INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN ENGINEERING
Int. J. Numer. Meth. Engng 2015; **00**:1–25
Published online in Wiley InterScience (www.interscience.wiley.com). DOI: 10.1002/nme

A hybrid variational-collocation immersed method for fluid-structure interaction using unstructured T-splines

Hugo Casquero [1*†], Lei Liu [2], Carles Bona-Casas [1], Yongjie Zhang [2], Hector Gomez [1]

[1] *Departamento de Métodos Matemáticos, Universidade da Coruña, Campus de A Coruña, 15071, A Coruña, Spain.*
[2] *Department of Mechanical Engineering, Carnegie Mellon University, Pittsburgh, PA 15213, U.S.A.*

SUMMARY

We present a hybrid variational-collocation, immersed, and fully-implicit formulation for fluid-structure interaction (FSI) using unstructured T-splines. In our immersed methodology, we define an Eulerian mesh on the whole computational domain and a Lagrangian mesh on the solid domain which moves arbitrarily on top of the Eulerian mesh. Mathematically, the problem reduces to solving three equations, namely, the linear momentum balance, mass conservation, and a condition of kinematic compatibility between the Lagrangian displacement and the Eulerian velocity. We use a weighted residual approach for the linear momentum and mass conservation equations, but we discretize directly the strong form of the kinematic relation, deriving a hybrid variational-collocation method. We use T-splines for both the spatial discretization and the information transfer between the Eulerian mesh and the Lagrangian mesh. T-splines offer us two main advantages against NURBS: They can be locally refined and they are unstructured. The generalized-α method is used for the time discretization. We validate our formulation with a common FSI benchmark problem achieving excellent agreement with the theoretical solution. An example involving a partially immersed solid is also solved. The numerical examples show how the use of T-junctions and extraordinary nodes results in an accurate, efficient, and flexible method.

Received . . .

KEY WORDS: Fluid-structure interaction; Isogeometric analysis; Immersed methods; Analysis-suitable T-splines; Variational multiscale; Collocation

1. INTRODUCTION

Isogeometric analysis (IGA) was presented by Hughes *et al.* in [1] and explained in detail in [2]. IGA was invented to fill the gap between computer aided design (CAD) and finite element analysis (FEA) in order to reduce the communication process between design and analysis. IGA's key idea is to utilize the blending functions of CAD in FEA to represent both the geometry and the solution using the isoparametric concept. Different CAD functions may be used, including non-uniform rational B-Splines (NURBS), T-splines and subdivision surfaces. NURBS-based IGA is completely widespread nowadays. An important aftereffect of using NURBS as a basis in analysis is that we can take advantage of their natural inter-element smoothness, which has positive consequences in several application areas, such as, body-fitted fluid-structure interaction (FSI) [3, 4, 5, 6], immersed FSI [7, 8, 9], fluid mechanics [10, 11, 12, 13, 14, 15], phase-filed models [16, 17, 18, 19, 20], biomechanics [21, 22], structural mechanics [23, 24, 25, 26], shape memory alloys [27, 28, 29], shell modeling [30, 31, 32, 33], contact problems [34, 35, 36], among others.

[*] Correspondence to: Hugo Casquero, Departamento de Métodos Matemáticos, Universidade da Coruña, Campus de A Coruña, 15071, A Coruña, Spain.
[†] E-mail: hugo.casquero@udc.es

1.1. *Isogeometric analysis: From NURBS to T-splines*

NURBS were the CAD technology of choice when isogeometric analysis was proposed. NURBS are widely used in engineering design and have a number of appealing features, such as, for example, higher-order global continuity, the convex hull property, the ability to represent exactly all quadric curves and surfaces, affine covariance, pointwise nonnegativity, the variation diminishing property, and capability for free-form surface modeling. Nevertheless, NURBS have some major limitations such as the difficulty to represent watertight surfaces. Another shortcoming is their tensor product structure, which avoids local refinement or unstructured meshing. In [37], Sederberg *et al.* introduced T-spline surfaces in CAD as a generalization of NURBS. T-splines overcome the aforementioned limitations of NURBS while maintaining all their good properties. In the last few years, Zhang *et al.* proposed several procedures to create volumetric T-splines [38, 39, 40, 41, 42]. T-splines are a superset of NURBS. An appealing feature of T-splines is that they are forward and backward compatible with NURBS. In particular, any T-spline surface can be converted into one or more NURBS surfaces by perfoming repeated local refinement to eliminate all T-junctions. In the commercial field, this is a significant advantage of T-splines over other candidates that are unstructured and allow local refinement (such as subdivision surfaces), but are not compatible with NURBS.

In [43], T-splines were introduced in the FEA field. In contrast with NURBS, T-spline blending functions as introduced in [37] do not define a basis because they might be linearly dependent. Linear independence is not necessary in CAD, but it is a requirement for successful analysis. This problem was addressed defining a subset of T-splines where linear independence is guaranteed [44, 45]. The newly-proposed subset of T-splines was called analysis-suitable T-splines (ASTS). Also, to facilitate the direct inclusion of T-splines in a finite element code, the concept of Bézier extraction was proposed [46, 47]. Significant developments have occurred since then: The local refinement of ASTS was analyzed in [48], the conversion of an unstructured quadrilateral mesh to a standard T-spline surface was detailed in [49], the conversion of a trimmed NURBS surface to an untrimmed T-spline surface was explained in [50], the application of the hierarchical refinement idea to ASTS was explored in [51], and the application of collocation methods [52, 53, 54, 55] to ASTS was studied in [56]. ASTS have been applied to several problems in computational mechanics such as dynamic brittle fracture [57], cohesive zone modeling [58], gradient damage models [59], large deformation frictionless contact [60], and acoustics [61]. However, their use in FSI, which is precisely the goal of this paper, has been limited to a body-fitted method where a bicubic T-spline surface discretization of a rotation-free Kirchhoff-Love shell model was used for the solid and volumetric quadratic NURBS were used for the fluid [62].

1.2. *Fluid-structure interaction methods*

FSI methods are typically divided into interface-tracking and interface-capturing techniques. In interface-tracking methods, as the fluid domain changes its shape due to the fluid-solid interface motion, the fluid mesh is updated to follow this movement. In interface-capturing methods, the fluid mesh does not move to track the fluid-solid interface. Interface-tracking (i.e. moving-mesh) methods are often based on the arbitrary Lagrangian-Eulerian (ALE) formulation [63]. The ALE description is employed in the fluid domain, whereas the Lagrangian description is used in the solid. ALE-based FSI methods have been used in various applications, such as, for example, hemodynamics [5, 64, 65, 66, 67, 68], wind turbines [69, 70], or aeroelasticity [71]. As an alternative to ALE-based methods, space-time body-fitted methods are becoming increasingly popular [72, 73]. For a thorough discussion about body-fitted approaches the reader is referred to [74].

An important example of the interface-capturing approach is the immersed boundary method [75], which was widely used in the simulation of biomechanical problems [76, 77, 78, 79, 80, 81]. In this immersed technique, the solid is discretized using a Lagrangian mesh [†] that can move freely on top

[†] By Lagrangian mesh, we refer to a mesh whose nodes are attached to material (rather than physical) points. It does not imply in any way the use of classical Lagrangian finite elements, which are not utilized in this work.

DOI: 10.1002/nme

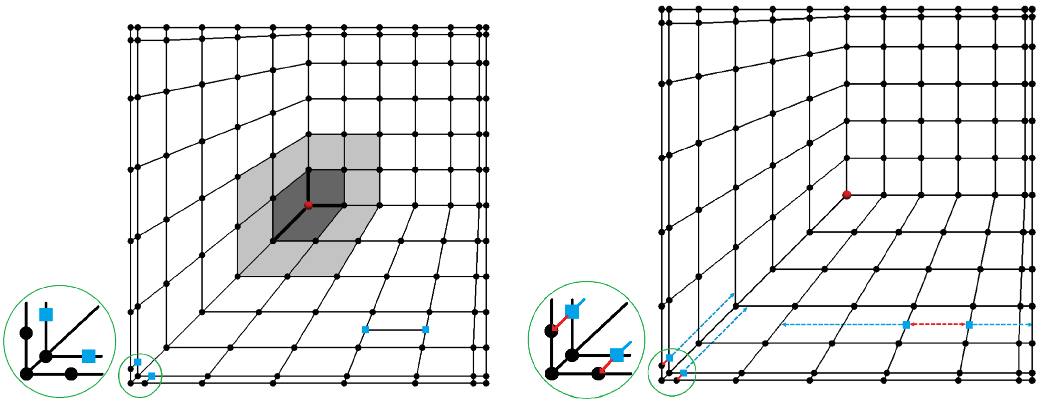
of a background Eulerian mesh that spans the whole computational domain. An information transfer algorithm is required to couple these two independent meshes and this was initially accomplished by way of smoothed-out Dirac delta functions. In many cases of practical interest, the immersed approach avoids the use of mesh-updating or remeshing procedures, which are needed in body-fitted FSI methods. This is an advantage of the immersed method because changing the mesh, either geometrically or topologically, increases the computational cost and may introduce some issues. For example, when remeshing is not used for cases involving large translations or rotations of the solid, the fluid mesh will become highly distorted, compromising the accuracy of the solution. When remeshing is used, projections between different meshes are necessary, this also introduces additional inaccuracies. Thus, for FSI applications which involve large deformations or topological changes of the fluid domain, body-fitted methods may not be the best option, and this is precisely the area where immersed algorithms shine. In the 2000's an important development occurred in the field of immersed FSI methods, namely, the introduction of the so-called immersed finite element method (IFEM) [82]. The IFEM has been successfully applied to a number of problems [83, 84].

Notably, recent research endeavors have targeted the creation of new FSI immersed methodologies using the IGA framework. This new developing type of computational techniques was termed *immersogeometric analysis*. It aims to capture the solid geometry exactly and embed it in a background mesh. This concept was proposed in [7, 9], where the authors used thin-shell structures modeled geometrically as surfaces embedded in a fluid. In those articles, various benchmark problems are considered. The results were compared with body-fitted methods and the algorithm was applied to a cardiovascular problem involving tri-leaflet bioprosthetic heart valves. In [8], we proposed a new method which falls in the same category. In contrast with that of [7, 9], our method considers volumetric solids. The algorithm was applied to 2D and 3D benchmark problems with theoretical solution achieving excellent agreement. In addition, we showed that the higher-order continuity of IGA basis functions increases the robustness of immersed methodologies in problems involving large deformations of the solid.

We feel that IGA-based, immersed FSI methods would benefit from the use of unstructured meshes and local refinement and this motivates our current work. Here, we bring the T-spline technology to the immersed FSI realm, discretizing both the fluid and the solid with analysis-suitable T-splines. We show that the local refinement capability of T-splines may be used to achieve enhanced spatial resolution where it is needed and the unstructured nature of T-splines may help avoid the presence of singular points in the parameterization, leading to more flexible immersed FSI formulations. Additionally, we use a hybrid variational-collocation approach, namely, we discretize the linear momentum balance and mass conservation equation using the variational multiscale (VMS) method and the kinematic relation between the Eulerian velocity and the Lagrangian displacement using collocation. Our hybrid variational-collocation approach enhances the efficiency of the algorithm, particularly when higher orders are used.

1.3. Structure and content of the paper

This paper is organized as follows. In Section 2, we summarize the T-spline paraphernalia. Section 3 briefly describes the immersed FSI method. We start by defining important nomenclature and describing the kinematics involved in the FSI methodology. We continue stating the main three equations involved in our methodology, which are the linear momentum balance equation, the mass conservation equation and the kinematic relation between the Lagrangian and the Eulerian motions. Next, we discretize in space using a hybrid variational-collocation method based on T-splines. Time discretization is performed using a fully-implicit, second-order accurate algorithm based on the generalized-α method [85, 86]. Section 4 displays two numerical examples that show the performance of the proposed method. The first example is a common FSI benchmark problem, namely, a free-falling cylinder embedded in a channel. Our numerical result is checked against the theoretical solution, showing good quantitative agreement. The second example involves a deformable solid partially immersed in a water flow. The solid has its bottom part fixed, what modifies the flow patterns creating a recirculation region downstream the solid. Finally, in Section 5, some concluding remarks will be drawn.



(a) Unstructured T-mesh

(b) Extended unstructured T-mesh

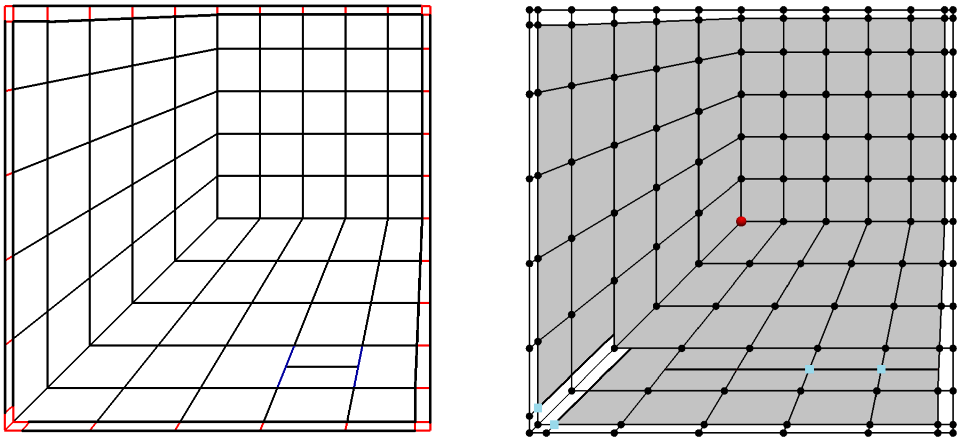
Figure 1. (Color online) (a) Unstructured T-mesh with one extraordinary node and four T-junctions. The extraordinary node is marked with a red circle and the T-junctions are highlighted with blue squares. The spoke edges of the extraordinary node are indicated by thick black lines. The one-ring neighborhood of the extraordinary node is drawn with darkly shaded faces, while the two-ring neighborhood is the one-ring and the lightly shaded faces. (b) Extended T-mesh. The extended T-mesh is obtained by adding the T-junction extensions to the T-mesh. Face extensions are denoted by blue dashed arrows and edge extensions are denoted by red dashed arrows. The inset represents a zoom in of the bottom left corner.

2. T-SPLINE OVERVIEW

In this section, we present a brief introduction to the T-spline technology and remit the reader to other references for further details when it is needed. We assume that the reader is familiarized with NURBS and use them as a starting point to describe T-splines. Although we will limit ourselves to cubic T-spline surfaces, in some cases we will use the notation d_p for the number of parametric dimensions, d for the number of spatial dimensions, and p for the polynomial degree.

