r = \sqrt{x^2+y^2}$, $\varphi = \operatorname{atan}\left(\frac{y}{x}\right)$. Results are provided in Fig. 17 and demonstrate the significantly improved stability of enhanced FLIGA over FLIGA 1.0. We emphasize that adaptive refinement is

[5] In [5] we provide the details about how to construct periodic floating B-Spline bases.



(a) Newtonian, horizontal velocity.

(b) Oldroyd-B, horizontal velocity.

(c) Oldroyd-B, pressure.

Fig. 17. Transient L^2 errors in $\log_{10}$-scale w.r.t. the analytical steady-state solutions for the Taylor–Couette problem.

not applied here and that very similar results are also obtained when using the old manual procedure for updating floating regulation points (instead of the new automated one). Consequently, we believe that the improvements are due to the novel quadrature concept.

We study the computation cost for enhanced FLIGA. To this end, let us consider the case $p = 2,\ q = 1$ with spatial discretization according to Table 2 and viscoelastic behavior. Different floating update intervals are tested, $n_f \in \{1, 2, 4, 8, 16, 32, 64, 128, 256, 512, 1024\}$. For each n_f we measure the computation time needed to process the first 1024 analysis load steps for the time discretization $\Delta t = 1\text{e}{-5}$ s. Additionally, we test the two cases of, one, quadrature knot vectors identical to the parent knot vectors, $\tilde{\Xi}^l = \mathcal{K}\left(n_p^S, p\right)$, and two, twice as dense, $\tilde{\Xi}^l = \mathcal{K}\left(2n_p^S, p\right)$; and it holds $G = 3$. All tests are performed in fivefold replication. We compare the computation times relative to the one of IGA with identical discretization, except that classical Gauss–Legendre quadrature (3×2 points per velocity element) is employed (and no floating operations are performed).

Results for the relative computation cost are provided in Fig. 18. We observe a decreasing trend of the cost for increasing floating update intervals n_f. For the case $\tilde{\Xi}^l = \mathcal{K}\left(n_p^S, p\right)$ the total quadrature point count in enhanced FLIGA and IGA is identical, and the cost converges towards 1 in the limit of no floating operations being applied. Conversely, for the case $\tilde{\Xi}^l = \mathcal{K}\left(2n_p^S, p\right)$ some extra cost remains in that limit due to a larger quadrature point set. We confirm that a suitable adjustment of n_f notably reduces the cost overhead due to the floating operations

Table 2
Parameters for the Taylor–Couette problem. Value pairs in braces refer to $p = 2$ and $p = 3$, respectively.

Problem parameter	Variable	Value	Unit	Simulation parameter	Variable	Value	Unit
Radius				Polynomial order			
Inner cylinder	R_I	0.1	m	Characteristic axis	p	$\{2, 3\}$	
Outer cylinder	R_O	0.2	m	Normal axis	q	1	
Angular frequency				Element count			
Inner cylinder	Ω_I	0	rad/s	Velocity		$36 \times \{24, 72\}$	
Outer cylinder	Ω_O	7.5	rad/s	Pressure		$18 \times \{12, 36\}$	
Relaxation time	λ	0.1	s	Time step Newtonian	Δt_{Newt}	5e−5	s
Polymeric viscosity	η_p	1.5	Pa s	Time step Oldroyd-B	Δt_{Old}	$\{5e-5, 1e-5\}$	s
Solvent viscosity	η_s	0.5	Pa s	Floating update interval	n_f	20	
Polymer stress at $t = 0$	$\boldsymbol{\tau}_p^0$	$\mathbf{0}$	Pa				

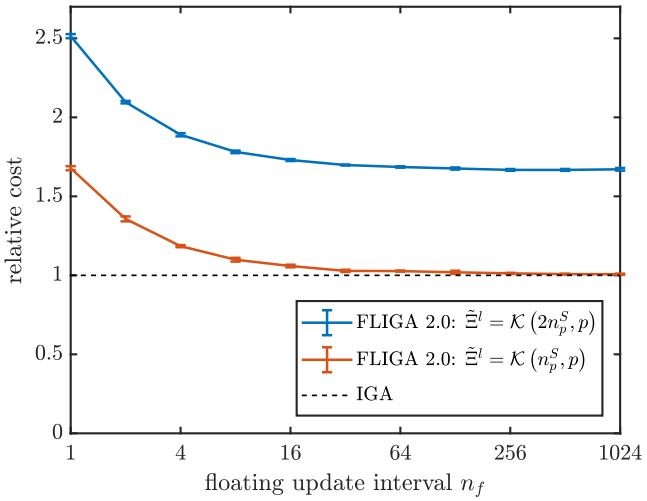


Fig. 18. Computation cost for $p = 2$, $q = 1$ with standard deviations.

(e.g., in the previous computations we had $n_f = 20$). Based on the current study we conclude that the majority of such overhead is avoided, even with smaller floating update intervals; as a rule of thumb, $n_f \approx 5$.

4.3. Planar extrusion

The problem of planar extrusion from a straight nozzle is investigated. We adopt as reference the computational fluid dynamics (CFD) steady-state study in [46]. Due to the Lagrangian treatment in our work, we modify the problem setup as follows (Fig. 19). To circumvent the inflow condition, a convergent segment (4 : 1) is attached to the nozzle and fluid motion is driven by a Dirichlet boundary condition ($v_{in} = 0.5$ mm/s); the length of the straight part is much longer ($l = 10$ mm) compared to the radius ($r_{out} = 0.2$ mm); the strict contact and no-slip conditions are relaxed by penalty enforcement; the penalty coefficients are set as $\kappa_P = 7.5e5$, $\kappa_R = 1.5e3$, $\kappa_S = 1e3$, respectively, with a ramp ($0 \rightarrow 1e3$) on κ_S in the region indicated in the figure; a small radius $r_e = \frac{r_o}{8}$ is added at the nozzle exit; all discretization parameters are listed in Table 3. Note that, in our setting, we are investigating a transient problem.

We focus on the Oldroyd-B model with solvent viscosity ratio

$$\beta = \frac{\eta_s}{\eta_s + \eta_p} = \frac{1}{9}, \tag{85}$$

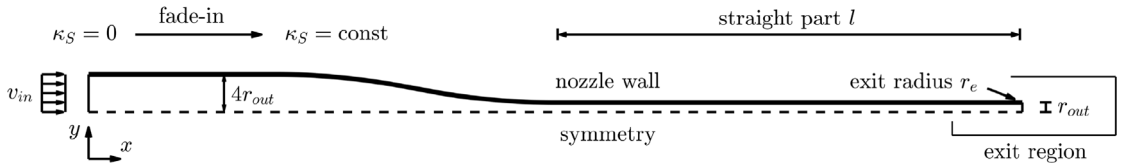


Fig. 19. Planar extrusion problem.

Table 3
Simulation parameters for the planar extrusion problem.

Simulation parameter	Variable	Value	Simulation parameter	Variable	Value
Polynomial order			Element count		
Characteristic axis			Velocity		(adaptive) × 16
Velocity	p	2	Pressure		(adaptive) × 8
Pressure	p	1	Quadrature point count		(adapt. initialized)
Normal axis			Per initial element		× $(q+1)$
Velocity	q	1	Time step	Δt	5e−4 s
Pressure	q	1	Floating update interval	n_f	10

obtained by the choice $\eta_s = 1000$ Pa s, $\eta_p = 8000$ Pa s. The relaxation time is adjusted so as to produce a given Weissenberg number Wi according to

$$\lambda = \mathrm{Wi} \cdot \frac{1}{\dot{\gamma}} = \mathrm{Wi} \cdot \frac{r_{out}}{12 v_{in}}, \tag{86}$$

where $\dot{\gamma}$ is a maximum shear rate estimate in the straight part of the nozzle. The initial condition for the polymer stress is $\boldsymbol{\tau}_p^0 = \mathbf{0}$.