2.1. T-mesh, T-junctions, extraordinary nodes, and T-junction extensions

Here, we follow [45] and define the *T-mesh* as a polygonal mesh which encodes all the topological information associated to the T-spline technology. The polygons that form the T-mesh will be referred to as *faces*. The word element is used in the literature as a synonym of face, but we will reserve it for something else. We define the T-mesh in such a way that its topology is identical to that of the control mesh. Fig. 1(a) shows an example of a T-mesh similar to those used in the computations presented in Sect. 4. Small circles and squares identify the *vertices* of the T-mesh, while a line joining (exactly) two vertices defines an *edge*. Vertices will also be referred to as *nodes*. We call *valence* the number of edges that touch a vertex. Note that our T-mesh is allowed to have T-junctions and extraordinary nodes. L-junctions and I-junctions are not allowed [44]. T-junctions are analogous to hanging nodes in classical finite elements [see the blue squares in Fig. 1(a)]. *Extraordinary nodes* are those interior vertices which not having valence four, do not qualify as T-junctions [see the red circular vertex in Fig. 1(a)]. The presence of extraordinary nodes makes the T-mesh unstructured. As we will see later, in cases of practical relevance, T-junctions will be located far away from extraordinary nodes. Therefore, in the vicinity of T-junctions, we can define a rectangular topology and a local index space [87, 88], which may be used to describe T-junction extensions exactly as in [44, 87]. Without going into the details of the definitions of the extensions, which can be found in the aforementioned papers, we just note that there are two types, namely, face extensions and edge extensions. We will also use the concept of first-bay face extension (used synonymously to one-bay face extension) as defined in [88, 45]. The *extendend T-mesh* is simply the T-mesh augmented with the T-junction extensions. As an illustration, we present in Fig. 1(b) the extended T-mesh associated to the T-mesh in Fig. 1(a). The blue dashed lines represent face extensions, while the red dashed lines denote edge extensions. For future reference, we also define



(a) Knot interval configuration

(b) Elemental T-mesh

Figure 2. (Color online) (a) Valid knot interval configuration associated to a particular T-mesh. The black, blue and red edges have associated knot intervals of 1, 1/2, and 0, respectively. (b) Elemental T-mesh. The polygons which are not shaded are not part of the elemental T-mesh. T-junction extensions become new edges which may split faces into several elements of the elemental T-mesh.

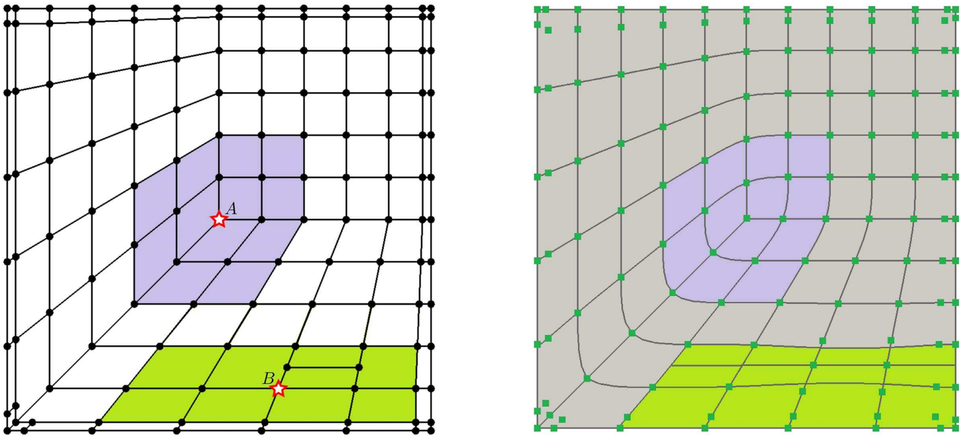
the one-, two- and n-ring neighborhood of a T-mesh vertex. The *one-ring neighborhood* of a vertex is the set of T-mesh faces that touch that particular vertex. The *two-ring neighborhood* of a vertex is the set of faces that comprise its one-ring neighborhood and those that touch the one-ring neighborhood. The *n-ring neighborhood* may be defined recursively in the obvious way. Fig. 1(a) shows the one-ring neighborhood of the extraordinary node highlighted with darkly-shaded faces. The two-ring neighborhood is composed by the darkly-shaded and the lightly-shaded faces. Finally, we call *spoke edges* the T-mesh edges that end in an extraordinary node [see in Fig. 1(a) the edges plotted with thick lines in the neighborhood of the extraordinary node].

2.2. Knot intervals and elemental T-mesh

We note that all the definitions given so far are purely topological and do not depend upon any geometrical information. However, to construct T-spline blending functions and perform analysis, we need to utilize geometrical information. The first step is to assign a non-negative real number to each T-mesh edge. These non-negative real numbers will be referred to as *knot intervals* [37] and they will be used to define the support of T-spline blending functions. There is significant freedom to select knot intervals, but they have to satisfy some conditions in order to define a valid knot interval configuration. A valid knot interval configuration requires that the knot intervals assigned to opposite edges of a face add up to the same value. Due to our choice to have a T-mesh with the same topology as the control mesh, we also define an outer ring of zero-length knot intervals in the T-mesh. This choice is analogous to that of using open knot vectors in NURBS-based analysis, which leads to repeated knots on the boundary and eases the imposition of Dirichlet boundary conditions. Fig. 2(a) shows a possible valid knot interval configuration associated to the T-mesh depicted in Fig. 1(a). The knot intervals assigned to red, blue and black edges take the value 0, 1/2 and 1, respectively. Note that edges associated to zero-length knot intervals are plotted with a finite length to make simpler the interpretation of the figure.

Another important object in the T-spline technology is the *elemental T-mesh*. The elemental T-mesh is a new set of polygons which might be obtained from the extended T-mesh and the knot interval configuration. The polygons that form the elemental T-mesh will be called *elements*. T-mesh faces of zero surface (i.e., zero measure) do not contribute elements to the elemental T-mesh and all T-mesh faces of non-zero surface give rise to at least one element in the elemental T-mesh. Those which are not crossed by a T-junction extension give rise to exactly one element, and those crossed by T-junction extensions may give rise to more than one. Fig. 2(b) shows as shaded areas

DOI: 10.1002/nme



(a) Sets $\mathcal{S}_A^T$ and $\mathcal{S}_B^T$

(b) Bézier mesh and Greville points

Figure 3. (Color online) (a) Sets $\mathcal{S}_A^T$ (violet) and $\mathcal{S}_B^T$ (green) associated to vertices A and B of the T-mesh. For simplicity, we have omitted faces of zero surface in $\mathcal{S}_B^T$ as they will not contribute elements in the Bézier mesh. (b) Bézier mesh in physical space and support of the basis functions associated to vertices A (violet) and B (green). Dark green squares denote Greville points in physical space.

delimited by black lines the polygons that compose the elemental T-mesh. See, in particular, how the T-mesh faces in the bottom right area have been split into two elements in the elemental T-mesh. Note also how the zero-surface faces do not form elements (see the outer ring and the area in the bottom left corner). The elements of the elemental T-mesh are important objects for analysis because they delimit areas in which all T-spline blending functions are $\mathcal{C}^\infty$. In analysis, the elements of the elemental T-mesh will be pushed forward to physical space using the isoparametric concept [89]. Each element of the elemental T-mesh gives rise to a so-called Bézier element in physical space. Bézier elements are suitable regions to perform numerical integration because they are bounded by lines across which at least one T-spline blending function fails to be $\mathcal{C}^\infty$. Within a Bézier element, however, all T-spline blending functions are $\mathcal{C}^\infty$.

2.3. *T-spline blending functions and Bézier extraction*

As indicated before, we defined the T-mesh in such a way that it is topologically identical to the control mesh. Thus, each vertex of the T-mesh has a control point associated and, therefore, a T-spline blending function. To each vertex, we associate a capital index $A \in \{1, \ldots, n\}$, where n is the global number of control points. We proceed now to define the T-spline blending function associated to the vertex A, which will be denoted by N_A. Let us begin by defining the support of N_A, which can be inferred from the T-mesh and the knot interval configuration. To each vertex A of the T-mesh, we will associate a set of T-mesh faces denoted by $\mathcal{S}_A^T$. Some of the faces in $\mathcal{S}_A^T$ will have a correspondence in the elemental T-mesh, giving rise to a new set that we call $\mathcal{S}_A^{T_e}$. The region occupied by the elements in $\mathcal{S}_A^{T_e}$ will be mapped to the physical space producing a set of Bézier elements. The region defined by those Bézier elements is the support of N_A in physical space. Let us describe now $\mathcal{S}_A^T$ for an arbitrary vertex A. If A is an extraordinary node or falls in the one-ring of an extraordinary node, then $\mathcal{S}_A^T$ is the two-ring neighborhood of A. Otherwise, $\mathcal{S}_A^T$ is constructed by marching through the T-mesh in each topological direction, starting at A, until $p-1$ vertices or perpendicular edges are intersected or until the boundary of the T-mesh is encountered. Fig. 3(a) shows the sets $\mathcal{S}_A^T$ and $\mathcal{S}_B^T$ for vertices A and B (marked with stars in the plot). $\mathcal{S}_A^T$ is composed of the violet faces, while $\mathcal{S}_B^T$ is comprised by the green faces. Note that the faces of zero surface in $\mathcal{S}_B^T$ have been disregarded, as they will not contribute Bézier elements in physical space [recall the knot interval configuration shown in Fig. 2(a)]. With the above information at hand, it is possible to anticipate which basis functions will have support on a given Bézier element e, which occupies a region Ω^e in physical space. To simplify notation, we introduce a local numbering for the basis functions as it is typically done in the finite element method. Here, we follow [90], and use the array IEN to establish a correspondence between local and global numbering. In particular, we use the formula $A = \text{IEN}(a, e)$, where A is a global basis function index, a is a local-to-element basis function counter, and e denotes the element number. Let us define the vector $\boldsymbol{M}^e = \{M_a^e\}_{a=1}^{n^e}$, where the functions M_a^e are progenitors of the basis functions with support on the element e. Note that, as emphasized with the superscript e in n^e, the length of the vector $\boldsymbol{M}^e$ changes from one element to another. This is a peculiar feature of T-splines analysis which does not hold true for NURBS-based analysis.

The concept of Bézier extraction permits computing $\boldsymbol{M}^e$ as a linear combination of the canonical tensor product Bernstein polynomials defined on a fixed parent element, namely, $\square = [-1, +1]^2$. This can be expressed mathematically as

$$\boldsymbol{M}^e(\boldsymbol{\xi}) = \boldsymbol{C}^e \boldsymbol{B}(\boldsymbol{\xi}), \quad \boldsymbol{\xi} \in \square, \tag{1}$$

where $\boldsymbol{B} = \{B_i^p\}_{i=1}^{(p+1)^{d_p}}$ is a vector containing the two-dimensional Bernstein polynomials of degree p in the domain $\square$ (see [45] for a precise description). The linear operator $\boldsymbol{C}^e$ can be represented by a rectangular matrix and it is called element Bézier extraction operator. Although we will not go into the details of the computation of $\boldsymbol{C}^e$, which may be found in [46] and [45], we point out that the Bézier extraction operator is always computed from the knot interval configuration of the T-mesh. However, we need to distinguish two different types of elements. The elements in the two-ring of an extraordinary node are called irregular elements while the remaining elements are called regular elements. The Bézier extraction procedure for the regular elements is based on knot insertion and was precisely described in [46]. However, the Bézier extraction procedure for the irregular elements is based on imposing basic analysis-suitability properties and geometric constraints and it was described in detail in [45]. In any case, Eq. (1) applies and can be used to compute $\boldsymbol{M}^e$ as a linear combination of Bernstein polynomials. Furthermore, in practical cases, it may be useful to utilize rational functions, typically to reproduce exactly a particular geometry. Given a set of weights $\{w_A\}_{A=1}^n$ and their local counterparts for element e, namely, $\{w_a^e\}$, we can rationalize the functions in $\boldsymbol{M}^e$ as

$$N_a^e(\boldsymbol{\xi}) = \frac{w_a^e M_a^e(\boldsymbol{\xi})}{\sum_{b=1}^{n^e} w_b^e M_b^e(\boldsymbol{\xi})}, \tag{2}$$

where N_a^e is the a-th rational T-spline blending function over the element e. This finalizes the process to construct the basis functions in parameter space. We would like to point out that if all the weights take the value one, then the rational T-spline blending functions become polynomials due to the partition of unity property, that is,

$$\sum_{b=1}^{n^e} M_b^e(\boldsymbol{\xi}) = 1. \tag{3}$$

Finally, to perform computations on non-trivial geometries, we need to map the parent element to physical space using a geometrical mapping $\boldsymbol{x}^e : \square \mapsto \Omega^e$. The basic information needed to produce this mapping is a set of geometry control points $\{\boldsymbol{Q}_B\}_{B=1}^n$ and the set of weights we just introduced, namely, $\{w_B\}_{B=1}^n$. A given geometry control point $\boldsymbol{Q}_A$ will have local indices associated, such that $\boldsymbol{Q}_A = \boldsymbol{Q}_a^e$, where $A = \text{IEN}(a, e)$. The same argument applies to weights. Thus, the geometrical mapping local to element e can be defined as

$$\boldsymbol{x}^e(\boldsymbol{\xi}) = \sum_{a=1}^{n^e} \boldsymbol{Q}_a^e N_a^e(\boldsymbol{\xi}) \quad \boldsymbol{\xi} \in \square. \tag{4}$$

Using the T-spline geometrical mapping (4), we can get the basis functions in physical space. In particular, the restriction of N_A to the element e is just the push forward of N_a^e, where $A = \text{IEN}(a, e)$. Eq. (4) will also be used to map the elemental T-mesh to the Bézier physical mesh. Fig. 3(b) shows a representative example of a Bézier mesh. In summary, as shown by Eqs. (1)–(4), the use of Bézier extraction allows to use T-splines in a finite element code modifying just the shape function subroutine.

2.4. Continuity of T-spline blending functions

A T-spline blending function is at least $\mathcal{C}^{p-1}$ across an element boundary shared by two regular elements if the knot intervals associated to the edges in the other topological direction are not zero. The number of continuous derivatives would be decreased by m if we introduce m zero-length knot intervals. Now, we proceed to describe the continuity in the two-ring neighborhood of an extraordinary node. A T-spline blending function is $\mathcal{C}^2$-continuous across an element boundary shared by a regular and an irregular element. The continuity is reduced to $\mathcal{C}^0$ across an element boundary shared by two irregular elements in the one-ring of an extraordinary node. The continuity across element boundaries in the two-ring of an extraordinary node not described heretofore is $\mathcal{C}^1$.

2.5. Greville points

In our FSI algorithm presented in the following section, we will use a collocation approach that takes inspiration from the emerging field of isogeometric collocation [52, 56]. Isogeometric collocation is often carried out using the so-called Greville points. Greville points are easily defined for B-Splines and NURBS [91], but their extension to unstructured T-splines is not trivial. Here, we utilize the same generalization of classical Greville points that was used in [45]. We associate a Greville point to each vertex of the T-mesh. Greville points associated to extraordinary nodes are located right on top of extraordinary nodes. The location of Greville points associated to vertices which are not extraordinary is defined by the knot intervals assigned to the edges that touch that vertex (see the details in [45]). We can collect all Greville points defined this way in the set $\{\widehat{\boldsymbol{\tau}}_i\}_{i=1}^n$. For each of these Greville points, we need to find the element in which it falls, and its corresponding coordinates in the parent element, namely, $\boldsymbol{\xi}_i^e$. The Greville points in physical space are computed as $\widetilde{\boldsymbol{\tau}}_i = \boldsymbol{x}^e\left(\boldsymbol{\xi}_i^e\right)$. The set of all those Greville points in physical space is denoted by $\widetilde{\mathcal{M}}^h = \{\widetilde{\boldsymbol{\tau}}_i\}_{i=1}^n$. The Greville points for the T-mesh shown in Fig. 1(a) are plotted in Fig. 3(b) using small green squares.

2.6. Analysis-suitable T-splines

Without any restriction on the T-mesh topology, T-splines blending functions do not inherit all the important mathematical properties of NURBS. In particular, some problems can appear with linear independence and partition of unity. For T-meshes of rectangular topology, this issue was solved in [44] by defining a subset of T-splines called analysis-suitable T-splines (ASTS) where all the mathematical properties of NURBS are guaranteed including linear independence and partition of unity. When the T-mesh is unstructured, assuring the linear independence and partition of unity of blending functions is more difficult. In what follows we summarize the current understanding concerning analysis suitability of T-splines with and without extraordinary nodes:

1. It was proven in [44] that a T-mesh with T-junctions and no extraordinary nodes gives rise to ASTS if no horizontal and vertical T-junction extensions intersect with each other.
2. It was proposed in [45] that a T-mesh with extraordinary nodes and no T-junctions gives rise to ASTS if no extraordinary node lies within the three-ring neighborhood of another extraordinary node.
3. It was proposed in [45] that a T-mesh with T-junctions and extraordinary nodes gives rise to ASTS if the conditions (1) and (2) hold true and no one-bay face extension spans a face in the three-ring neighborhood of an extraordinary node.

Note that ASTS are defined using simple topological rules. Although it is possible to construct linear independent T-spline blending functions violating the above conditions, in what follows, we restrict ourselves to the set of ASTS.

DOI: 10.1002/nme

3. IMMERSED FLUID-STRUCTURE INTERACTION METHOD

In this Section, we describe the basics of our formulation at the continuous and discrete levels and we focus on the main differences between working with T-splines and NURBS for this immersed FSI methodology. The detailed derivations of the formulation at the continuous and discrete levels can be found in our earlier paper [8].

3.1. *Fluid-solid system*

In what follows, $\Omega \subset \mathbb{R}^d$ is a time-independent open set which encloses our fluid-solid system. $\Omega_t^f \subset \Omega$ denotes the open subset of Ω occupied by the fluid at time t, which we call the fluid domain. The solid domain at time t is $\Omega_t^s \subset \Omega$. Note that, although we have assumed Ω to be fixed in time, the fluid and solid domains do depend on time, as indicated by the subscript. The solid and fluid domains define a partition of Ω such that $\overline{\Omega} = \overline{\Omega_t^f \cup \Omega_t^s}$. The fluid and solid domains are not allowed to overlap ($\Omega_t^f \cap \Omega_t^s = \emptyset$, where $\emptyset$ is the empty set), but they meet at the solid-fluid interface that we call Γ_t^I $(\overline{\Omega_t^f} \cap \overline{\Omega_t^s} = \Gamma_t^I)$. The boundaries of Ω, Ω_t^f and Ω_t^s are denoted by Γ, Γ_t^f and Γ_t^s , respectively, and their unit outward normals by $\boldsymbol{n}$, $\boldsymbol{n}^f$ and $\boldsymbol{n}^s$, where the subscript t is omitted for notational simplicity.

3.2. *Kinematics*

A Lagrangian description will be used for the solid domain, so let us define a reference configuration Ω_0^s for our solid body. We consider the mapping $\boldsymbol{\varphi} : \Omega_0^s \times (0,T) \mapsto \mathbb{R}^d$, where $(0,T)$ is the time interval of interest. We assume that $\boldsymbol{\varphi}$ is sufficiently smooth, orientation preserving and invertible [92]. Points $\boldsymbol{X}$ in Ω_0^s are called material points or particles, while points in $\mathbb{R}^d$ are denoted $\boldsymbol{x}$ and are called spatial points. To define the mapping $\boldsymbol{\varphi}$ we make use of the displacement field $\overline{\boldsymbol{u}}^s : \Omega_0^s \times (0,T) \mapsto \mathbb{R}^d$. In particular,

$$\boldsymbol{\varphi}(\boldsymbol{X},t) = \boldsymbol{X} + \overline{\boldsymbol{u}}^s(\boldsymbol{X},t) \, . \tag{5}$$

We define the velocity $\overline{\boldsymbol{v}}^s : \Omega_0^s \times (0,T) \mapsto \mathbb{R}^d$, and the acceleration $\overline{\boldsymbol{a}}^s : \Omega_0^s \times (0,T) \mapsto \mathbb{R}^d$ of a material point as

$$\overline{\boldsymbol{v}}^s(\boldsymbol{X},t) = \frac{\partial \overline{\boldsymbol{u}}^s(\boldsymbol{X},t)}{\partial t} \, , \tag{6}$$

$$\overline{\boldsymbol{a}}^s(\boldsymbol{X},t) = \frac{\partial \overline{\boldsymbol{v}}^s(\boldsymbol{X},t)}{\partial t} = \frac{\partial^2 \overline{\boldsymbol{u}}^s(\boldsymbol{X},t)}{\partial t^2} \, . \tag{7}$$

Associated to the function $\overline{\boldsymbol{v}}^s$, we define $\boldsymbol{v}^s : \Omega_t^s \times (0,T) \mapsto \mathbb{R}^d$ as

$$\overline{\boldsymbol{v}}^s(\boldsymbol{X},t) = \boldsymbol{v}^s(\boldsymbol{\varphi}(\boldsymbol{X},t),t) \quad \text{for all} \quad \boldsymbol{X} \in \Omega_0^s,\ t \in [0,T] \, . \tag{8}$$

Note that although $\overline{\boldsymbol{v}}^s$ and $\boldsymbol{v}^s$ represent the same physical quantity, namely the solid velocity, they are different functions, and thus, we use different notation for them. In what follows, we will call $\overline{\boldsymbol{v}}^s$ Lagrangian or material velocity and $\boldsymbol{v}^s$ Eulerian or spatial velocity. In general, we will use a bar for Lagrangian functions whenever there is possibility of confusion. Applying the chain rule to (8), we obtain

$$\overline{\boldsymbol{a}}^s(\boldsymbol{X},t) = \frac{\partial \overline{\boldsymbol{v}}^s}{\partial t}(\boldsymbol{X},t) = \frac{\partial \boldsymbol{v}^s}{\partial t}(\boldsymbol{x},t) + \boldsymbol{v}^s(\boldsymbol{x},t) \cdot \nabla_{\boldsymbol{x}} \boldsymbol{v}^s(\boldsymbol{x},t) \quad \text{if} \quad \boldsymbol{x} = \boldsymbol{\varphi}(\boldsymbol{X},t) \, . \tag{9}$$

In the following, we will also use the notation

$$\dot{\boldsymbol{v}^s} = \frac{\partial \boldsymbol{v}^s}{\partial t} + \boldsymbol{v}^s \cdot \nabla_{\boldsymbol{x}} \boldsymbol{v}^s \, , \tag{10}$$

which is standard in computational mechanics. From the mapping $\boldsymbol{\varphi}$, we may also define the deformation gradient $\boldsymbol{F} : \Omega_0^s \times (0,T) \mapsto \mathbb{R}^{d \times d}$ as

$$\boldsymbol{F} = \nabla_{\boldsymbol{X}} \boldsymbol{\varphi} = \boldsymbol{I} + \nabla_{\boldsymbol{X}} \overline{\boldsymbol{u}}^s \, . \tag{11}$$

DOI: 10.1002/nme

In (11), $\boldsymbol{I}$ denotes the identity tensor in $\mathbb{R}^{d\times d}$. We will also make use of the Green-Lagrange strain tensor $\boldsymbol{E} : \Omega_0^s \times (0,T) \mapsto \mathbb{R}^{d\times d}_{\text{sym}}$ defined by

$$\boldsymbol{E} = \frac{1}{2}\left(\boldsymbol{C} - \boldsymbol{I}\right) \,, \tag{12}$$

where $\boldsymbol{C} : \Omega_0^s \times (0,T) \mapsto \mathbb{R}^{d\times d}_{\text{sym}}$ is called Cauchy-Green deformation tensor and takes on the form $\boldsymbol{C} = \boldsymbol{F}\boldsymbol{F}^T$.

3.3. Governing equations in strong form

In this paper, we assume that both the fluid and the solid are incompressible [‡]. Therefore, the main unknowns of the problem will be a pressure field and the velocity. The main idea of the method is to define global functions $\boldsymbol{v} : \Omega \times (0,T) \mapsto \mathbb{R}^d$, $p : \Omega \times (0,T) \mapsto \mathbb{R}$, such that

$$\boldsymbol{v} = \begin{cases} \boldsymbol{v}^f & \text{on} \quad \Omega_t^f \times (0,T) \\ \boldsymbol{v}^s & \text{on} \quad \Omega_t^s \times (0,T) \end{cases} \quad ; \quad p = \begin{cases} p^f & \text{on} \quad \Omega_t^f \times (0,T) \\ p^s & \text{on} \quad \Omega_t^s \times (0,T) \end{cases} \quad ; \tag{13}$$

where $\boldsymbol{v}^f$ and p^f are the fluid velocity and pressure in Eulerian coordinates, while $\boldsymbol{v}^s$ and p^s are analogous quantities for the solid. The pressure field will be used as a Lagrange multiplier to impose the incompressibility constraint which holds in both the solid and the fluid. The function $\boldsymbol{v}$ is globally continuous due to the no-slip condition at the fluid-solid interface

$$\boldsymbol{v}^f = \boldsymbol{v}^s \quad \text{on} \quad \Gamma_t^I \times (0,T) \,, \tag{14}$$

but the global pressure could, in principle, be discontinuous. Using the functions $\boldsymbol{v}$ and p, taking into account that both fluid and solid are considered to be incompressible, and assuming that gravity is the only external force acting on the system, the linear momentum balance and mass conservation equations of our FSI problem can be written as follows

$$\rho^f \dot{\boldsymbol{v}} = \nabla_{\boldsymbol{x}} \cdot \boldsymbol{\sigma}^f + \rho^f \boldsymbol{g} + \boldsymbol{\mathcal{F}} \qquad \text{in} \quad \Omega \times (0,T) \,, \tag{15}$$

$$\nabla_{\boldsymbol{x}} \cdot \boldsymbol{v} = 0 \qquad \text{in} \quad \Omega \times (0,T) \,, \tag{16}$$

where $\boldsymbol{\mathcal{F}}$ is defined as

$$\boldsymbol{\mathcal{F}} = \begin{cases} 0, & \boldsymbol{x} \in \Omega_t^f \\ \left(\rho^f - \rho^s\right)\left(\dot{\boldsymbol{v}} - \boldsymbol{g}\right) + \nabla_{\boldsymbol{x}} \cdot \left(\boldsymbol{\sigma}^s - \boldsymbol{\sigma}^f\right), & \boldsymbol{x} \in \Omega_t^s \end{cases} \,. \tag{17}$$

In Eqs. (15)–(17), ρ^f and $\boldsymbol{\sigma}^f$ denote the fluid density and Cauchy stress tensor, respectively, while ρ^s and $\boldsymbol{\sigma}^s$ denote the same quantities for the solid. The vector $\boldsymbol{g}$ denotes the acceleration of gravity. To guarantee correct transmission of forces at the fluid-solid interface, we need to impose the constraint

$$\boldsymbol{\sigma}^f \boldsymbol{n}^f = -\boldsymbol{\sigma}^s \boldsymbol{n}^s \quad \text{on} \quad \Gamma_t^I \times (0,T) \,. \tag{18}$$

At this point, we need to introduce a constitutive theory to define the mechanical response of the fluid and the solid. We will assume the fluid to be Newtonian. Therefore,

$$\boldsymbol{\sigma}^f = -p\boldsymbol{I} + 2\mu \nabla_{\boldsymbol{x}}^{\text{sym}} \boldsymbol{v} \,, \tag{19}$$

where $\mu > 0$ is the dynamic viscosity and $\nabla_{\boldsymbol{x}}^{\text{sym}} \boldsymbol{v} = (\nabla_{\boldsymbol{x}} \boldsymbol{v} + \nabla_{\boldsymbol{x}} \boldsymbol{v}^T)/2$. Note that we have extended the definition of $\boldsymbol{\sigma}^f$ to the entire fluid-solid system, even if it does not have a clear physical meaning on Ω_t^s. We consider deformable solids in this paper [§]. The Cauchy stress tensor of the solid is given

[‡] The reader interested in the necessary modifications to account for a compressible solid embedded in an incompressible fluid is remitted to [93].

[§] For the inclusion of rigid bodies in a immersed technique such as the one presented in this manuscript various changes need to be performed. The interested reader is remitted to [94].

by the expression

$$\boldsymbol{\sigma}^s = -p\boldsymbol{I} + \boldsymbol{F}\boldsymbol{S}\boldsymbol{F}^T/J\,, \tag{20}$$

where $J = \det(\boldsymbol{F})$ and $\boldsymbol{S}$ is the second Piola-Kirchhoff stress tensor. We will consider a hyperelastic material, in particular, the Neo-Hookean model with dilatation penalty proposed in [95]. For this particular material type, the second Piola-Kirchhoff stress tensor takes on the form

$$\boldsymbol{S} = \mu^s J^{-2/d}\left(\boldsymbol{I} - \frac{1}{d}\mathrm{tr}(\boldsymbol{C})\boldsymbol{C}^{-1}\right) + \frac{1}{2}\kappa^s\left(J^2 - 1\right)\boldsymbol{C}^{-1}\,, \tag{21}$$

where μ^s is the shear modulus and κ^s is the bulk modulus. Note that the last term of the above equation (called dilatational term) is zero if the solid is incompressible, but we maintain it in the formulation for reasons that will be made clear in Section 4.1. To compute the stress tensor $\boldsymbol{S}$ we need the Lagrangian displacement of the solid $\overline{\boldsymbol{u}}^s(\boldsymbol{X}, t)$, which in our immersed method will be obtained using the equation

$$\frac{\partial\overline{\boldsymbol{u}}^s(\boldsymbol{X}, t)}{\partial t} = \boldsymbol{v}(\boldsymbol{x}, t) \quad \text{if} \quad \boldsymbol{x} = \boldsymbol{\varphi}(\boldsymbol{X}, t) \quad \text{for all} \quad \boldsymbol{X} \in \Omega_0^s\,, \tag{22}$$

which follows from the definition of the global velocity $\boldsymbol{v}$.