In Fig. 20(a–d) we visualize for $Wi = 2.5$ the build-up of the free surface behind the nozzle exit at different time steps, which is controlled by viscoelastic swelling. Contours of vertical velocities in the nozzle exit area concentrate close to the wall tip. After $t = 7.5$ s, steady-state behavior is reached in the region, see Fig. 20(e). Moreover, Fig. 21(a) shows the velocity magnitude contours during steady state at $t = 7.5$ s, indicating a parabolic-type velocity profile inside the nozzle as well as a transition towards plug profile behind the exit. Our qualitative observations agree well with the description in [46]. The corresponding control point set is illustrated in Fig. 21(b) and adaptive refinement is recognized from the nodal distances. The *a priori* chosen thresholds for knot insertion Eq. (48) and removal Eq. (50) are illustrated in Fig. 22.

Finally, we provide a quantitative comparison to Comminal et al. [46] for the swell ratios

$$S = \frac{r_s}{r_{out}}, \tag{87}$$

at different $Wi \in \{0, 0.5, 1, 1.5, 2, 2.5\}$.[6] Here r_s is the steady-state extrudate radius, see Fig. 20(e). Results are depicted in Fig. 23 and compared to the results obtained in [46] using a selection of CFD techniques. In view of the deviations between results of the different CFD techniques, our results show very satisfactory agreement for high Wi. The small underestimation of swell ratios for (fluid-like) low Wi may due to our approximate enforcement of the no-slip condition at the wall.

4.4. Extrusion-based AM

Lastly, we apply enhanced FLIGA to the simulation of extrusion-based AM. The problem setup is adopted from the previous benchmark of planar extrusion, however with the following modifications: the complete setup is rotated such that the nozzle opens in downwards vertical direction; the symmetry assumption is removed and the entire plane represented; the lengths of the contraction segment and of the straight part ($l = 0.2$ mm) are decreased; the nozzle exit radius is increased ($r_e = r_o/4$); and the penalty coefficients are adjusted ($\kappa_P = 1\text{e}4$, $\kappa_R = 1\text{e}3$,

[6] For the Newtonian fluid, $Wi = 0$, we adjust $\kappa_S = 250$.

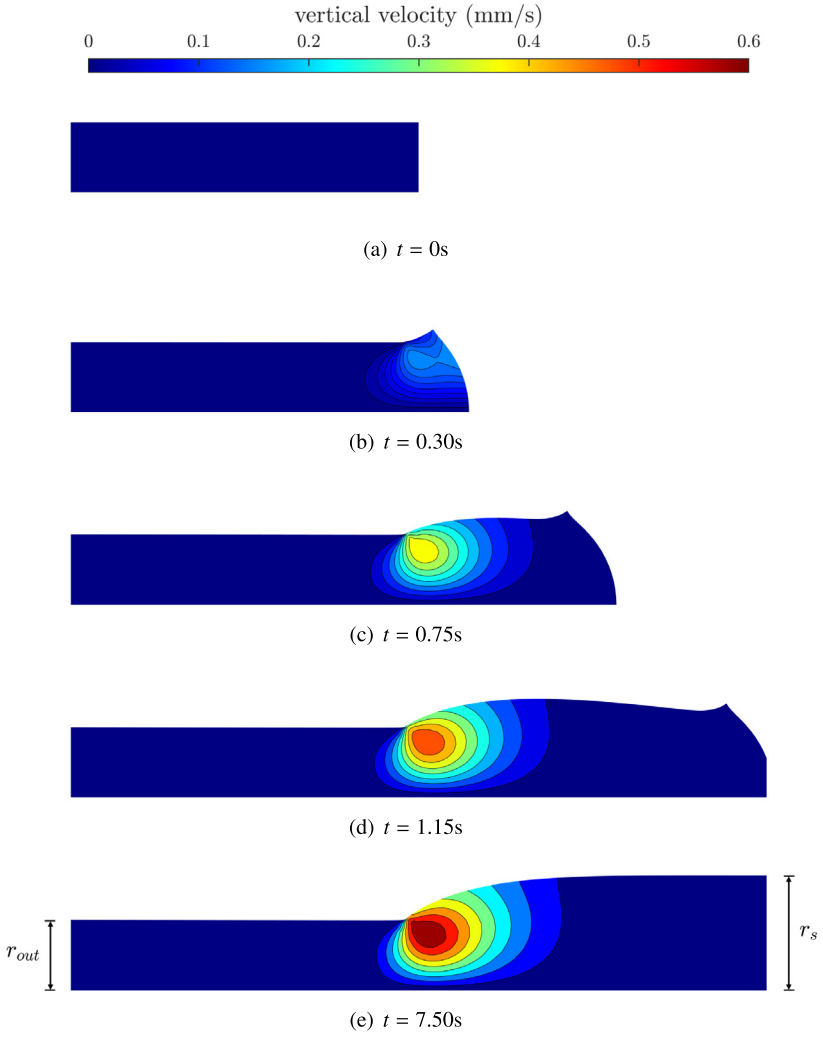
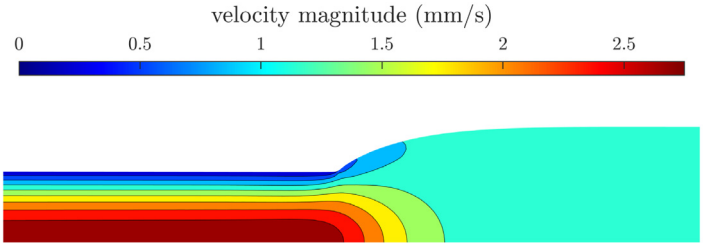


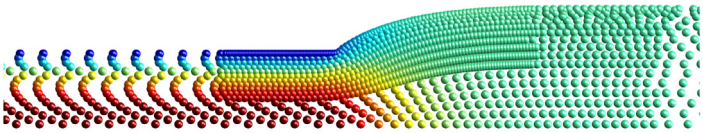
Fig. 20. Vertical velocity contours during transient strand formation (a–d) and steady-state (e) in the exit region.

$\kappa_S = 100$). In addition, we model a substrate below the nozzle exit by imposing homogeneous Dirichlet boundary conditions for the vertical velocity on all control points penetrating a user-defined substrate shape. We simulate the horizontal movement of the nozzle by imposing a constant Dirichlet boundary condition for the horizontal velocity ($v_n = 2.4$ mm/s) to the control points in contact with the substrate. We proceed similarly to simulate a vertical movement of the nozzle. For simplicity, a renewed material detachment from the substrate is not considered here. The discretization parameters are summarized in Table 4.

We investigate the viscoelastic Oldroyd-B material with the same parameters as in the planar extrusion example for the case $Wi = 1.5$. Four different combinations of substrate shape and nozzle movement are tested, see Fig. 24. The first case (a) corresponds to an idealized setup with mere horizontal nozzle movement along a linear path and deposition on a planar substrate. Secondly (b), we investigate an imperfection of the nozzle movement in terms of a superposed harmonic vibration. The third case (c) explores substrate imperfections in terms of surface variations. And finally (d), we put an obstacle of parabolic shape and adjust the nozzle path. In all examples, the center of the nozzle exit starts at the left black circles in the figure, following the dashed path until the final state, represented by the black circles on the right. Also, we add a small step to the front segment of the substrates so as to avoid a singularity in the physical map of the rectangular parametric domain at the respective corner. The characteristic element division in the undeformed state is illustrated by the white curves in Fig. 24. Finally, the



(a) Field contours



(b) Control points

Fig. 21. Velocity magnitude during steady state ($t = 7.50$ s) in the exit region.