To complete the definition of a well-posed initial/boundary-value problem, we need to impose suitable initial and boundary conditions. To define adequate boundary conditions, we split the boundary Γ into non-overlapping sets associated to the fluid boundary and the solid boundary. Thus, we define $\mathcal{B}^s = \Gamma \cap \Gamma_t^s$ and $\mathcal{B}^f = \Gamma \cap \Gamma_t^f$ such that $\Gamma = \overline{\mathcal{B}^s \cup \mathcal{B}^f}$ and $\emptyset = \mathcal{B}^s \cap \mathcal{B}^f$. Note, that in general $\mathcal{B}^s \neq \Gamma_t^s$, because part of Γ_t^s may be immersed into Ω due to the nature of the method. Now, we specify standard boundary conditions for a fluid- and solid-mechanics problem on $\mathcal{B}^s$ and $\mathcal{B}^f$, respectively. For consistency with our hypothesis about the time independence of Ω, we only consider homogeneous displacement boundary conditions on $\mathcal{B}^s$. On the fluid boundary $\mathcal{B}^f$, we assume the typical split between Dirichlet and Neumann boundary conditions, which can be expressed as $\mathcal{B}^f = \overline{\mathcal{B}_D^f \cup \mathcal{B}_N^f}$, with $\mathcal{B}_D^f \cap \mathcal{B}_N^f = \emptyset$. Mathematically, our boundary conditions may be expressed as

$$\boldsymbol{v} = \boldsymbol{v}_B \qquad \text{on} \quad \mathcal{B}_D^f\,, \tag{23}$$

$$\boldsymbol{\sigma}^f\boldsymbol{n}^f = \boldsymbol{h} \qquad \text{on} \quad \mathcal{B}_N^f\,, \tag{24}$$

$$\overline{\boldsymbol{u}}^s = 0 \qquad \text{on} \quad \mathcal{B}^s\,, \tag{25}$$

where $\boldsymbol{v}_B$ is a given velocity and $\boldsymbol{h}$ is a prescribed traction vector. As initial conditions, we set an initial velocity for the entire fluid-solid system. In the solid domain, we also need to solve for displacements, so we have to set initial displacements on Ω_0^s . Therefore, our initial conditions are given by

$$\boldsymbol{v}(\boldsymbol{x}, 0) = \boldsymbol{v}_I(\boldsymbol{x}), \quad \boldsymbol{x} \in \Omega\,, \tag{26}$$

$$\overline{\boldsymbol{u}}^s(\boldsymbol{X}, 0) = 0, \quad \boldsymbol{X} \in \Omega_0^s\,. \tag{27}$$

In summary, the main three equations to be solved in our immersed strategy are the linear momentum balance equation (15), the mass conservation equation (16) and the kinematic relation between the Lagrangian displacement and the Eulerian velocity given by Eq. (22).

3.4. Linear momentum balance and mass conservation equations in weak form

As mentioned before, Eq. (22) will be directly discretized in strong form, but the linear momentum balance equation (15), and the mass conservation equation (16) will be solved using a weighted residual formulation. Thus, before discretization, we need to derive a weak form of Eqs. (15) and (16). To do so, we follow a variational multiscale (VMS) approach proposed in [8]. We will omit the details here, but the main idea is to split the velocity and pressure fields into coarse-scale and

DOI: 10.1002/nme

fine-scale components as $\boldsymbol{v} = \widetilde{\boldsymbol{v}} + \boldsymbol{v}'$ and $p = \widetilde{p} + p'$. Then, we multiply Eqs. (15) and (16) with weight functions $\boldsymbol{w} = \widetilde{\boldsymbol{w}} + \boldsymbol{w}'$ and $q = \widetilde{q} + q'$, integrate over the relevant domain, and perform basic manipulations (see [8] for a detailed derivation). Proceeding as indicated, we obtain the following variational problem: Find $\boldsymbol{v}$ and p such that for all $\widetilde{\boldsymbol{w}}$ and $\widetilde{p}$

$$B^{ED}\left(\{\widetilde{\boldsymbol{w}}, \widetilde{q}\}, \{\widetilde{\boldsymbol{v}} + \boldsymbol{v}', \widetilde{p} + p'\}\right) - L^{ED}\left(\widetilde{\boldsymbol{w}}\right) + B^{LD}\left(\widetilde{\boldsymbol{w}}, \widetilde{\boldsymbol{v}}\right) - L^{LD}\left(\widetilde{\boldsymbol{w}}\right) = 0\,, \tag{28}$$

with

$$\begin{aligned} B^{ED}\left(\{\widetilde{\boldsymbol{w}}, \widetilde{q}\}, \{\widetilde{\boldsymbol{v}} + \boldsymbol{v}', \widetilde{p} + p'\}\right) = &\left(\widetilde{\boldsymbol{w}}, \frac{\partial \widetilde{\boldsymbol{v}}}{\partial t} + \widetilde{\boldsymbol{v}} \cdot \nabla_{\boldsymbol{x}} \widetilde{\boldsymbol{v}}\right)_{\Omega} \\ &- (\nabla_{\boldsymbol{x}} \cdot \widetilde{\boldsymbol{w}}, \widetilde{p})_{\Omega} + (\nabla^{\text{sym}}_{\boldsymbol{x}} \widetilde{\boldsymbol{w}}, 2\nu \nabla^{\text{sym}}_{\boldsymbol{x}} \widetilde{\boldsymbol{v}})_{\Omega} \\ &+ (\widetilde{q}, \nabla_{\boldsymbol{x}} \cdot \widetilde{\boldsymbol{v}})_{\Omega} + (\widetilde{\boldsymbol{w}}, \boldsymbol{v}' \cdot \nabla \widetilde{\boldsymbol{v}})_{\Omega} \\ &- (\nabla_{\boldsymbol{x}} \widetilde{\boldsymbol{w}}, (\widetilde{\boldsymbol{v}} + \boldsymbol{v}') \otimes \boldsymbol{v}')_{\Omega} - (\nabla_{\boldsymbol{x}} \cdot \widetilde{\boldsymbol{w}}, p')_{\Omega} \\ &- (\nabla_{\boldsymbol{x}} \widetilde{q}, \boldsymbol{v}')_{\Omega}\ , \end{aligned} \tag{29}$$

$$L^{ED}\left(\widetilde{\boldsymbol{w}}\right) = \left(\widetilde{\boldsymbol{w}}(\boldsymbol{x}), \boldsymbol{g}\right)_{\Omega} + \left(\widetilde{\boldsymbol{w}}(\boldsymbol{x}), \boldsymbol{h}(\boldsymbol{x})/\rho^f\right)_{\mathcal{B}^f_N}\ , \tag{30}$$

$$\begin{aligned} B^{LD}\left(\widetilde{\boldsymbol{w}}, \widetilde{\boldsymbol{v}}\right) = &- \left(\widetilde{\boldsymbol{w}}(\boldsymbol{\varphi}(\boldsymbol{X},t)), \left(1 - \frac{\rho^s}{\rho^f}\right) \frac{\partial \widetilde{\boldsymbol{v}}(\boldsymbol{\varphi}(\boldsymbol{X},t),t)}{\partial t} J(\boldsymbol{X},t)\right)_{\Omega^s_0} \\ &- \left(\widetilde{\boldsymbol{w}}(\boldsymbol{\varphi}(\boldsymbol{X},t)), \left(1 - \frac{\rho^s}{\rho^f}\right) \widetilde{\boldsymbol{v}}(\boldsymbol{\varphi}(\boldsymbol{X},t),t) \cdot \nabla_{\boldsymbol{x}} \widetilde{\boldsymbol{v}}(\boldsymbol{\varphi}(\boldsymbol{X},t),t) J(\boldsymbol{X},t)\right)_{\Omega^s_0} \\ &- (\nabla^{\text{sym}}_{\boldsymbol{x}} \widetilde{\boldsymbol{w}}(\boldsymbol{\varphi}(\boldsymbol{X},t)), 2\nu \nabla^{\text{sym}}_{\boldsymbol{x}} \widetilde{\boldsymbol{v}}(\boldsymbol{\varphi}(\boldsymbol{X},t),t) J(\boldsymbol{X},t))_{\Omega^s_0} \\ &+ \left(\nabla^{\text{sym}}_{\boldsymbol{x}} \widetilde{\boldsymbol{w}}(\boldsymbol{\varphi}(\boldsymbol{X},t)), \frac{1}{\rho^f} \boldsymbol{F}(\boldsymbol{X},t) \boldsymbol{S}(\boldsymbol{X},t) \boldsymbol{F}^T(\boldsymbol{X},t)\right)_{\Omega^s_0}\ , \end{aligned} \tag{31}$$

$$L^{LD}\left(\widetilde{\boldsymbol{w}}\right) = -\left(\widetilde{\boldsymbol{w}}(\boldsymbol{\varphi}(\boldsymbol{X},t)), \left(1 - \frac{\rho^s}{\rho^f}\right) \boldsymbol{g} J(\boldsymbol{X},t)\right)_{\Omega^s_0}\ . \tag{32}$$

Here, $(\cdot,\cdot)$ denotes the $\mathcal{L}^2$ inner product over the domain indicated with the corresponding subscript, $\otimes$ denotes the outer product, and $\nu = \mu^f/\rho^f$ is the kinematic viscosity. The superscript ED emphasizes that the equation is defined in the Eulerian description whereas the superscript LD indicates the use of the Lagrangian description.

3.5. *Space discretization*

We apply a semi-discrete formulation to our FSI problem. Therefore, we first discretize in space, reducing Eq. (28) to a system of ordinary differential equations that will be eventually integrated in time. To perform space discretization, we need to introduce two discrete spaces associated to the Eulerian and Lagrangian unknowns. In both cases, our discrete spaces will be spanned by linearly independent T-spline basis functions constructed as indicated in Sect. 2. Let us introduce the set of T-spline basis functions $\{N_A(\boldsymbol{x})\}_{A=1}^{n_{ED}}$, which is defined over the physical space Ω and is employed to discretize Eulerian unknowns. Similarly, we define the set of T-spline basis functions $\{\overline{N}_B(\boldsymbol{X})\}_{B=1}^{n_{LD}}$ over the reference configuration of the solid Ω^s_0. Assuming that Dirichlet boundary conditions will be strongly enforced on the discrete space at a later stage, we can write

$$\boldsymbol{v}^h(\boldsymbol{x},t) = \sum_{A=1}^{n_{ED}} \boldsymbol{v}_A(t) N_A(\boldsymbol{x}); \quad p^h(\boldsymbol{x},t) = \sum_{A=1}^{n_{ED}} p_A(t) N_A(\boldsymbol{x}); \tag{33}$$

$$\boldsymbol{w}^h(\boldsymbol{x}) = \sum_{A=1}^{n_{ED}} \boldsymbol{w}_A N_A(\boldsymbol{x}); \quad q^h(\boldsymbol{x}) = \sum_{A=1}^{n_{ED}} q_A N_A(\boldsymbol{x}); \tag{34}$$

DOI: 10.1002/nme

$$\overline{\boldsymbol{u}}^h(\boldsymbol{X},t) = \sum_{B=1}^{n_{LD}} \overline{\boldsymbol{u}}_B(t)\overline{N}_B(\boldsymbol{X}); \tag{35}$$

where $\boldsymbol{v}^h$, p^h, $\boldsymbol{w}^h$, q^h, and $\overline{\boldsymbol{u}}^h$ are the discrete counterparts of $\widetilde{\boldsymbol{v}}$, $\widetilde{p}$, $\widetilde{\boldsymbol{w}}$, $\widetilde{q}$, and $\overline{\boldsymbol{u}}^s$, respectively. A discrete mapping is also defined $\boldsymbol{\varphi}^h(\boldsymbol{X},t)$, which will be used to push forward the Lagrangian mesh to physical space. We will now use the isoparametric concept, that is, we will use the same shape functions to represent the geometry. Thus, the basis functions $\{N_A(\boldsymbol{x})\}_{A=1}^{n_{ED}}$ are used to define a computational mesh on the whole domain Ω. We will refer to this mesh as the Eulerian mesh and it will be utilized to compute the integrals of the weak form (28) on Ω and $\mathcal{B}_N^f$. Analogously, the integrals of the weak form (28) on Ω_0^s will be computed on the mesh defined by the basis functions $\{\overline{N}_B(\boldsymbol{X})\}_{B=1}^{n_{LD}}$. This second mesh, since it is defined in the material description, will be referred to as the Lagrangian mesh. Note that the Lagrangian mesh is arbitrarily located on top of the Eulerian mesh.

Finally, to compute Eq. (28), we need a model for the fine-scale velocity $\boldsymbol{v}'$ and fine-scale pressure p'. Here, we use the standard approach

$$\boldsymbol{v}' = -\,\tau_M \boldsymbol{r}_M\left(\boldsymbol{v}^h, p^h\right)\,, \tag{36}$$

$$p' = -\,\tau_C r_C\left(\boldsymbol{v}^h\right)\,, \tag{37}$$

where τ_M and τ_C are the stabilization parameters defined in [10], and

$$\boldsymbol{r}_M\left(\boldsymbol{v}^h, p^h\right) = \frac{\partial \boldsymbol{v}^h}{\partial t} + \boldsymbol{v}^h\cdot\nabla_{\boldsymbol{x}}\boldsymbol{v}^h + \nabla_{\boldsymbol{x}}p^h - \nu\Delta_{\boldsymbol{x}}\boldsymbol{v}^h - \boldsymbol{g}\,, \tag{38}$$

$$r_C\left(\boldsymbol{v}^h\right) = \nabla_{\boldsymbol{x}}\cdot\boldsymbol{v}^h\,. \tag{39}$$

Replacing continuous with discrete variables in Eq. (28) and using (36)–(37), we obtain the final form of our semi-discrete weak formulation, which may be expressed as: Find $\boldsymbol{v}^h$ and p^h such that, for all $\boldsymbol{w}^h$ and q^h defined as in Eq. (34),

$$B_{MS}^{ED}\left(\{\boldsymbol{w}^h, q^h\},\{\boldsymbol{v}^h, p^h\}\right) - L^{ED}\left(\boldsymbol{w}^h\right) + B^{LD}\left(\boldsymbol{w}^h, \boldsymbol{v}^h\right) - L^{LD}\left(\boldsymbol{w}^h\right) = 0\,, \tag{40}$$

with

$$\begin{aligned} B_{MS}^{ED}\left(\{\boldsymbol{w}^h, q^h\},\{\boldsymbol{v}^h, p^h\}\right) = {} & \left(\boldsymbol{w}^h(\boldsymbol{x}), \frac{\partial \boldsymbol{v}^h(\boldsymbol{x},t)}{\partial t} + \boldsymbol{v}^h(\boldsymbol{x},t)\cdot\nabla_{\boldsymbol{x}}\boldsymbol{v}^h(\boldsymbol{x},t)\right)_\Omega \\ & - \left(\nabla_{\boldsymbol{x}}\cdot\boldsymbol{w}^h(\boldsymbol{x}), p^h(\boldsymbol{x})\right)_\Omega + \left(\nabla_{\boldsymbol{x}}^{\mathrm{sym}}\boldsymbol{w}^h(\boldsymbol{x}), 2\nu\nabla_{\boldsymbol{x}}^{\mathrm{sym}}\boldsymbol{v}^h(\boldsymbol{x},t)\right)_\Omega \\ & + \left(q^h(\boldsymbol{x}), \nabla_{\boldsymbol{x}}\cdot\boldsymbol{v}^h(\boldsymbol{x},t)\right)_\Omega - \left(\boldsymbol{w}^h(\boldsymbol{x}), \tau_M\boldsymbol{r}_M(\boldsymbol{u}^h, p^h)\cdot\nabla\boldsymbol{v}^h\right)_\Omega \\ & + \left(\nabla_{\boldsymbol{x}}\boldsymbol{w}^h(\boldsymbol{x}), (\boldsymbol{v}^h - \tau_M\boldsymbol{r}_M(\boldsymbol{v}^h, p^h))\otimes\tau_M\boldsymbol{r}_M(\boldsymbol{v}^h, p^h)\right)_\Omega \\ & + \left(\nabla_{\boldsymbol{x}}\cdot\boldsymbol{w}^h(\boldsymbol{x}), \tau_C\boldsymbol{r}_C(\boldsymbol{u}^h)\right)_\Omega + \left(\nabla_{\boldsymbol{x}}q^h, \tau_M\boldsymbol{r}_M(\boldsymbol{u}^h, p^h)\right)_\Omega\,. \end{aligned} \tag{41}$$