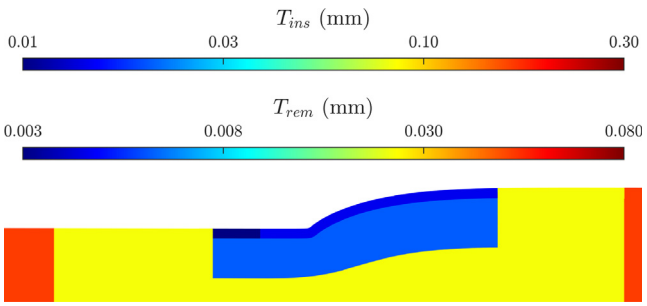


Fig. 22. Refinement thresholds.

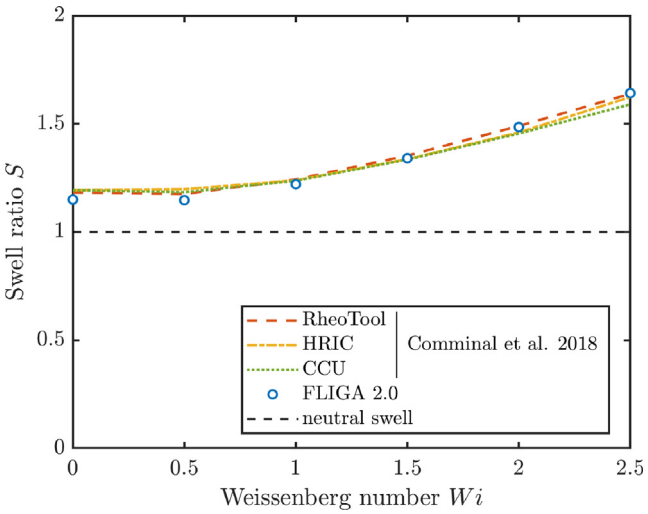


Fig. 23. Die swell ratios.

Table 4
Simulation parameters for the extrusion-based AM problem.

Simulation parameter	Variable	Value	Simulation parameter	Variable	Value
Polynomial order			Element count		
Characteristic axis			Velocity		(adaptive) × 16
Velocity	p	2	Pressure		(adaptive) × 8
Pressure	p	1	Quadrature point count		(adapt. initialized)
Normal axis			Per initial element		× $(q+1)$
Velocity	q	1	Time step	Δt	$1.\overline{6}$e−3 s
Pressure	q	1	Floating update interval	n_f	5

predefined thresholds for knot insertion and removal used in the present study are depicted in Fig. 25. Note that the threshold definitions inside of the black box are always in the same reference to the nozzle position.

Simulation results are illustrated in Fig. 26. For the first printing scenario (a), the characteristic element interfaces (white curves) align evenly along the extrusion strand. A vibration of the nozzle (b) causes deviations in the alignment leading to a periodic, wavy pattern of material arrangement. For the uneven substrate (c), the extrudate adapts to the substrate shape. Interestingly, the effect is local and almost completely fades out up to the top edge of the strand, resulting in an even surface. The nozzle path we have designed for case (d) allows to overcome the obstacle with the material being pressed also in the two corners. For all examples (a–d), we show the vertical velocity magnitudes at the final configuration by contour plots. The approximately enforced no-slip condition is recognized from the blue colors at the nozzle walls, indicating low material speed. This is an important enhancement with respect to the extrusion-based AM problem investigated for FLIGA 1.0 in [5], where the resolution of the resulting extreme elongation rates behind the nozzle exit would not have been possible due to the lack of adaptive refinement.

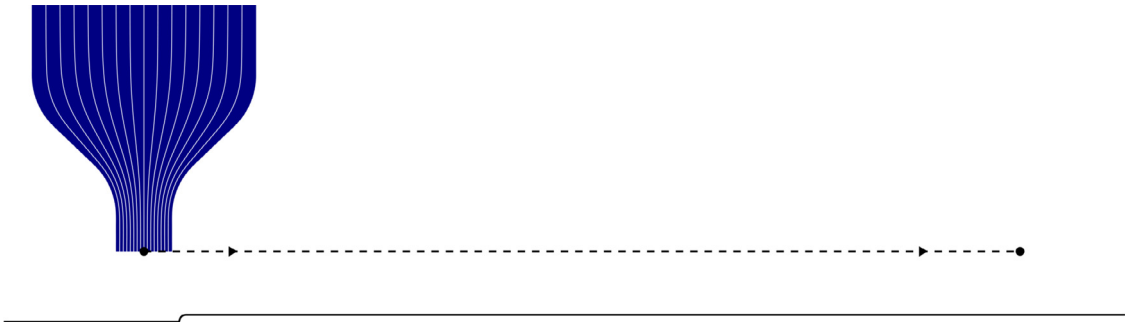
We confirm a good overall stability and robustness of enhanced FLIGA limited to the validity of the assumption from Section 2.4.2, i.e., that inner characteristic element interfaces reasonably align to the outer patch boundaries. From the element topology in the above AM test cases, we see the assumption fulfilled even when modeling moderate process imperfections.

5. Conclusions

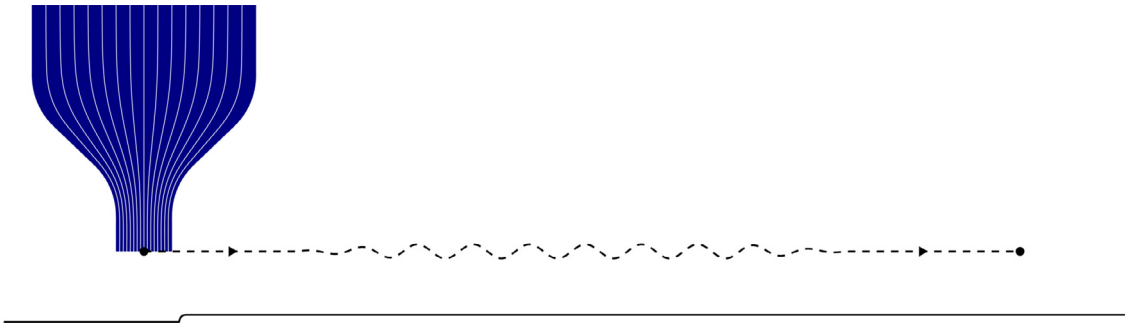
In this paper, we have proposed enhancements for Floating Isogeometric Analysis with respect to the initial formulation in [5]. A new quadrature concept conserves the material point character of quadrature points, while reducing the relative errors in the patch test up to four orders of magnitude and improving the range of numerical stability in the Taylor–Couette benchmark from five to more than 25 turns. The novel procedure for floating regulation removes shear mesh distortion along the characteristic direction in a fully automatic fashion and allows for the prediction of complicated strand geometries in extrusion-based additive manufacturing simulation, where the manually designed floating regulation of Hille et al. [5] would have been prohibitively complicated. Moreover, dilatational mesh distortion is overcome by adaptive refinement, enabling the solution of extrusion problems featuring extremely extensional flow.

While offering an effective solution to extreme mesh distortion along the characteristic direction, enhanced FLIGA shares many discretization concepts with Lagrangian IGA, as well as many of its benefits. These include excellent stability, direct imposition of boundary conditions, easy construction of stable mixed ansatz spaces, adjustable smoothness of the basis functions (in the characteristic direction), absence of advection terms, and natural treatment of free surface evolution. On the other hand, there are still some limitations. It is not possible to apply the strategy to problems with critical mesh distortion along the normal direction. The conceptual generalization to multi-patch geometries appears limited to the separate floating within each patch coming along with a distortion of patch interfaces. Finally, the polynomial order is fixed to $q = 1$ in the normal direction. These limitations may well be addressed in future investigations. In addition, it would be useful to thoroughly investigate stability and accuracy from a numerical analysis perspective. Also, while we focused on spatial discretization, further investigations may consider the aspect of time discretization in more detail. Finally, further significant applications of FLIGA beyond viscoelastic extrusion are worth exploring.