This completes the space discretization of the linear momentum balance and the mass conservation equations. To finalize the space discretization we also need to discretize Eq. (22). As mentioned before, we will discretize Eq. (22) directly in strong form using a collocation approach. Thus, we replace Eq. (22) with a set of discrete equations given by

$$\frac{\partial \overline{\boldsymbol{u}}^h}{\partial t}(\widetilde{\boldsymbol{\tau}}_i, t) = \boldsymbol{v}^h(\boldsymbol{\varphi}^h(\widetilde{\boldsymbol{\tau}}_i, t), t) \quad \text{for all} \quad \widetilde{\boldsymbol{\tau}}_i \in \widetilde{\mathcal{M}}^h\,, \tag{42}$$

where $\widetilde{\mathcal{M}}^h$ is the set of Greville points in physical space as defined in Sect. 2. Note that Eqs. (40) and (42) are fully coupled and will be solved simultaneously using a monolithic approach. Associated to Eqs. (40) and (42), we define the following residual vectors

$$\boldsymbol{R}^M = \left\{R_{A,i}^M\right\}; \quad \boldsymbol{R}^C = \left\{R_A^C\right\}; \quad \boldsymbol{R}^P = \left\{R_{B,i}^P\right\}; \tag{43}$$

DOI: 10.1002/nme

where $A \in \{1, \dots, n_{ED}\}$ is a control-variable index of the Eulerian mesh, $B \in \{1, \dots, n_{LD}\}$ is a control-variable index of the Lagrangian mesh and i is a dimension index which goes from 1 to d. The components of the residual vectors have the following expressions

$$R^M_{A,i} = B^{ED}_{MS}\left(\{N_A \boldsymbol{e}_i, 0\}, \{\boldsymbol{v}^h, p^h\}\right) - L^{ED}\left(N_A \boldsymbol{e}_i\right) + B^{LD}\left(N_A \boldsymbol{e}_i, \boldsymbol{v}^h\right) - L^{LD}\left(N_A \boldsymbol{e}_i\right) \text{ ,} \tag{44}$$

$$R^C_{A} = B^{ED}_{MS}\left(\{0, N_A\}, \{\boldsymbol{v}^h, p^h\}\right) \text{ ,} \tag{45}$$

$$R^P_{B,i} = \boldsymbol{e}_i \cdot \left[\frac{\partial \overline{\boldsymbol{u}}^h}{\partial t}(\widetilde{\boldsymbol{\tau}}_B, t) - \boldsymbol{v}^h\left(\boldsymbol{\varphi}^h(\widetilde{\boldsymbol{\tau}}_B, t), t\right)\right] \text{ ,} \tag{46}$$

where $\boldsymbol{e}_i$ is the ith vector of the Cartesian basis.

3.6. *Information transfer between the Eulerian and the Lagrangian meshes*

Eq. (46) connects the Lagrangian displacement and the Eulerian velocity at locations given by the Greville points $\{\widetilde{\boldsymbol{\tau}}_B\}_{B=1}^{n_{LD}}$, which live in the solid's reference configuration. These points have been computed from their parametric counterparts $\{\widehat{\boldsymbol{\tau}}_B\}_{B=1}^{n_{LD}}$, which are known and given by the parametric configuration of the Lagrangian mesh. Thus, the term involving $\overline{\boldsymbol{u}}^h$ in Eq. (46) can be readily evaluated using typical procedures of the finite element method. The same argument does not apply to the term involving $\boldsymbol{v}^h$ because it is defined in the Eulerian mesh. According to Eq. (46), $\boldsymbol{v}^h$ has to be evaluated in the physical point $\boldsymbol{x}_B = \boldsymbol{\varphi}^h(\widetilde{\boldsymbol{\tau}}_B, t)$. We know the parametric origin of $\boldsymbol{x}_B$ in the Lagrangian mesh, that is $\widehat{\boldsymbol{\tau}}_B$, but what we need to compute $\boldsymbol{v}^h(\boldsymbol{\varphi}^h(\widetilde{\boldsymbol{\tau}}_B, t), t)$ in a standard finite element fashion is its parametric origin in the Eulerian mesh, which is not known *a priori*. Thus, to evaluate $\boldsymbol{v}^h(\boldsymbol{\varphi}^h(\widetilde{\boldsymbol{\tau}}_B, t), t)$ in Eq. (46), we need to invert the push forward to the physical space of the Eulerian mesh. Note that a similar issue arises when computing the operators B^{LD} and L^{LD} in Eq. (44). These operators involve integrals over the Lagrangian domain which require evaluations of Eulerian functions on quadrature points of the Lagrangian mesh. The parametric origin of these points in the Eulerian mesh is not known either.

Our computations are based on Bézier extraction. Therefore, the mapping from parametric to physical space is local to elements as shown in Eq. (4). As a consequence, if we want to compute the parametric point associated to $\boldsymbol{x}_B$ in the Eulerian mesh, we first need to find the Bézier element in which $\boldsymbol{x}_B$ is located. Bézier elements, in general, will have curvilinear shape, which makes it difficult to locate the element without inverting the mapping. Thus, the most general way to state the problem mathematically is: Find $e \in \{1, \dots, N_{el}\}$ and $\boldsymbol{\xi}_B \in \square$ such that

$$\boldsymbol{x}_B = \sum_{a=1}^{n^e} \boldsymbol{Q}^e_a N^e_a(\boldsymbol{\xi}_B) \text{ ,} \tag{47}$$

where N_{el} is the global number of Bézier elements. Solving Eq. (47) for all Bézier elements would be computationally very intensive. Therefore, in practical dynamic computations we proceed as follows: For a given point $\widetilde{\boldsymbol{\tau}}_B$, we store the Bézier element in which its physical counterpart fell at the previous time step. When we have to find the Bézier element in which it falls in the current time step, we only solve Eq. (47) for a very small set of elements. That reduced set of elements is composed by that which hosted the point in the previous time step and their neighbors.

Remarks:

1. The local geometrical mapping is inverted solving a $d \times d$ nonlinear system using the Newton-Raphson algorithm.
2. When inverting the local geometrical mapping, it is important to check that the solution $\boldsymbol{\xi}_B$ actually satisfies $\boldsymbol{\xi}_B \in \square$. As a matter of fact, Eq. (47) usually has roots outside the parent element and they need to be disregarded.
3. The strategy to transfer information between the Eulerian and the Lagrangian meshes is simpler when using NURBS rather than T-splines. In NURBS-based analysis, the geometrical

mapping is local to patches rather than elements. Therefore, the first step in the process is to find the patch that hosts the physical point. Since the number of patches is typically much lower than the number of elements, the process is simplified. In addition, in many cases of practical interest, NURBS geometrical mappings can be inverted analytically. Within this latter category we can include computations on boxes, cylinders or spheres, for example.

3.7. Time integration

Once space discretization has been sorted, we must now proceed with time discretization. A monolithic and fully-implicit algorithm based on the generalized-α method was proposed by some of the authors in [8] and will be used here. The generalized-α algorithm dates back to 1993, when it was first proposed to solve the structural mechanics equations [85]. The method was later extended to first-order systems in the fluid dynamics domain [86]. As usual, we divide the time interval of interest $[0, T]$ into a sequence of subintervals (t_n, t_{n+1}) with fixed time-step size $\Delta t = t_{n+1} - t_n$. In the following, the time-discrete approximation at t_n of the global vectors of control variables of $\boldsymbol{v}^h$, p^h, and $\frac{\partial \boldsymbol{v}^h}{\partial t}$, will be referred to as $\boldsymbol{V}_n$, $\boldsymbol{P}_n$ and $\boldsymbol{A}_n$, respectively. Analogously, the time-discrete approximation at t_n of the global vectors of control variables of $\overline{\boldsymbol{u}}^h$ and $\frac{\partial \overline{\boldsymbol{u}}^h}{\partial t}$ will be $\overline{\boldsymbol{U}}_n$ and $\overline{\boldsymbol{V}}_n$. We will also introduce $\overline{\boldsymbol{U}}_n^G$ as the global vector that collects the time-discrete approximation of the solid displacements at the Greville points, that is, $\overline{\boldsymbol{U}}_n^G \approx \{\overline{u}_i^h(\widetilde{\boldsymbol{\tau}}_A, t_n)\}_{A=1}^{n_{LD}}$ for all $i = 1, \ldots, d$. In a similar manner, we define $\overline{\boldsymbol{V}}_n^G \approx \{\frac{\partial \overline{u}_i^h}{\partial t}(\widetilde{\boldsymbol{\tau}}_A, t_n)\}_{A=1}^{n_{LD}}$ for all $i = 1, \ldots, d$. Following this notation, our time-integration algorithm may be defined as follows: Given $\boldsymbol{V}_n$, $\boldsymbol{A}_n$, $\overline{\boldsymbol{U}}_n$, and $\overline{\boldsymbol{V}}_n$, find $\boldsymbol{V}_{n+1}$, $\boldsymbol{A}_{n+1}$, $\boldsymbol{V}_{n+\alpha_f}$, $\boldsymbol{A}_{n+\alpha_m}$, $\boldsymbol{P}_{n+1}$, $\overline{\boldsymbol{U}}_{n+1}$, $\overline{\boldsymbol{V}}_{n+1}$, $\overline{\boldsymbol{U}}_{n+\alpha_f}$, and $\overline{\boldsymbol{V}}_{n+\alpha_m}$ such that

$$\boldsymbol{R}^M(\boldsymbol{V}_{n+\alpha_f}, \boldsymbol{A}_{n+\alpha_m}, \boldsymbol{P}_{n+1}, \overline{\boldsymbol{U}}_{n+\alpha_f}) = 0 \,, \tag{48}$$

$$\boldsymbol{R}^C(\boldsymbol{V}_{n+\alpha_f}, \boldsymbol{A}_{n+\alpha_m}, \boldsymbol{P}_{n+1}) = 0 \,, \tag{49}$$

$$\boldsymbol{R}^P(\overline{\boldsymbol{U}}_{n+\alpha_f}^G, \overline{\boldsymbol{V}}_{n+\alpha_m}^G, \boldsymbol{V}_{n+\alpha_f}) = 0 \,, \tag{50}$$

$$\boldsymbol{V}_{n+\alpha_f} = \boldsymbol{V}_n + \alpha_f(\boldsymbol{V}_{n+1} - \boldsymbol{V}_n) \,, \tag{51}$$

$$\boldsymbol{A}_{n+\alpha_m} = \boldsymbol{A}_n + \alpha_m(\boldsymbol{A}_{n+1} - \boldsymbol{A}_n) \,, \tag{52}$$

$$\overline{\boldsymbol{U}}_{n+\alpha_f}^G = \overline{\boldsymbol{U}}_n^G + \alpha_f(\overline{\boldsymbol{U}}_{n+1}^G - \overline{\boldsymbol{U}}_n^G) \,, \tag{53}$$

$$\overline{\boldsymbol{V}}_{n+\alpha_m}^G = \overline{\boldsymbol{V}}_n^G + \alpha_m(\overline{\boldsymbol{V}}_{n+1}^G - \overline{\boldsymbol{V}}_n^G) \,, \tag{54}$$

$$\boldsymbol{V}_{n+1} = \boldsymbol{V}_n + \Delta t((1-\gamma)\boldsymbol{A}_n + \gamma \boldsymbol{A}_{n+1}) \,, \tag{55}$$

$$\overline{\boldsymbol{U}}_{n+1}^G = \overline{\boldsymbol{U}}_n^G + \Delta t((1-\gamma)\overline{\boldsymbol{V}}_n^G + \gamma \overline{\boldsymbol{V}}_{n+1}^G) \,, \tag{56}$$

where α_m, α_f and γ are real-valued parameters that control the accuracy and stability of the algorithm. We will choose here the values proposed by Jansen *et al.* [86] for first order ordinary differential equations. Note also that the vectors $\overline{\boldsymbol{U}}_{n+\alpha_f}$ and $\overline{\boldsymbol{U}}_{n+\alpha_f}^G$ are related through a linear equation. In fact, to obtain the vector of control variables from the values at Greville points, we just need to collocate Eq. (35) at Greville points as

$$\overline{\boldsymbol{u}}^h(\widetilde{\boldsymbol{\tau}}_i, t_{n+\alpha_f}) = \sum_{B=1}^{n_{LD}} \overline{\boldsymbol{u}}_B(t_{n+\alpha_f}) \overline{N}_B(\widetilde{\boldsymbol{\tau}}_i) \quad \text{for all} \quad \widetilde{\boldsymbol{\tau}}_i \in \widetilde{\mathcal{M}}^h \,. \tag{57}$$

Eq. (57) represents a linear system that establishes a correspondence between $\overline{\boldsymbol{U}}_{n+\alpha_f}$ and $\overline{\boldsymbol{U}}_{n+\alpha_f}^G$. The matrix associated to (57) depends on the parametric configuration of the Lagrangian mesh only, making it time independent. Therefore, we just need to compute this matrix once and store it, then we will use it in each Newton-Rapshon of each time step.

Note that (50) is solved for each Greville point independently, that is, we do not need to assemble any matrix for the kinematic equation at each Newton-Raphson iteration of every time step. This is one of the consequences of using our hybrid variational-collocation method. It increases the global efficiency of the methodology by saving computational time especially for higher orders.

4. NUMERICAL EXAMPLES

In this section we present two numerical examples that illustrate the accuracy, stability and robustness of our algorithm. The examples also compare global refinement (NURBS) against local refinement (T-splines) showing the enhanced efficiency of ASTS. The code used to perform these simulations has been developed on top of the scientific library PETSc [96].

4.1. Free-falling cylinder in a channel

Let us consider a channel of infinite length, infinite height, and width $2L$, which contains a viscous fluid. We assume that there is a solid cylinder of infinite length, and radius a immersed in the fluid. The solid is denser than the surrounding fluid, and it is initially at rest. Then, it is allowed to fall freely under the action of gravity. When the solid is released, it will accelerate until the effect of the drag force, which becomes more prominent with velocity, added to buoyancy counterbalance the downward pull of gravity. At that point, the net force acting on the solid will be zero, and the object will move with constant velocity. This velocity is known as the terminal or settling velocity v_T. If we consider the cylinder diameter to be the characteristic length-scale of the problem and the aforementioned terminal speed to be the characteristic velocity, then the relevant Reynolds number is

$$R_e = \frac{2\rho^f v_T a}{\mu} \, . \tag{58}$$

It may be shown that, under the assumption of creeping flow ($R_e << 1$), the terminal velocity may be approximated by the expression

$$v_T = \frac{\left(\rho^s - \rho^f\right) g a^2}{4\mu} \left[\ln\left(\frac{L}{a}\right) - 0.9157 + 1.7244 \left(\frac{a}{L}\right)^2 - 1.7302 \left(\frac{a}{L}\right)^4 \right] , \tag{59}$$

where g is the Euclidean norm of $\boldsymbol{g}$. In this numerical example, we perform a full-scale FSI simulation to compute the solid motion and compare the numerical approximation to the terminal velocity with its theoretical value as given by Eq. (59). The infinite length of the cylinder and the channel allows us to solve this problem in a two-dimensional domain of width $2L$. In the direction of gravity, we take a computational domain which is large enough to produce negligible errors due to its finite size. Our data indicates that the computational domain $\Omega = [0, 2L] \times [0, 4L]$ is sufficiently large to reproduce accurately the results of the free-falling cylinder. In the bottom and lateral boundaries of the box, we impose no-slip conditions, while on the top boundary we prescribe stress-free conditions. In our formulation, the cylinder is modeled as a deformable solid, but we use sufficiently large stiffness moduli in the constitutive theory [Eq. (21)] so that strains are negligible, that is, the solid will just translate in vertical direction as a rigid body. Note that, even if we enforce incompressibility in the formulation, the dilatational term has also been included in (21). Consistent with previous findings by other authors [82], we have noticed that non-negligible errors in the incompressibility constraint arise if the dilatational term is dropped. We have followed [82] in order to fix this issue, that is, we have used relatively small values of κ^s for our computations (of the order of $\mu^s/10$) in order to maintain J close enough to one. The values of the physical parameters used in the computations are indicated in Table I. Introducing these parameters in Eq. (59), we obtain $v_{\mathrm{T}} = 0.9122\ \mathrm{cm/s}$ and the Reynolds number becomes $R_e = 0.09122$, which falls well within the creeping-flow regime.