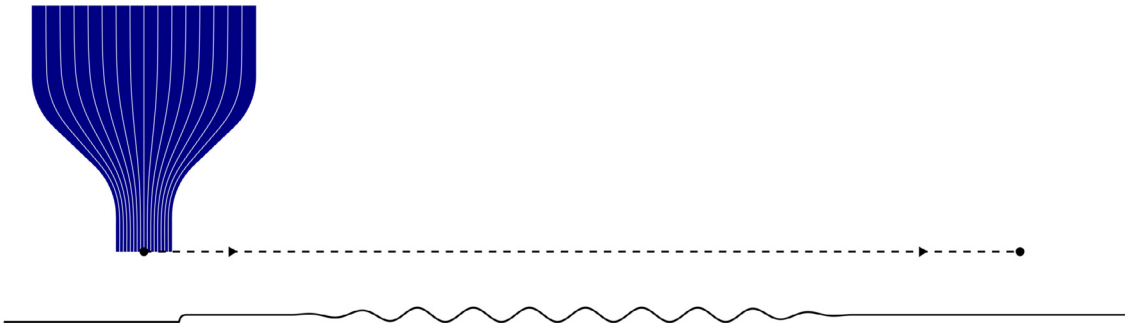
The extension to 3D appears interesting for the application to industrially relevant problems. This would require the introduction of a third parametric direction ζ, i.e. another *normal* direction (again with polynomial order $r = 1$).



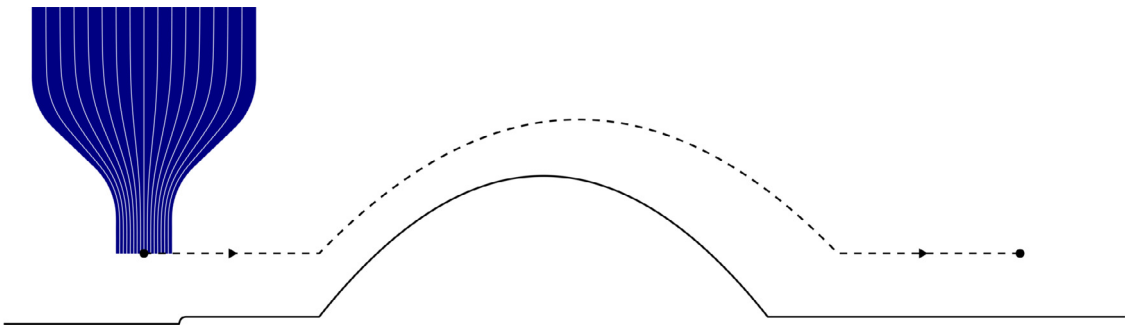
(a) Straight deposition



(b) Nozzle vibration



(c) Uneven substrate



(d) Obstacle

Fig. 24. Extrusion-based AM problems.

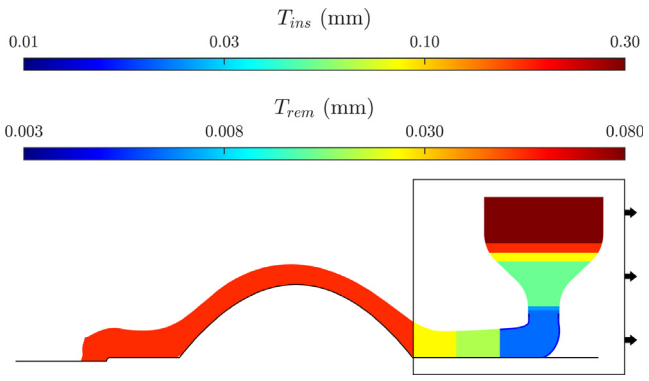


Fig. 25. Refinement thresholds.

Gauss–Lobatto points in the normal directions are again to be placed at the Kronecker-delta positions of the normal basis functions $\hat{M}_j$ in η and $\hat{L}_k$ in ζ. The floating tensor product structure would yield the trivariate basis by coupling the two normal directions with the unique characteristic direction

$$\hat{B}_{ijk}\left(\boldsymbol{\xi};\mathcal{H}\right)=\hat{N}_{ijk}\left(\xi;\mathcal{H}\right)\hat{M}_j\left(\eta\right)\hat{L}_k\left(\zeta\right)\qquad i=1,\ldots,I_j;\ j=1,\ldots,J;\ k=1,\ldots,K. \tag{88}$$

where now $\boldsymbol{\xi}=(\xi,\eta,\zeta)^T$. Note that this equation is the 3D version of Eq. (8). Regarding the implementation, we do not foresee specific problems from our current viewpoint. Along the characteristic direction, mesh distortion is overcome in the same sense as so far for 2D simulations. However, depending on the deformations, the bivariate normal elements might undergo a distortion in physical space. E.g., in the simulation of extrusion processes, small distances between nozzle and substrate may lead to severe cross-sectional strand deformations as was shown in [47]. In such cases, the associated normal element distortion may possibly affect the numerics and require additional remedies.

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.

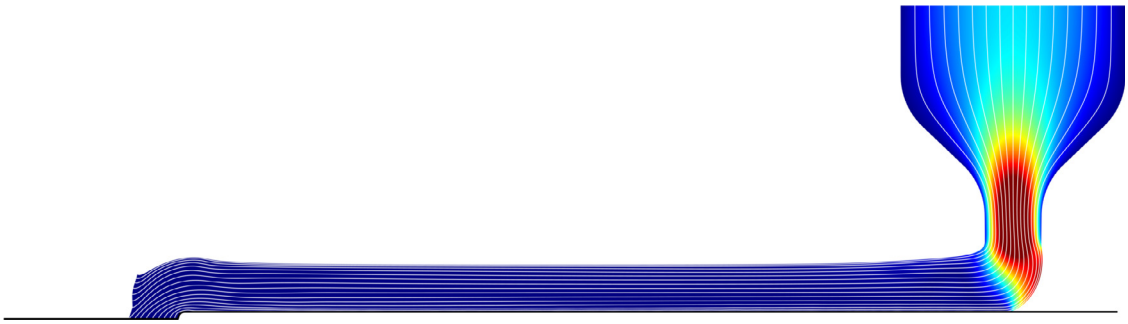
Data availability

Data will be made available on request.

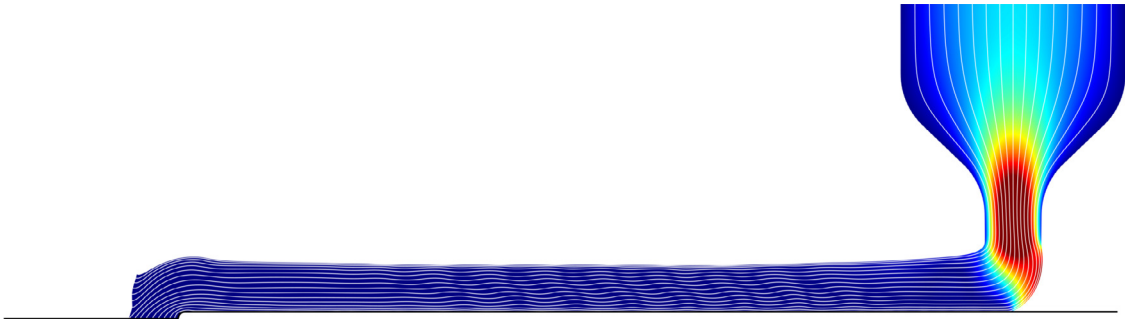
Appendix A. Parametrization ansatz

The exact representation of ξ by the ansatz in Eq. (42) is shown as

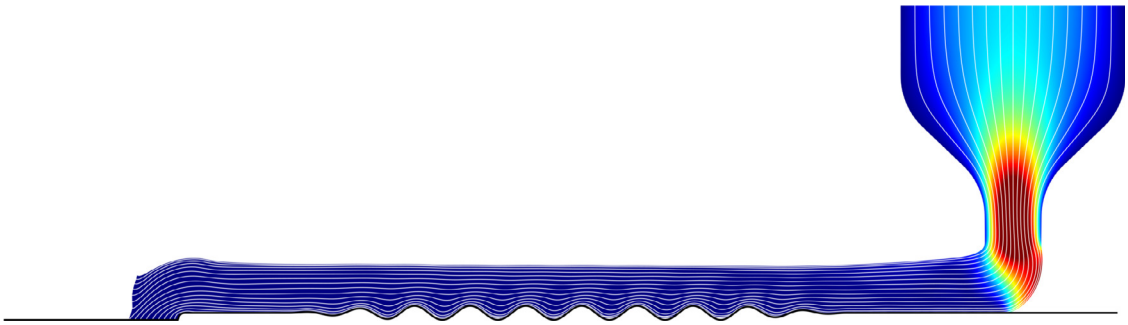
$$\begin{aligned}\xi^h\left(\boldsymbol{x};\mathcal{H}\right)&=\sum_{m=1}^{M}h_m B_m\left(\boldsymbol{x};\mathcal{H}\right)=\sum_{m=1}^{M}h_m\hat{B}_m\left(\mathcal{F}^{-1}\left(\boldsymbol{x};\mathcal{H}\right);\mathcal{H}\right)=\sum_{m=1}^{M}h_m\hat{B}_m\left(\boldsymbol{\xi}\left(\bullet;\mathcal{H}\right);\mathcal{H}\right)\\&=\sum_{j=1}^{J}\left(\sum_{i=1}^{I_j}h_{ij}\hat{N}_{ij}\left(\xi\left(\bullet;\mathcal{H}\right);\mathcal{H}\right)\right)\hat{M}_j(\eta\left(\bullet;\mathcal{H}\right))\\&=\sum_{j=1}^{J}\left(\sum_{i=1}^{I_j}h_{ij}\tilde{N}_{ij}\left(\mathcal{G}_j^{-1}\left(\xi\left(\bullet;\mathcal{H}\right);\mathcal{H}\right)\right)\right)\hat{M}_j(\eta\left(\bullet;\mathcal{H}\right))\\&=\sum_{j=1}^{J}\mathcal{G}_j\left(\mathcal{G}_j^{-1}\left(\xi\left(\bullet;\mathcal{H}\right);\mathcal{H}\right);\mathcal{H}\right)\hat{M}_j(\eta\left(\bullet;\mathcal{H}\right))\end{aligned}$$



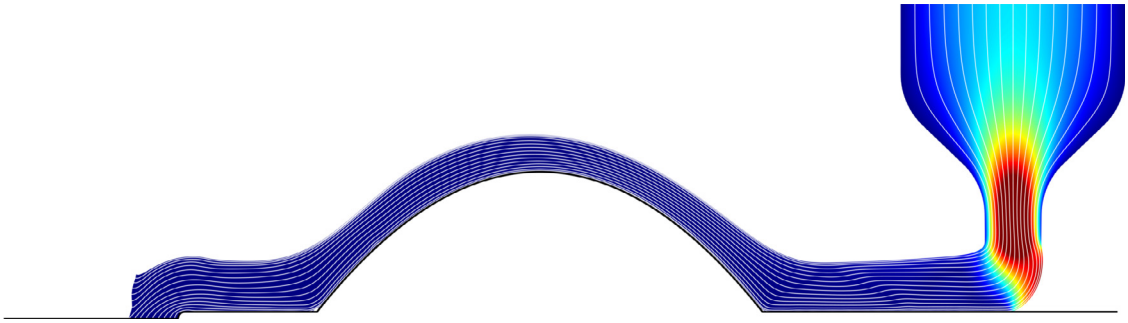
(a) Straight deposition



(b) Nozzle vibration



(c) Uneven substrate



(d) Obstacle

Fig. 26. Final strand geometries. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

$$= \sum_{j=1}^{J} \xi\left(\bullet;\mathcal{H}\right) \hat{M}_j(\eta\left(\bullet;\mathcal{H}\right)) = \xi\left(\bullet;\mathcal{H}\right) \sum_{j=1}^{J} \hat{M}_j(\eta\left(\bullet;\mathcal{H}\right))$$
$$= \xi\left(\bullet;\mathcal{H}\right).$$

Appendix B. Tangent matrix for solving the parametrization equation

Let us denote

$$\boldsymbol{J}_{sn}^{gl}\left(\bullet;\mathcal{H}\right) := \boldsymbol{J}\left(\tilde{\boldsymbol{\xi}}_{sn}^{gl}\left(\bullet;\mathcal{H}\right);\mathcal{H}\right).$$

Then the global tangent to Eq. (46) is given as

$$\begin{aligned}
K_{ik}^{R} = \frac{\partial}{\partial h_k} R_i = \sum_{l=1}^{2J-2} \sum_{g=1}^{n^{QP,l}} \sum_{m=1}^{M} \sum_{a=1}^{2} \Bigg[& \frac{\partial^2}{\partial h_k \partial x_a} \hat{B}_i\left(\tilde{\boldsymbol{\xi}}_{sn}^{gl}\left(\bullet;\mathcal{H}\right);\mathcal{H}\right) \;\; \frac{\partial}{\partial x_a} \hat{B}_m\left(\tilde{\boldsymbol{\xi}}_{sn}^{gl}\left(\bullet;\mathcal{H}\right);\mathcal{H}\right) \;\; h_m \;\; W^{gl}\left(\bullet;\mathcal{H}\right) \\
& + \frac{\partial}{\partial x_a} \hat{B}_i\left(\tilde{\boldsymbol{\xi}}_{sn}^{gl}\left(\bullet;\mathcal{H}\right);\mathcal{H}\right) \;\; \frac{\partial^2}{\partial h_k \partial x_a} \hat{B}_m\left(\tilde{\boldsymbol{\xi}}_{sn}^{gl}\left(\bullet;\mathcal{H}\right);\mathcal{H}\right) \;\; h_m \;\; W^{gl}\left(\bullet;\mathcal{H}\right) \\
& + \frac{\partial}{\partial x_a} \hat{B}_i\left(\tilde{\boldsymbol{\xi}}_{sn}^{gl}\left(\bullet;\mathcal{H}\right);\mathcal{H}\right) \;\; \frac{\partial}{\partial x_a} \hat{B}_m\left(\tilde{\boldsymbol{\xi}}_{sn}^{gl}\left(\bullet;\mathcal{H}\right);\mathcal{H}\right) \;\; \delta_{mk} \;\; W^{gl}\left(\bullet;\mathcal{H}\right) \\
& + \frac{\partial}{\partial x_a} \hat{B}_i\left(\tilde{\boldsymbol{\xi}}_{sn}^{gl}\left(\bullet;\mathcal{H}\right);\mathcal{H}\right) \;\; \frac{\partial}{\partial x_a} \hat{B}_m\left(\tilde{\boldsymbol{\xi}}_{sn}^{gl}\left(\bullet;\mathcal{H}\right);\mathcal{H}\right) \;\; h_m \;\; \frac{\partial}{\partial h_k} W^{gl}\left(\bullet;\mathcal{H}\right) \Bigg],
\end{aligned}$$

$$\begin{aligned}
\frac{\partial^2}{\partial h_k \partial x_a} \hat{B}_i\left(\tilde{\boldsymbol{\xi}}_{sn}^{gl}\left(\bullet;\mathcal{H}\right);\mathcal{H}\right) = \Bigg[& -\sum_{o=1}^{M} \sum_{b=1}^{2} \sum_{p=1}^{2} \sum_{q=1}^{2} \left(J_{sn}^{gl}\right)_{bp}^{-1}\left(\bullet;\mathcal{H}\right) \;\; \left(J_{sn}^{gl}\right)_{qa}^{-1}\left(\bullet;\mathcal{H}\right) \;\; c_{op} \;\; \frac{\partial}{\partial \xi_b} \hat{B}_i\left(\tilde{\boldsymbol{\xi}}_{sn}^{gl}\left(\bullet;\mathcal{H}\right);\mathcal{H}\right) \\
& \frac{\partial^2}{\partial h_k \partial \xi_q} \hat{B}_o\left(\tilde{\boldsymbol{\xi}}_{sn}^{gl}\left(\bullet;\mathcal{H}\right);\mathcal{H}\right) \Bigg] + \sum_{b=1}^{2} \left(J_{sn}^{gl}\right)_{ba}^{-1}\left(\bullet;\mathcal{H}\right) \;\; \frac{\partial^2}{\partial h_k \partial \xi_b} \hat{B}_i\left(\tilde{\boldsymbol{\xi}}_{sn}^{gl}\left(\bullet;\mathcal{H}\right);\mathcal{H}\right),
\end{aligned}$$