The time step used in the simulations is $\Delta t = 10^{-3}$ s. We have discretized the problem in space using cubic T-spline shape functions. T-splines, unlike Lagrange piecewise polynomials, can represent the axial section of the cylinder exactly. Additionally, the unstructured nature of T-splines allows such an exact representation without the need to introduce singular points in the parameterization. We note that this is not possible using single-patch NURBS. We use the same Lagrangian mesh in all the simulations of this Section, it is plotted in Fig. 4 along with its T-mesh. The shape functions of the Lagrangian mesh are at least $\mathcal{C}^2$-continuous everywhere except in some Bézier-element boundaries of reduced continuity which are indicated in Fig. 4 (see the caption of

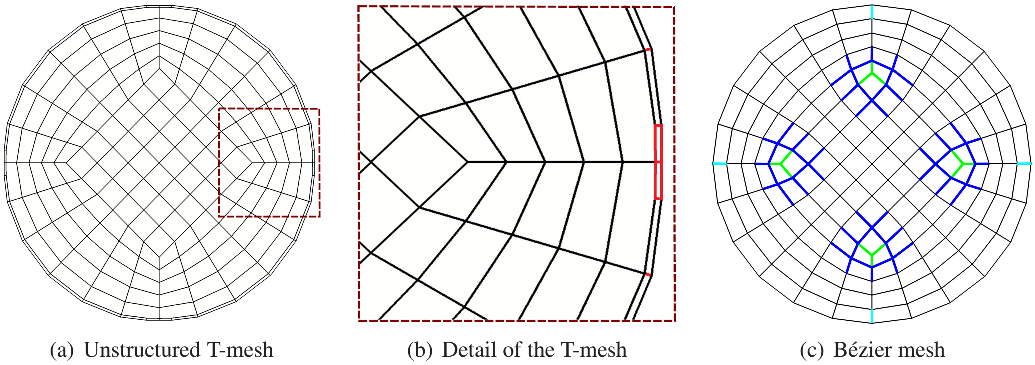

(a) Unstructured T-mesh (b) Detail of the T-mesh (c) Bézier mesh

Figure 4. (Color online) Solid's T-mesh and Bézier mesh for the free-falling cylinder example. (a) T-mesh with four extraordinary nodes which allow us to avoid degenerated points in the parameterization. (b) Zoom-in of the T-mesh close to one of the extraordinary nodes. The color code in the edges of the T-mesh defines the knot interval configuration. All edges have a knot interval of 1 associated, except those colored in red which have associated a zero-length knot interval. (c) Bézier mesh in physical space. The basis functions are $\mathcal{C}^{\infty}$ everywhere, except at the boundaries of the Bézier elements, where the continuity is reduced. The continuity across black Bézier-element boundaries is at least $\mathcal{C}^2$. Dark blue lines represent element boundaries of $\mathcal{C}^1$ continuity due to the presence of extraordinary nodes. Green and light-blue lines represent element boundaries of $\mathcal{C}^0$ continuity due to the presence of extraordinary nodes and our choice to use exact geometry, respectively.

Table I. Values of the relevant physical parameters for the free falling cylinder benchmark problem.

Physical parameter	Value and units
g	$981\ \mathrm{cm/s^2}$
ρ^f	$1.0\ \mathrm{g/cm^3}$
μ	$5\ \mathrm{dyn/cm^2}$
ρ^s	$1.25\ \mathrm{g/cm^3}$
μ^s	$40000\ \mathrm{g/(cm\ s^2)}$
L	$2\ \mathrm{cm}$
a	$0.25\ \mathrm{cm}$

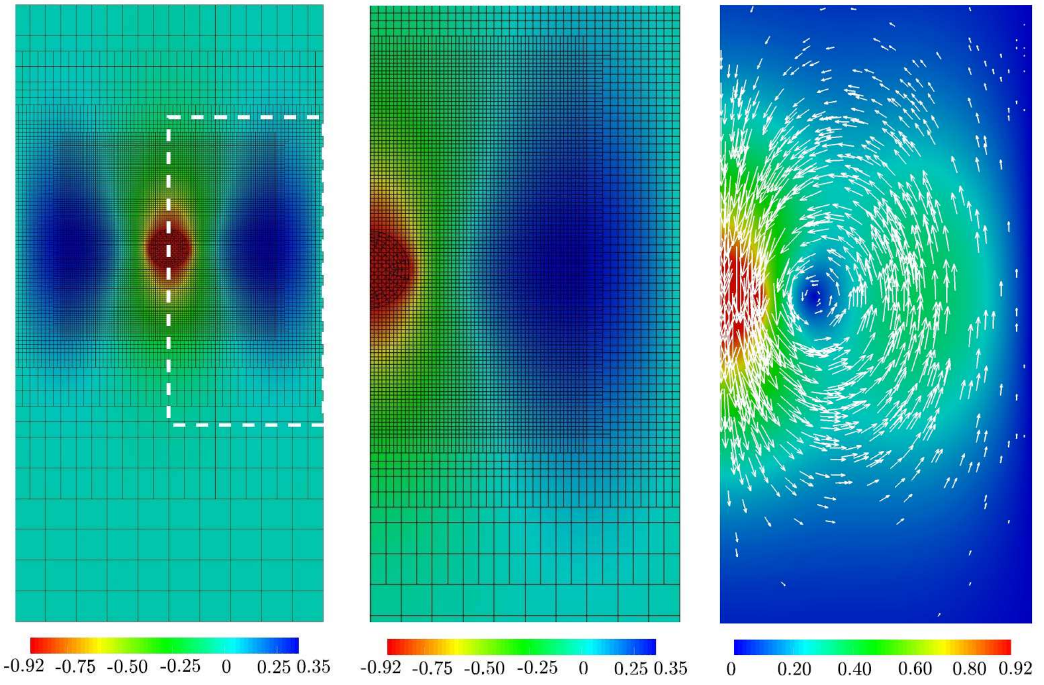
Table II. Different Eulerian meshes considered for the free falling cylinder benchmark problem.

Eulerian mesh	Refinement	Smallest element size	Total number of Bézier elements
ASTS1	Three levels of refinement	0.1	1162
NURBS1	Uniform	0.1	3200
ASTS2	Four levels of refinement	0.05	3868
NURBS2	Uniform	0.05	12800
ASTS3	Five levels of refinement	0.025	14428
NURBS3	Uniform	0.025	51200

Fig. 4 for details). We consider three ASTS Eulerian meshes and three NURBS Eulerian meshes. These meshes are depicted in Table II. Note that the size of the elements in each uniform NURBS mesh is always picked equal to the size of the smallest elements in its corresponding ASTS mesh. This is done in order to make a fair comparison between local and global refinement. The shape functions of all Eulerian meshes are $\mathcal{C}^2$-continuous globally.

The entirety of the ASTS Eulerian mesh with five levels of refinement is shown in Fig. 5 (left panel) superimposed with a color plot of the vertical velocity and the Lagrangian mesh. The area marked with a dashed white line (left panel) has been zoomed in for a clearer observation of the mesh and the solution (middle and right panels). The color code in Fig. 5 indicates the vertical velocity (left and middle panels), and the velocity magnitude (right panel) at time $t = 0.75$ s. In the right panel we have also plotted velocity vectors superimposed on the color plot. It is apparent that there is a large vortex close to the solid.

DOI: 10.1002/nme



(a) Vertical velocity (b) Vertical velocity (zoomed in) (c) Velocity magnitude (zoomed in)

Figure 5. (Color online) (a) Eulerian and Lagrangian meshes along with a color plot of the vertical velocity. The Eulerian mesh is locally refined where the flow is most complex. Local refinement permitted us to reduce by 68% the number of Bézier elements with respect to a uniform mesh that achieves the same accuracy. (b) Zoom in of the inset area in the left panel. (c) Velocity magnitude in the inset area. The color plot is superimposed with velocity vectors, which show the formation of a large vortex close to the solid.

We have also computed the time evolution of the average vertical velocity of the cylinder for each Eulerian mesh. In Fig. 6, it can be seen how the numerical terminal velocity gets closer to its theoretical value as we increase the spatial resolution for both ASTS and NURBS meshes. The globally and locally refined meshes with the same element size give essentially the same solution which shows the potential of ASTS in order to enhance the algorithm's efficiency. The relative error for the meshes ASTS3 and NURBS3 is below 0.2%, which seems to be similar to the error in the theoretical solution given the number of digits employed in the constants in Eq. (59). Note that the mesh ASTS3 has 72% less Bézier elements than the mesh NURBS3 which exhibits the importance of local refinement. Additionally, we would like to point out that only twenty Bézier elements are behind the solid for the meshes ASTS1 and NURBS1 which is clearly not enough to give a good solution.

A peculiar feature of immersed FSI methods that use Lagrangian meshes arbitrarily embedded into a background Eulerian mesh is that the finite element spaces ensued from the meshes might have to comply with an inf-sup type condition to achieve stability [97, 98]. In the early stages of the IFEM, which is a particular case of our method for classical $\mathcal{C}^0$ finite elements, it was reported that the elements of the Lagrangian mesh had to be at least two times smaller than those of the Eulerian mesh. In our earlier work using globally $\mathcal{C}^1$-continuous basis functions we showed that could get stable and accurate results working with the same element size in both meshes [8]. Here, we show that the adoption of globally $\mathcal{C}^2$-continuous basis functions permits the use of solid elements two times larger than those of the background Eulerian mesh (see the meshes in the middle panel of Fig. 5). These data suggest that increasing the global continuity may alleviate the inf-sup-type condition that the solid and fluid spaces have to satisfy. Although we acknowledge that we do not have a mathematical proof of this statement at this point, we do believe that this observation warrants further investigation.

DOI: 10.1002/nme

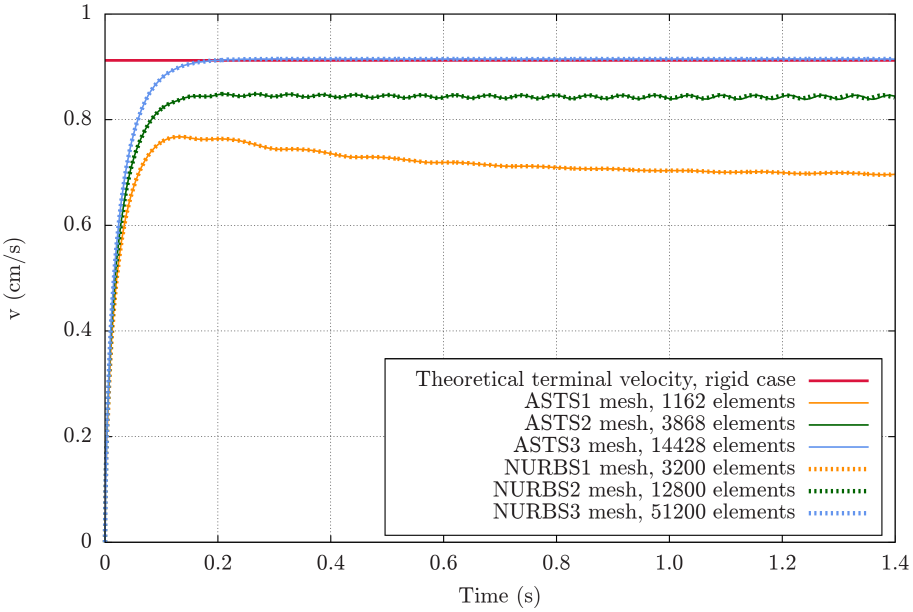


Figure 6. (Color online) Time evolution of the average cylinder velocity for each Eulerian mesh compared against the theoretical terminal velocity (red line). The Lagrangian mesh is the same in all simulations and it is composed of 156 Bézier elements and 209 control points. The ASTS Eulerian meshes have been locally refined where the physics is more interesting without perceptible lost of accuracy from their globally refined NURBS meshes counterparts. Excellent agreement is obtained with respect to the theoretical terminal velocity when we use enough spatial resolution. Cubic basis functions have been used for all the meshes.

Another important aspect is that the computation of the residual requires integrating on the Lagrangian mesh fields which live in the space defined by the Eulerian mesh. This leads to suboptimal quadrature because we are integrating on the elements of the Lagragian mesh, but some of the fields we integrate have lines of reduced continuity in those integration regions [99]. In classical finite elements, those lines have $\mathcal{C}^0$ continuity. However, the continuity of those lines is increased to $\mathcal{C}^2$ using cubic T-splines in the Eulerian mesh. It seems obvious that increasing the regularity of those lines of reduced continuity will lead to more accurate integration and better results.

In all, this example shows that, for this particular type of algorithms, it is important to be able to use local refinement *and* higher order global continuity simultaneously, which is very difficult to achieve without the use of T-spline-based isogeometric analysis.

4.2. Flow past a deformable solid with one end fixed

$$R_e = \frac{\rho^f b\, v_{if}}{\mu}\ , \tag{60}$$

where v_{if} is the inflow velocity and b is the length of the inflow boundary of the channel. As shown in Fig. 7 and Table III, $b = 2$ cm. Using the remaining data in Table III, we find that $R_e = 800$.

Let us consider the two-dimensional fluid-solid system depicted in Fig. 7, where lengths are expressed in centimeters. The green rectangle in the plot represents a deformable solid of size $0.8\ \mathrm{cm} \times 0.2\ \mathrm{cm}$. The bottom of the solid is fixed to the channel wall. The system is driven by a uniform inflow velocity at the left boundary of the domain. The outlet on the right hand side is subjected to a zero-traction boundary condition, and the top and bottom boundaries are assumed to satisfy no-slip boundary conditions. The values of the relevant physical parameters are indicated in Table III. We assume that the relevant Reynolds number for this problem is

As in the above example, three cubic ASTS and NURBS Eulerian meshes are considered in order to do comparisons. The ASTS Eulerian meshes are locally refined where the flow is supposed to

DOI: 10.1002/nme

Table III. Values of the relevant physical parameters for the flow past a solid example.

Physical parameter	Value and units
g	$0\ \mathrm{cm/s^2}$
ρ^f	$1.0\ \mathrm{g/cm^3}$
μ	$0.02\ \mathrm{dyn/cm^2}$
ρ^s	$1.5\ \mathrm{g/cm^3}$
μ^s	$8000\ \mathrm{g/(cm\ s^2)}$
v_{if}	$8\ \mathrm{cm/s}$
b	$2\ \mathrm{cm}$

Table IV. Different Eulerian meshes considered for the flow past a solid example.