$$\frac{\partial^2}{\partial h_k \partial \boldsymbol{\xi}} \hat{B}_i\left(\tilde{\boldsymbol{\xi}}_{sn}^{gl}\left(\bullet;\mathcal{H}\right);\mathcal{H}\right) = \begin{cases}
\begin{pmatrix} -J_s\left(\tilde{\xi}^{gl};\mathcal{H}\right)^{-2} \;\; \frac{\partial}{\partial \tilde{\xi}} \tilde{N}_{ts}\left(\tilde{\xi}^{gl}\right) \;\; \hat{M}_s\left(\eta^l\right) \;\; \frac{\partial}{\partial \tilde{\xi}} \tilde{N}_{us}\left(\tilde{\xi}^{gl}\right) \\ 0 \end{pmatrix}, & \text{for } j = s \wedge v = s, \\
\begin{pmatrix} 0 \\ 0 \end{pmatrix}, & \text{for } j = s \wedge v = n, \\
\begin{pmatrix} 0 \\ J_n\left(\tilde{\xi}_{sn}^{gl}\left(\bullet;\mathcal{H}\right);\mathcal{H}\right)^{-1} \;\; \frac{\partial}{\partial \tilde{\xi}} \tilde{N}_{tn}\left(\tilde{\xi}_{sn}^{gl}\left(\bullet;\mathcal{H}\right)\right) \;\; \frac{\partial}{\partial \eta} \hat{M}_n\left(\eta^l\right) \;\; \tilde{N}_{us}\left(\tilde{\xi}^{gl}\right) \end{pmatrix}, & \text{for } j = n \wedge v = s, \\
\begin{pmatrix} 0 \\ -J_n\left(\tilde{\xi}_{sn}^{gl}\left(\bullet;\mathcal{H}\right);\mathcal{H}\right)^{-1} \;\; \frac{\partial}{\partial \tilde{\xi}} \tilde{N}_{tn}\left(\tilde{\xi}_{sn}^{gl}\left(\bullet;\mathcal{H}\right)\right) \;\; \frac{\partial}{\partial \eta} \hat{M}_n\left(\eta^l\right) \;\; \tilde{N}_{un}\left(\tilde{\xi}_{sn}^{gl}\left(\bullet;\mathcal{H}\right)\right) \end{pmatrix}, & \text{for } j = n \wedge v = n, \\
\begin{pmatrix} 0 \\ 0 \end{pmatrix}, & \text{otherwise},
\end{cases}$$

$$\frac{\partial}{\partial h_k} W^{gl}\left(\bullet;\mathcal{H}\right) = \begin{cases}
\Bigg[\frac{\partial}{\partial \tilde{\xi}} \tilde{N}_{us}\left(\tilde{\xi}^{gl}\right) \;\; \det\left(\boldsymbol{J}^{gl}\left(\bullet;\mathcal{H}\right)\right) + \\
\quad \sum_{o=1}^{M} \sum_{a=1}^{2} \sum_{b=1}^{2} J_s\left(\tilde{\xi}^{gl};\mathcal{H}\right) \;\; \det\left(\boldsymbol{J}^{gl}\left(\bullet;\mathcal{H}\right)\right) \;\; \left(\boldsymbol{J}^{gl}\left(\bullet;\mathcal{H}\right)^{-1}\right)_{ba} \;\; c_{oa} \\
\quad \frac{\partial^2}{\partial h_k \partial \xi_b} \hat{B}_o\left(\tilde{\boldsymbol{\xi}}_{sn}^{gl}\left(\bullet;\mathcal{H}\right);\mathcal{H}\right) \Bigg] \tilde{w}^{gl} w^l, & \text{for } v = s, \\
\sum_{o=1}^{M} \sum_{a=1}^{2} \sum_{b=1}^{2} J_s\left(\tilde{\xi}^{gl};\mathcal{H}\right) \;\; \det\left(\boldsymbol{J}^{gl}\left(\bullet;\mathcal{H}\right)\right) \;\; \left(\boldsymbol{J}^{gl}\left(\bullet;\mathcal{H}\right)^{-1}\right)_{ba} \;\; c_{oa} \\
\quad \frac{\partial^2}{\partial h_k \partial \xi_b} \hat{B}_o\left(\tilde{\boldsymbol{\xi}}_{sn}^{gl}\left(\bullet;\mathcal{H}\right);\mathcal{H}\right) \;\; \tilde{w}^{gl} w^l, & \text{for } v = n, \\
0, & \text{otherwise},
\end{cases}$$

for $(t, j) \leftarrow i$ and $(u, v) \leftarrow k$ according to Eq. (38) as well as $s = \mathcal{S}(l)$ and $n = \mathcal{N}(l)$ according to Eq. (19) and Eq. (24), respectively.

Appendix C. Tangent matrix for solving the governing equations

The node-wise linearizations of the discrete governing equations Eq. (64) and Eq. (65) are:

$$\boldsymbol{K}^n_{(m)(j)} = \frac{\partial \boldsymbol{F}^n_{int,m}}{\partial \boldsymbol{d}^n_j} = \sum_{g=1}^{n^{QP}} \eta_s \left[\left(\frac{\partial}{\partial \boldsymbol{x}} \hat{B}^{TP}_m \cdot \frac{\partial}{\partial \boldsymbol{x}} \hat{B}^{TP}_j \right) \boldsymbol{I} + \frac{\partial}{\partial \boldsymbol{x}} \hat{B}^{TP}_m \left(\frac{\partial}{\partial \boldsymbol{x}} \hat{B}^{TP}_j \right)^T \right]_{\boldsymbol{\xi}=\boldsymbol{\xi}^g} W^{g,n} \qquad (2 \times 2)$$

$$\boldsymbol{K}^n_{(m)(M+\zeta)} = \frac{\partial \boldsymbol{F}^n_{int,m}}{\partial q^n_\zeta} = -\sum_{g=1}^{n^{QP}} \left[\frac{\partial}{\partial \boldsymbol{x}} \hat{B}^{TP}_m \; \hat{A}^{TP}_\zeta \right]_{\boldsymbol{\xi}=\boldsymbol{\xi}^g} W^{g,n} \qquad (2 \times 1)$$

$$\boldsymbol{K}^n_{(M+z)(j)} = \frac{\partial Q^n_z}{\partial \boldsymbol{d}^n_j} = -\sum_{g=1}^{n^{QP}} \left[\hat{A}^{TP}_z \; \left(\frac{\partial}{\partial \boldsymbol{x}} \hat{B}^{TP}_j \right)^T \right]_{\boldsymbol{\xi}=\boldsymbol{\xi}^g} W^{g,n} \qquad (1 \times 2)$$

$$\boldsymbol{K}^n_{(M+z)(M+\zeta)} = 0. \qquad (1 \times 1)$$

Appendix D. Tangent matrix for solving contact problems

The linearization of the discrete nodal contact force Eq. (74) in addition to Appendix C leads to the node-wise stiffness tangent for contact problems