Eulerian mesh	Refinement	Smallest element size	Total number of Bézier elements
ASTS1	Three levels of refinement	0.05	2154
NURBS1	Uniform	0.05	6400
ASTS2	Three levels of refinement	0.025	7675
NURBS2	Uniform	0.025	25600
ASTS3	Three levels of refinement	0.0125	29549
NURBS3	Uniform	0.0125	102400

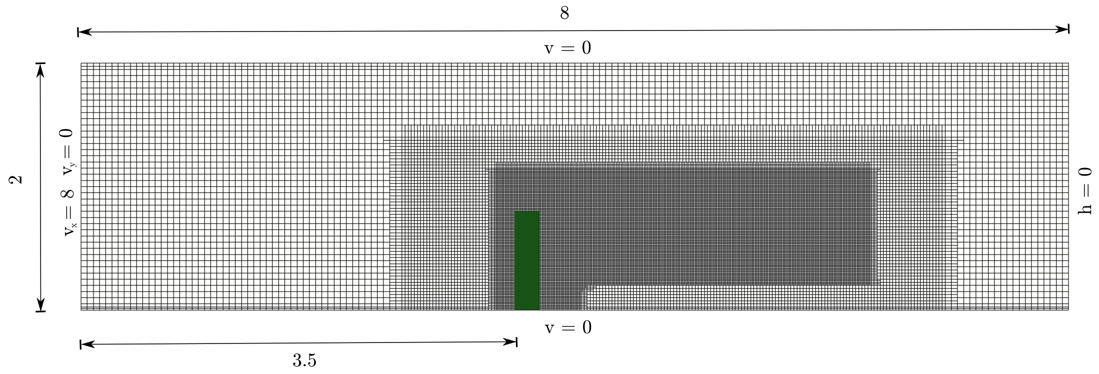


Figure 7. (Color online) Flow past a solid. Geometry, boundary conditions, initial position of the solid, and Eulerian mesh. The introduction of T-junctions allows us to locally refine the Eulerian mesh at the boundary layers and downstream the solid. There are three levels of refinement. The solid is a rectangle of size $0.2\ \mathrm{cm} \times 0.8\ \mathrm{cm}$ and its bottom end is fixed. The solid mesh is uniform. The lengths marked in the plot are expressed in centimeters and the velocities in centimeters per second.

be more complex due to the presence of the solid. Three levels of refinement are introduced in all ASTS meshes with different spatial resolutions. The main features of the ASTS and NURBS Eulerian meshes are described in Table IV. The mesh ASTS3 is shown in Fig. 7. The Lagrangian mesh used in our simulation is uniform, with 25×100 Bézier elements and 2884 control points. The basis functions associated to both meshes are $\mathcal{C}^2$-continuous globally. The time step used in the simulation is $\Delta t = 2 \cdot 10^{-4}$ s.

Fig. 8 shows the solution at time $t = 0.193$ s using the mesh ASTS3. The deformation of the solid is apparent. The color scale on the fluid represents the pressure and the superimposed arrows correspond to the velocity field. As expected, there is a large vortex downstream the solid, located in the low-pressure area. Fig. 9 shows the time evolution of the horizontal and vertical displacements of the top left corner of the solid for different Eulerian meshes. The displacement curves barely vary from NURBS2 to NURBS3, which suggests that NURBS3 is a converged result. The mesh ASTS3 gives essentially the same solution than the mesh NURBS3, but it has 71% less Bézier elements which evidences the advantages of the local refinement capability of T-splines. Finally, the results of Fig. 9 suggest that a steady configuration has been reached.

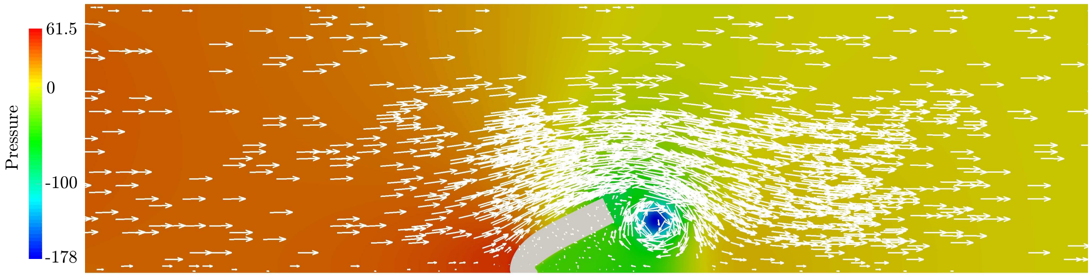


Figure 8. (Color online) Flow past a solid. Pressure field color plot superimposed with arrows that represent the velocity at time $t = 0.193$ s. The plot clearly shows a vortex downstream the solid which coincides with the lowest-pressure area.

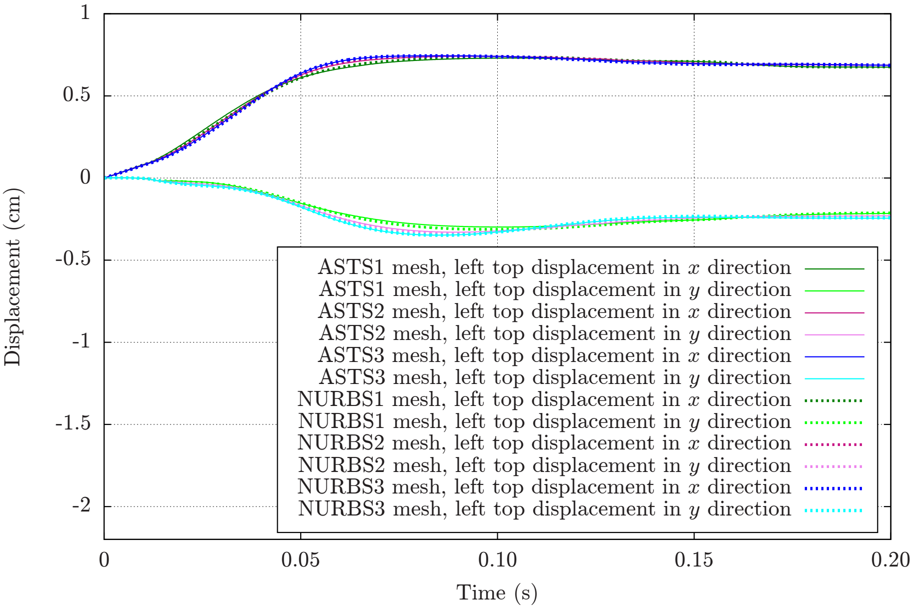


Figure 9. (Color online) Flow past a solid. Time evolution of the displacement in the x and y directions of the left top corner of the solid for the different Eulerian meshes considered. The Lagrangian mesh is the same in all simulations and it is composed of 2500 Bézier elements and 2884 control points. The ASTS Eulerian meshes have been locally refined close and after the solid. The meshes NURBS2 and NURBS3 give very similar results despite of the much higher spatial resolution used in NURBS3, which indicates that enough spatial resolution has been used in the mesh NURBS3. The mesh ASTS3 provides basically the same displacement curves than NURBS3, however, it has 71% less Bézier elements thanks to the use of local refinement.

5. CONCLUSIONS

We proposed a hybrid variational-collocation immersed fluid-structure interaction method which uses T-spline-based isogeometric analysis for both the fluid and the solid. The information transfer between the Eulerian and the Lagrangian mesh is carried out using the T-splines shape functions as well. T-splines bring a number of possibilities to the fluid-structure interaction field thanks to their local refinement capability and their unstructured nature which allows to represent complex geometries. At the same time, T-splines maintain all the advantages that NURBS have against other types of shape functions such as high inter-element continuity. Higher-order global continuity was shown to increase the accuracy per degree of freedom in solid and fluid mechanics and we showed that it is also beneficial for immersed FSI. Indeed, increasing the inter-element continuity of the

DOI: 10.1002/nme

discrete spaces alleviates previously-reported constraints on the element size of the Lagrangian and Eulerian meshes. We have validated our computational framework solving a common FSI benchmark problem with theoretical solution. We took advantage of T-junctions to perform local refinement and extraordinary nodes to avoid singular points. Our results were in very good agreement with theoretical solutions, which confirms the feasibility of our methodology.

ACKNOWLEDGEMENTS

This work was accomplished during a four-month visit of HC to Carnegie Mellon University (CMU) in 2014, for which financial support from the PhD student grant UDC-Inditex is gratefully acknowledged. HG, HC and CB-C were partially supported by the European Research Council through the FP7 Ideas Starting Grant (project # 307201). HG was also partially supported by Ministerio de Economía y Competitividad (contract # DPI2013-44406-R) cofinanced with FEDER funds. LL and YZ were partially supported by the PECASE Award N00014-14-10234. We acknowledge the open source scientific library PETSc and their developers.

REFERENCES

1. Hughes TJR, Cottrell JA, Bazilevs Y. Isogeometric analysis CAD, finite elements, NURBS, exact geometry and mesh refinement. *Computacional Methods in Applied Mechanics and Engineering* 2005; **194**:4135–4195.
2. Cottrell JA, Hughes TJR, Bazilevs Y. *Isogeometric Analysis Toward Integration of CAD and FEA*. Wiley, 2009.
3. Bazilevs Y, Calo VM, Hughes TJR, Zhang Y. Isogeometric fluid-structure interaction: Theory, algorithms, and computations. *Computational Mechanics* 2008; **43**:3–37.
4. Bazilevs Y, Calo VM, Zhang Y, Hughes TJR. Isogeometric fluid-structure interaction analysis with applications to arterial blood flow. *Computational Mechanics* 2006; **38**:310–322.
5. Zhang Y, Bazilevs Y, Goswami S, Bajaj C, Hughes TJR. Patient-specific vascular NURBS modeling for isogeometric analysis of blood flow. *Computer Methods in Applied Mechanics and Engineering* 2007; **196**:2943–2959.
6. Bueno J, Bona-Casas C, Bazilevs Y, Gomez H. Interaction of complex fluids and solids: theory, algorithms and application to phase-change-driven implosion. *Computational Mechanics* 2014; :1–14.
7. Kamensky D, Hsu MC, Schillinger D, Evans JA, Aggarwal A, Bazilevs Y, Sacks MS, Hughes TJR. An immersogeometric variational framework for fluid-structure interaction: Application to bioprosthetic heart valves. *Computer Methods in Applied Mechanics and Engineering* 2015; **284**:1005–1053.
8. Casquero H, Bona-Casas C, Gomez H. A NURBS-based immersed methodology for fluid-structure interaction. *Computer Methods in Applied Mechanics and Engineering* 2015; **284**:943–970.
9. Hsu MC, Kamensky D, Bazilevs Y, Sacks M, Hughes TJR. Fluid-structure interaction analysis of bioprosthetic heart valves: Significance of arterial wall deformation. *Computational Mechanics* 2014; **54**:1055–1071.
10. Bazilevs Y, Calo VM, Cottrell JA, Hughes TJR, Reali A, Scovazzi G. Variational multiscale residual-based turbulence modeling for large eddy simulation of incompressible flows. *Computer Methods in Applied Mechanics and Engineering* 2007; **197**:173–201.
11. Evans JA, Hughes TJR. Isogeometric divergence-conforming B-splines for the unsteady Navier–Stokes equations. *Journal of Computational Physics* 2013; **241**:141–167.
12. Gomez H, Cueto-Felgueroso L, Juanes R. Three-dimensional simulation of unstable gravity-driven infiltration of water into a porous medium. *Journal of Computational Physics* 2013; **238**:217–239.
13. Gomez H, Hughes TJR, Nogueira X, Calo VM. Isogeometric analysis of the isothermal Navier–Stokes–Korteweg equations. *Computer Methods in Applied Mechanics and Engineering* 2010; **199**:1828–1840.
14. Liu J, Gomez H, Evans JA, Hughes TJR, Landis CM. Functional entropy variables: A new methodology for deriving thermodynamically consistent algorithms for complex fluids, with particular reference to the isothermal Navier–Stokes–Korteweg equations. *Journal of Computational Physics* 2013; **248**:47–86.
15. Gomez H, Nogueira X. A new space–time discretization for the Swift–Hohenberg equation that strictly respects the Lyapunov functional. *Communications in Nonlinear Science and Numerical Simulation* 2012; **17**:4930–4946.
16. Gomez H, Calo VM, Bazilevs Y, Hughes TJR. Isogeometric analysis of the Cahn-Hilliard phase-field model. *Computacional Methods in Applied Mechanics and Engineering* 2008; **197**:4333–4352.
17. Gomez H, Hughes TJR. Provably unconditionally stable, second-order time-accurate, mixed variational methods for phase-field models. *Journal of Computational Physics* 2011; **230**:5310–5327.
18. Gomez H, Nogueira X. An unconditionally energy-stable method for the phase field crystal equation. *Computer Methods in Applied Mechanics and Engineering* 2012; **249**:52–61.
19. Thiele U, Archer AJ, Robbins MJ, Gomez H, Knobloch E. Localized states in the conserved Swift-Hohenberg equation with cubic nonlinearity. *Physical Review E* 2013; **87**(4):042 915.
20. Gomez H, París J. Numerical simulation of asymptotic states of the damped Kuramoto-Sivashinsky equation. *Physical Review E* 2011; **83**:046 702.

DOI: 10.1002/nme

21. Vilanova G, Colominas I, Gomez H. Capillary networks in tumor angiogenesis: From discrete endothelial cells to phase-field averaged descriptions via isogeometric analysis. *International Journal for Numerical Methods in Biomedical Engineering* 2013; **29**:1015–1037.
22. Vilanova G, Colominas I, Gomez H. Coupling of discrete random walks and continuous modeling for three-dimensional tumor-induced angiogenesis. *Computational Mechanics* 2013; **53**:449–464.
23. Cottrell JA, Reali A, Bazilevs Y, Hughes TJR. Isogeometric analysis of structural vibrations. *Computer Methods in Applied Mechanics and Engineering* 2006; **195**:5257–5296.
24. Hughes TJR, Reali A, Sangalli G. Duality and unified analysis of discrete approximations in structural dynamics and wave propagation: Comparison of p-method finite elements with k-method NURBS. *Computer methods in applied mechanics and engineering* 2008; **197**:4104–4124.
25. Lipton S, Evans J, Bazilevs Y, Elguedj T, Hughes TJR. Robustness of isogeometric structural discretizations under severe mesh distortion. *Computer Methods in Applied Mechanics and Engineering* 2010; **199**:357–373.
26. Elguedj T, Bazilevs Y, Calo VM, Hughes TJRH. $\overline{B}$ and $\overline{F}$ projection methods for nearly incompressible linear and non-linear elasticity and plasticity using higher-order NURBS elements. *Computer methods in applied mechanics and engineering* 2008; **197**:2732–2762.
27. Dhote R, Gomez H, Melnik R, Zu J. Isogeometric analysis of a dynamic thermo-mechanical phase-field model applied to shape memory alloys. *Computational Mechanics* 2013; **53**:1235–1250.
28. Dhote R, Gomez H, Melnik R, Zu J. Shape memory alloy nanostructures with coupled dynamic thermo-mechanical effects. *Computer Physics Communications* 2015; (0):–.
29. Dhote R, Gomez H, Melnik R, Zu J. 3D coupled thermo-mechanical phase-field modeling of shape memory alloy dynamics via isogeometric analysis. *Computers & Structures* 2015; (0):–.
30. Kiendl J, Bletzinger KU, Linhard J, Wuchner R. Isogeometric shell analysis with Kirchhoff-Love elements. *Computer Methods in Applied Mechanics and Engineering* 2009; **198**:3902–3914.
31. Kiendl J, Bazilevs Y, Hsu MC, Wüchner R, Bletzinger KU. The bending strip method for isogeometric analysis of Kirchhoff–Love shell structures comprised of multiple patches. *Computer Methods in Applied Mechanics and Engineering* 2010; **199**:2403–2416.
32. Nguyen-Thanh N, Kiendl J, Nguyen-Xuan H, Wuchner R, Bletzinger K, Bazilevs Y, Rabczuk T. Rotation free isogeometric thin shell analysis using PHT-splines. *Computer Methods in Applied Mechanics and Engineering* 2011; **200**:3410–3424.
33. Kiendl J, Hsu MC, Wu MC, Reali A. Isogeometric kirchhoff–love shell formulations for general hyperelastic materials. *Computer Methods in Applied Mechanics and Engineering* 2015; .
34. De Lorenzis L, Temizer I, Wriggers P, Zavarise G. A large deformation frictional contact formulation using NURBS-based isogeometric analysis. *International Journal for Numerical Methods in Engineering* 2011; **87**:1278–1300.
35. Dimitri R, De Lorenzis L, Scott M, Wriggers P, Taylor R, Zavarise G. Isogeometric large deformation frictionless contact using T-splines. *Computer Methods in Applied Mechanics and Engineering* 2014; **269**:394–414.
36. Dimitri R, De Lorenzis L, Wriggers P, Zavarise G. NURBS-and T-spline-based isogeometric cohesive zone modeling of interface debonding. *Computational Mechanics* 2014; **54**:369–388.
37. Sederberg TW, Zheng J, Bakenov A, Nasri A. T-splines and T-NURCCs. *ACM Trans. Graph.* 2003; **22**:477–484.
38. Zhang Y, Wang W, Hughes TJR. Solid T-spline construction from boundary representations for genus-zero geometry. *Computer Methods in Applied Mechanics and Engineering* 2012; **249-252**:185–197.
39. Zhang Y, Wang W, Hughes TJRH. Conformal solid T-spline construction from boundary T-spline representations. *Computational Mechanics* 2013; **51**:1051–1059.
40. Wang W, Zhang Y, Liu L, Hughes TJR. Trivariate solid T-spline construction from boundary triangulations with arbitrary genus topology. *Computer-Aided Design* 2013; **45**(2):351–360.
41. Liu L, Zhang Y, Hughes TJR, Scott MA, Sederberg TW. Volumetric T-spline construction using boolean operations. *Engineering with Computers* 2014; **30**:425–439.
42. Liu L, Zhang Y, Liu Y, Wang W. Feature-preserving t-mesh construction using skeleton-based polycubes. *Computer-Aided Design* 2015; **58**:162–172.
43. Bazilevs Y, Calo V, Cottrell J, Evans J, Hughes TJR, Lipton S, Scott M, Sederberg T. Isogeometric analysis using T-splines. *Computer Methods in Applied Mechanics and Engineering* 2010; **199**:229–263.
44. Li X, Zheng J, Sederberg TW, Hughes TJR, Scott MA. On linear independence of T-spline blending functions. *Computer Aided Geometric Design* 2012; **29**:63–76.
45. Scott M, Simpson R, Evans J, Lipton S, Bordas S, Hughes TJR, Sederberg T. Isogeometric boundary element analysis using unstructured T-splines. *Computer Methods in Applied Mechanics and Engineering* 2013; **254**:197–221.
46. Scott MA, Borden MJ, Verhoosel CV, Sederberg TW, Hughes TJR. Isogeometric finite element data structures based on Bézier extraction of T-splines. *International Journal for Numerical Methods in Engineering* 2011; **88**:126–156.
47. Borden MJ, Scott MA, Evans JA, Hughes TJR. Isogeometric finite element data structures based on Bézier extraction of NURBS. *International Journal for Numerical Methods in Engineering* 2011; **87**(1-5):15–47.
48. Scott M, Li X, Sederberg T, Hughes TJR. Local refinement of analysis-suitable T-splines. *Computer Methods in Applied Mechanics and Engineering* 2012; **213-216**:206–222.
49. Wang W, Zhang Y, Scott MA, Hughes TJR. Converting an unstructured quadrilateral mesh to a standard T-spline surface. *Computational Mechanics* 2011; **48**:477–498.
50. Liu L, Zhang YJ, Wei X. Weighted t-splines with application in reparameterizing trimmed {NURBS} surfaces. *Computer Methods in Applied Mechanics and Engineering* 2015; **295**:108 – 126, doi:http://dx.doi.org/10.1016/j.cma.2015.06.020. URL `http://www.sciencedirect.com/science/article/pii/S0045782515002121`.
51. Evans E, Scott M, Li X, Thomas D. Hierarchical T-splines: Analysis-suitability, Bézier extraction, and application as an adaptive basis for isogeometric analysis. *Computer Methods in Applied Mechanics and Engineering* 2015; **284**:1–20.

52. Auricchio F, Da Veiga L, Hughes TJR, Reali A, Sangalli G. Isogeometric collocation methods. *Mathematical Models and Methods in Applied Sciences* 2010; **20**:2075–2107.
53. Gomez H, Reali A, Sangalli G. Accurate, efficient, and (iso)geometrically flexible collocation methods for phase-field models. *Journal of Computational Physics* 2014; **262**:153–171.
54. Schillinger D, Evans J, Reali A, Scott M, Hughes TJR. Isogeometric collocation: Cost comparison with Galerkin methods and extension to adaptive hierarchical NURBS discretizations. *Computer Methods in Applied Mechanics and Engineering* 2013; **267**:170–232.
55. Reali A, Gomez H. An isogeometric collocation approach for Bernoulli-Euler beams and Kirchhoff plates. *Computer Methods in Applied Mechanics and Engineering* 2015; **284**:623–636.
56. Casquero H, Lei L, Zhang J, Reali A, Gomez H. Isogeometric collocation using analysis-suitable T-splines of arbitrary degree 2015; Submitted for publication.
57. Borden M, Verhoosel C, Scott M, Hughes TJR, Landis C. A phase-field description of dynamic brittle fracture. *Computer Methods in Applied Mechanics and Engineering* 2012; **217-220**:77–95.
58. Verhoosel CV, Scott MA, de Borst R, Hughes TJR. An isogeometric approach to cohesive zone modeling. *International Journal for Numerical Methods in Engineering* 2011; **87**:336–360.
59. Verhoosel CV, Scott MA, Hughes TJR, de Borst R. An isogeometric analysis approach to gradient damage models. *International Journal for Numerical Methods in Engineering* 2011; **86**:115–134.
60. Dimitri R, Lorenzis LD, Scott M, Wriggers P, Taylor R, Zavarise G. Isogeometric large deformation frictionless contact using T-splines. *Computer Methods in Applied Mechanics and Engineering* 2014; **269**:394–414.
61. Simpson R, Scott M, Taus M, Thomas D, Lian H. Acoustic isogeometric boundary element analysis. *Computer Methods in Applied Mechanics and Engineering* 2014; **269**:265–290.
62. Bazilevs Y, Hsu MC, Scott M. Isogeometric fluid-structure interaction analysis with emphasis on non-matching discretizations, and with application to wind turbines. *Computer Methods in Applied Mechanics and Engineering* 2012; **249-252**:28–41.
63. Hughes TJR, Liu WK, Zimmermann TK. Lagrangian-Eulerian finite element formulation for incompressible viscous flows. *Computer methods in applied mechanics and engineering* 1981; **29**:329–349.
64. Bazilevs Y, Hsu MC, Zhang Y, Wang W, Kvamsdal T, Hentschel S, Isaksen J. Computational vascular fluid-structure interaction: Methodology and application to cerebral aneurysms. *Biomechanics and Modeling in Mechanobiology* 2010; **9**:481–498.
65. Bazilevs Y, Gohean J, Hughes TJR, Moser R, Zhang Y. Patient-specific isogeometric fluid-structure interaction analysis of thoracic aortic blood flow due to implantation of the Jarvik 2000 left ventricular assist device. *Computer Methods in Applied Mechanics and Engineering* 2009; **198**:3534–3550.
66. Isaksen JG, Bazilevs Y, Kvamsdal T, Zhang Y, Kaspersen JH, Waterloo K, Romner B, Ingebrigtsen T. Determination of wall tension in cerebral artery aneurysms by numerical simulation. *Stroke* 2008; **39**:3172–3178.
67. Bazilevs Y, Hsu MC, Zhang Y, Wang W, Liang X, Kvamsdal T, Brekken R, Isaksen J. A fully-coupled fluid-structure interaction simulation of cerebral aneurysms. *Computational Mechanics* 2010; **46**(1):3–16.
68. Zhang Y, Wang W, Liang X, Bazilevs Y, Hsu MC, Kvamsdal T, Brekken R, Isaksen J. High-fidelity tetrahedral mesh generation from medical imaging data for fluid-structure interaction analysis of cerebral aneurysms. *Computer Modeling in Engineering and Sciences (CMES)* 2009; **42**(2):131.
69. Bazilevs Y, Hsu MC, Akkerman I, Wright S, Takizawa K, Henicke B, Spielman T, Tezduyar T. 3D simulation of wind turbine rotors at full scale. part I: Geometry modeling and aerodynamics. *International Journal for Numerical Methods in Fluids* 2011; **65**:207–235.
70. Bazilevs Y, Hsu MC, Kiendl J, Wuchner R, Bletzinger KU. 3D simulation of wind turbine rotors at full scale. part ii: Fluid-structure interaction modeling with composite blades. *International Journal for Numerical Methods in Fluids* 2011; **65**:236–253.
71. Farhat C, van der Zee KG, Geuzaine P. Provably second-order time-accurate loosely-coupled solution algorithms for transient nonlinear computational aeroelasticity. *Computer methods in applied mechanics and engineering* 2006; **195**:1973–2001.
72. Tezduyar TE, Takizawa K, Moorman C, Wright S, Christopher J. Space-time finite element computation of complex fluid-structure interactions. *International Journal for Numerical Methods in Fluids* 2010; **64**:1201–1218.
73. van Brummelen EH, van der Zee KG, de Borst R. Space/time multigrid for a fluid-structure-interaction problem. *Applied Numerical Mathematics* 2008; **58**:1951–1971.
74. Bazilevs Y, Takizawa K, Tezduyar TE. *Computational fluid-structure interaction: methods and applications*. John Wiley & Sons, 2012.
75. Peskin C. Flow patterns around heart valves: A numerical method. *Journal of Computational Physics* 1972; **10**:252–271.
76. McQueen D, Peskin C. Computer-assisted design of butterfly bileaflet valves for the mitral position. *Scandinavian Journal of Thoracic and Cardiovascular Surgery* 1985; **19**:139–148.
77. Beyer Jr R. A computational model of the cochlea using the immersed boundary method. *Journal of Computational Physics* 1992; **98**:145–162.
78. Dillon R, Fauci L, Fogelson A, Gaver III D. Modeling biofilm processes using the immersed boundary method. *Journal of Computational Physics* 1996; **129**:57–73.
79. Gil AJ, Carreño AA, Bonet J, Hassan O. The immersed structural potential method for haemodynamic applications. *Journal of Computational Physics* 2010; **229**(22):8613–8641.
80. Mountrakis L, Lorenz E, Hoekstra A. Where do the platelets go? A simulation study of fully resolved blood flow through aneurysmal vessels. *Interface focus* 2013; **3**(2):20120 089.
81. Mountrakis L, Lorenz E, Malaspinas O, Alowayyed S, Chopard B, Hoekstra A. Parallel performance of an IB-LBM suspension simulation framework. *Journal of Computational Science* 2015; .
82. Zhang L, Gerstenberger A, Wang X, Liu WK. Immersed finite element method. *Computer Methods in Applied Mechanics and Engineering* 2004; **193**:2051–2067.

83. Liu WK, Liu Y, Farrell D, Zhang L, Wang X, Fukui Y, Patankar N, Zhang Y, Bajaj C, Lee J, *et al.*. Immersed finite element method and its applications to biological systems. *Computer Methods in Applied Mechanics and Engineering* 2006; **195**:1722–1749.
84. Gay M, Zhang L, Liu WK. Stent modeling using immersed finite element method. *Computer Methods in Applied Mechanics and Engineering* 2006; **195**:4358–4370.
85. Chung J, Hulbert G. A time integration algorithm for structural dynamics with improved numerical dissipation: The generalized- method. *Journal of Applied Mechanics* 1993; **60**:371–375.
86. Jansen K, Whiting C, Hulbert G. Generalized-α method for integrating the filtered navier-stokes equations with a stabilized finite element method. *Computer Methods in Applied Mechanics and Engineering* 2000; **190**:305–319.
87. Beirao da Veiga L, Buffa A, Sangalli G, Vzquez R. Analysis suitable T-splines of arbitrary degree: Definition, linear independence, and approximation properties. *Mathematical Models and Methods in Applied Sciences* 2013; **23**(11):1979–2003.
88. Analysis-Suitable T-splines are Dual-Compatible. *Computer Methods in Applied Mechanics and Engineering* 2012; **249252**(0):42 – 51. Higher Order Finite Element and Isogeometric Methods.
89. Zienkiewicz OC, Taylor RL, Zienkiewicz OC, Taylor RL. *The finite element method*. McGraw-hill London, 1977.
90. Hughes TJR. *The Finite Element Method: Linear Static and Dynamic Finite Element Analysis* 2000; Cited By (since 1996)1023.
91. De Boor C. A practical guide to splines. *Mathematics of Computation* 1978; .
92. Marsden JE, Hughes TJR. *Mathematical foundations of elasticity*. Courier Dover Publications, 1994.
93. Hesch C, Gil A, Arranz Carreo A, Bonet J. On continuum immersed strategies for fluid-structure interaction. *Computer Methods in Applied Mechanics and Engineering* 2012; **247-248**:51–64.
94. Hesch C, Gil A, Carreño AA, Bonet J, Betsch P. A mortar approach for fluid–structure interaction problems: Immersed strategies for deformable and rigid bodies. *Computer Methods in Applied Mechanics and Engineering* 2014; **278**:853–882.
95. Simo JC, Hughes TJR. *Computational inelasticity*. Springer, 2008.
96. Balay S, Adams MF, Brown J, Brune P, Buschelman K, Eijkhout V, Gropp WD, Kaushik D, Knepley MG, McInnes LC, *et al.*. PETSc Web page. `http://www.mcs.anl.gov/petsc` 2014.
97. Boffi D, Cavallini N, Gastaldi L. The finite element immersed boundary method with distributed Lagrange multiplier. *arXiv preprint arXiv:1407.5184* 2014; .
98. Auricchio F, Boffi D, Gastaldi L, Lefieux A, Reali A. On a fictitious domain method with distributed Lagrange multiplier for interface problems. *Applied Numerical Mathematics* 2014; (0):–.
99. Auricchio F, Boffi D, Gastaldi L, Lefieux A, Reali A. A study on unfitted 1d finite element methods. *Computers & Mathematics with Applications* 2014; **68**(12):2080–2102.

DOI: 10.1002/nme