$$\boldsymbol{K}^n_{C,(m)(j)} = \frac{\partial \boldsymbol{R}^n_{C,m}}{\partial \boldsymbol{d}^n_j} = \boldsymbol{K}^n_{(m)(j)} - \frac{\partial \boldsymbol{F}^n_{C,m}}{\partial \boldsymbol{d}^n_j} = \boldsymbol{K}^n_{(m)(j)} + \sum_{g=1}^{n_b^{QP}} \left[\left(\kappa_R^{h,*} \boldsymbol{n}^n_W \boldsymbol{n}^{n\,T}_W + \kappa^*_S \boldsymbol{I} \right) \hat{B}^{TP}_m \hat{B}^{TP}_j \right]_{\boldsymbol{\xi}=\boldsymbol{\xi}^g} L^{g,n} \qquad (2 \times 2)$$

$$\boldsymbol{K}^n_{C,(m)(M+\zeta)} = \boldsymbol{K}^n_{(m)(M+\zeta)} \qquad (2 \times 1)$$

$$\boldsymbol{K}^n_{C,(M+z)(j)} = \boldsymbol{K}^n_{(M+z)(j)} \qquad (1 \times 2)$$

$$\boldsymbol{K}^n_{C,(M+z)(M+\zeta)} = 0. \qquad (1 \times 1)$$

References

[1] I. Gibson, D. Rosen, B. Stucker, M. Khorasani, Additive Manufacturing Technologies, Springer Cham, 2020, URL: https://doi.org/10.1007/978-3-030-56127-7.

[2] T. Hughes, J. Cottrell, Y. Bazilevs, Isogeometric analysis: CAD, finite elements, NURBS, exact geometry and mesh refinement, Comput. Methods Appl. Mech. Engrg. 194 (39) (2005) 4135–4195, http://dx.doi.org/10.1016/j.cma.2004.10.008, URL: https://www.sciencedirect.com/science/article/pii/S0045782504005171.

[3] S. Lipton, J. Evans, Y. Bazilevs, T. Elguedj, T. Hughes, Robustness of isogeometric structural discretizations under severe mesh distortion, Comput. Methods Appl. Mech. Engrg. 199 (5) (2010) 357–373, http://dx.doi.org/10.1016/j.cma.2009.01.022, URL: https://www.sciencedirect.com/science/article/pii/S0045782509000346. Computational Geometry and Analysis.

[4] I. Akkerman, Y. Bazilevs, C. Kees, M. Farthing, Isogeometric analysis of free-surface flow, J. Comput. Phys. 230 (11) (2011) 4137–4152, http://dx.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.

[5] H.C. Hille, S. Kumar, L. De Lorenzis, Floating isogeometric analysis, Comput. Methods Appl. Mech. Engrg. 392 (2022) 114684, http://dx.doi.org/10.1016/j.cma.2022.114684, URL: https://www.sciencedirect.com/science/article/pii/S0045782522000688.

[6] Z. Wieckowski, The material point method in large strain engineering problems, Comput. Methods Appl. Mech. Engrg. 193 (39) (2004) 4417–4438, http://dx.doi.org/10.1016/j.cma.2004.01.035, URL: https://www.sciencedirect.com/science/article/pii/S0045782504002282. The Arbitrary Lagrangian-Eulerian Formulation.

[7] D. Fußeder, Isogeometric finite element methods for shape optimization, 2015.

[8] A. Shamanskiy, Mesh Deformation in the Context of Isogeometric Analysis, (Ph.D. thesis), Technische Universität Kaiserslautern, 2020, URL: http://nbn-resolving.de/urn:nbn:de:hbz:386-kluedo-60820.

[9] P. Hennig, Adaptive Isogeometric Analysis of Phase-Field Models, (Ph.D. thesis), 2021, URL: https://nbn-resolving.org/urn:nbn:de:bsz:14-qucosa2-738116.

[10] T. Belytschko, Y. Krongauz, D. Organ, M. Fleming, P. Krysl, Meshless methods: An overview and recent developments, Comput. Methods Appl. Mech. Engrg. 139 (1) (1996) 3–47, http://dx.doi.org/10.1016/S0045-7825(96)01078-X, URL: https://www.sciencedirect.com/science/article/pii/S004578259601078X.

[11] J.-S. Chen, M. Hillman, S.-W. Chi, Meshfree methods: Progress made after 20 years, J. Eng. Mech. 143 (4) (2017) 04017001, http://dx.doi.org/10.1061/(ASCE)EM.1943-7889.0001176.

[12] M. Pasetto, J. Baek, J.-S. Chen, H. Wei, J.A. Sherburn, M.J. Roth, A Lagrangian/semi-Lagrangian coupling approach for accelerated meshfree modelling of extreme deformation problems, Comput. Methods Appl. Mech. Engrg. 381 (2021) 113827, http://dx.doi.org/10.1016/j.cma.2021.113827, URL: https://www.sciencedirect.com/science/article/pii/S0045782521001638.

[13] 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) URL: https://doi.org/10.1155/2020/2574127.
[14] S. Kumar, A. Vidyasagar, D.M. Kochmann, An assessment of numerical techniques to find energy-minimizing microstructures associated with nonconvex potentials, Internat. J. Numer. Methods Engrg. 121 (7) (2020) 1595–1628, http://dx.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.
[15] T. Matsuda, K. Tsukui, S. Ii, A particle-based method using the mesh-constrained discrete point approach for two-dimensional Stokes flows, 2022, arXiv:2206.01530.
[16] T. Belytschko, Y. Guo, W. Kam Liu, S. Ping Xiao, A unified stability analysis of meshless particle methods, Internat. J. Numer. Methods Engrg. 48 (9) (2000) 1359–1400, http://dx.doi.org/10.1002/1097-0207(20000730)48:9<1359::AID-NME829>3.0.CO;2-U.
[17] S. Kumar, K. Danas, D.M. Kochmann, Enhanced local maximum-entropy approximation for stable meshfree simulations, Comput. Methods Appl. Mech. Engrg. 344 (2019) 858–886, http://dx.doi.org/10.1016/j.cma.2018.10.030, URL: https://www.sciencedirect.com/science/article/pii/S0045782518305346.
[18] 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. Engrg. 329 (2018) 575–598, http://dx.doi.org/10.1016/j.cma.2017.10.002, URL: https://www.sciencedirect.com/science/article/pii/S0045782517306710.
[19] M. Arroyo, M. Ortiz, Local maximum-entropy approximation schemes: a seamless bridge between finite elements and meshfree methods, Internat. J. Numer. Methods Engrg. 65 (13) (2006) 2167–2202, http://dx.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.
[20] A. Rosolen, M. Arroyo, Blending isogeometric analysis and local maximum entropy meshfree approximants, Comput. Methods Appl. Mech. Engrg. 264 (2013) 95–107, http://dx.doi.org/10.1016/j.cma.2013.05.015, URL: https://www.sciencedirect.com/science/article/pii/S0045782513001321.
[21] 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, http://dx.doi.org/10.1007/s00466-014-0977-5, URL: https://link.springer.com/article/10.1007%2Fs00466-014-0977-5.
[22] D. Millán, N. Sukumar, M. Arroyo, Cell-based maximum-entropy approximants, Comput. Methods Appl. Mech. Engrg. 284 (2015) 712–731, http://dx.doi.org/10.1016/j.cma.2014.10.012, URL: https://www.sciencedirect.com/science/article/pii/S004578251400382X. Isogeometric Analysis Special Issue.
[23] F. Fathi, L. Chen, R. de Borst, X-IGALME: Isogeometric analysis extended with local maximum entropy for fracture analysis, Internat. J. Numer. Methods Engrg. n/a (n/a) (2021) http://dx.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.
[24] C. Hirt, A. Amsden, J. Cook, An arbitrary Lagrangian-Eulerian computing method for all flow speeds, J. Comput. Phys. 14 (3) (1974) 227–253, http://dx.doi.org/10.1016/0021-9991(74)90051-5, URL: https://www.sciencedirect.com/science/article/pii/0021999174900515.
[25] H. Hu, Direct simulation of flows of solid-liquid mixtures, Int. J. Multiph. Flow 22 (2) (1996) 335–352, http://dx.doi.org/10.1016/0301-9322(95)00068-2, URL: https://www.sciencedirect.com/science/article/pii/0301932295000682.
[26] D. Aubram, Development and experimental validation of an arbitrary Lagrangian-Eulerian (ALE) method for soil mechanics, geotechnik 38 (3) (2015) 193–204, http://dx.doi.org/10.1002/gete.201400030, URL: https://onlinelibrary.wiley.com/doi/abs/10.1002/gete.201400030, arXiv:https://onlinelibrary.wiley.com/doi/pdf/10.1002/gete.201400030.
[27] J. Cottrell, A. Reali, Y. Bazilevs, T. Hughes, Isogeometric analysis of structural vibrations, Comput. Methods Appl. Mech. Engrg. 195 (41) (2006) 5257–5296, http://dx.doi.org/10.1016/j.cma.2005.09.027, URL: https://www.sciencedirect.com/science/article/pii/S0045782505005451. John H. Argyris Memorial Issue. Part II.
[28] H. Gómez, V.M. Calo, Y. Bazilevs, T.J. Hughes, Isogeometric analysis of the cahn–hilliard phase-field model, Comput. Methods Appl. Mech. Engrg. 197 (49) (2008) 4333–4352, http://dx.doi.org/10.1016/j.cma.2008.05.003, URL: https://www.sciencedirect.com/science/article/pii/S0045782508001953.
[29] L. De Lorenzis, P. Wriggers, T.J. Hughes, Isogeometric contact: a review, GAMM-Mitt. 37 (1) (2014) 85–123, http://dx.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.
[30] I.J. Schoenberg, Contributions to the problem of approximation of equidistant data by analytic functions. Part A. On the problem of smoothing or graduation. A first class of analytic approximation formulae, Quart. Appl. Math. 4 (1946) 45–99.
[31] L. Piegl, W. Tiller, The NURBS Book, second ed., Springer-Verlag, New York, NY, USA, 1996.
[32] M.G. Cox, The numerical evaluation of B-splines, IMA J. Appl. Math. 10 (2) (1972) 134–149, http://dx.doi.org/10.1093/imamat/10.2.134, arXiv:https://academic.oup.com/imamat/article-pdf/10/2/134/2008789/10-2-134.pdf.
[33] C. de Boor, On calculating with B-splines, J. Approx. Theory 6 (1) (1972) 50–62, http://dx.doi.org/10.1016/0021-9045(72)90080-9, URL: https://www.sciencedirect.com/science/article/pii/0021904572900809.
[34] T. Hughes, A. Reali, G. Sangalli, Efficient quadrature for NURBS-based isogeometric analysis, Comput. Methods Appl. Mech. Engrg. 199 (5) (2010) 301–313, http://dx.doi.org/10.1016/j.cma.2008.12.004, URL: https://www.sciencedirect.com/science/article/pii/S0045782508004295. Computational Geometry and Analysis.
[35] F. Auricchio, F. Calabrò, T. Hughes, A. Reali, G. Sangalli, A simple algorithm for obtaining nearly optimal quadrature rules for NURBS-based isogeometric analysis, Comput. Methods Appl. Mech. Engrg. 249–252 (2012) 15–27, http://dx.doi.org/10.1016/j.cma.2012.04.014, URL: https://www.sciencedirect.com/science/article/pii/S004578251200134X. Higher Order Finite Element and Isogeometric Methods.
[36] C. Adam, T. Hughes, S. Bouabdallah, M. Zarroug, H. Maitournam, Selective and reduced numerical integrations for NURBS-based isogeometric analysis, Comput. Methods Appl. Mech. Engrg. 284 (2015) 732–761, http://dx.doi.org/10.1016/j.cma.2014.11.001, URL: https://www.sciencedirect.com/science/article/pii/S0045782514004228. Isogeometric Analysis Special Issue.

[37] F. Fahrendorf, L. De Lorenzis, H. Gomez, Reduced integration at superconvergent points in isogeometric analysis, Comput. Methods Appl. Mech. Engrg. 328 (2018) 390–410, http://dx.doi.org/10.1016/j.cma.2017.08.028, URL: https://www.sciencedirect.com/science/article/pii/S0045782517303717.

[38] Z. Zou, T. Hughes, M. Scott, D. Miao, R. Sauer, Efficient and robust quadratures for isogeometric analysis: Reduced Gauss and Gauss–Greville rules, Comput. Methods Appl. Mech. Engrg. 392 (2022) 114722, http://dx.doi.org/10.1016/j.cma.2022.114722, URL: https://www.sciencedirect.com/science/article/pii/S0045782522000871.

[39] M. Meßmer, T. Teschemacher, L.F. Leidinger, R. Wüchner, K.-U. Bletzinger, Efficient CAD-integrated isogeometric analysis of trimmed solids, Comput. Methods Appl. Mech. Engrg. 400 (2022) 115584, http://dx.doi.org/10.1016/j.cma.2022.115584, URL: https://www.sciencedirect.com/science/article/pii/S0045782522005564.

[40] Y. Bazilevs, K. Takizawa, T. Tezduyar, Computational Fluid–Structure Interaction, John Wiley & Sons, Ltd, 2013, http://dx.doi.org/10.1002/9781118483565, URL: https://onlinelibrary.wiley.com/doi/abs/10.1002/9781118483565, arXiv:https://onlinelibrary.wiley.com/doi/pdf/10.1002/9781118483565.

[41] J. Hinz, M. Möller, C. Vuik, Elliptic grid generation techniques in the framework of isogeometric analysis applications, Comput. Aided Geom. Design 65 (2018) 48–75, http://dx.doi.org/10.1016/j.cagd.2018.03.023, URL: https://www.sciencedirect.com/science/article/pii/S0167839618300372.

[42] M. Eck, J. Hadenfeld, Knot removal for B-spline curves, Comput. Aided Geom. Design 12 (3) (1995) 259–282, http://dx.doi.org/10.1016/0167-8396(94)00012-H, URL: https://www.sciencedirect.com/science/article/pii/016783969400012H.

[43] R.B. Bird, R.C. Armstrong, O. Hassager, Dynamics of Polymeric Liquids. Vol. 1, 2nd Ed. : Fluid Mechanics, Vol. 1, 1987, URL: https://www.osti.gov/biblio/6164599.

[44] A. Buffa, C. de Falco, G. Sangalli, IsoGeometric analysis: Stable elements for the 2D Stokes equation, Internat. J. Numer. Methods Fluids 65 (11–12) (2011) 1407–1422, http://dx.doi.org/10.1002/fld.2337, URL: https://onlinelibrary.wiley.com/doi/abs/10.1002/fld.2337, arXiv:https://onlinelibrary.wiley.com/doi/pdf/10.1002/fld.2337.

[45] T. Rüberg, F. Cirak, Subdivision-stabilised immersed b-spline finite elements for moving boundary flows, Comput. Methods Appl. Mech. Engrg. 209–212 (2012) 266–283, http://dx.doi.org/10.1016/j.cma.2011.10.007, URL: https://www.sciencedirect.com/science/article/pii/S0045782511003197.

[46] R. Comminal, F. Pimenta, J.H. Hattel, M.A. Alves, J. Spangenberg, Numerical simulation of the planar extrudate swell of pseudoplastic and viscoelastic fluids with the streamfunction and the VOF methods, J. Non-Newton. Fluid Mech. 252 (2018) 1–18, http://dx.doi.org/10.1016/j.jnnfm.2017.12.005, URL: https://www.sciencedirect.com/science/article/pii/S0377025717304883.

[47] M.P. Serdeczny, R. Comminal, D.B. Pedersen, J. Spangenberg, Experimental validation of a numerical model for the strand shape in material extrusion additive manufacturing, Addit. Manuf. 24 (2018) 145–153, http://dx.doi.org/10.1016/j.addma.2018.09.022, URL: https://www.sciencedirect.com/science/article/pii/S2214860418304585.