The tetrahedral finite cell method:
Higher-order immersogeometric analysis on adaptive
non-boundary-fitted meshes

Vasco Varduhn[1,*], Ming-Chen Hsu[2], Martin Ruess[3], Dominik Schillinger[1]

[1] *Department of Civil, Environmental, and Geo-Engineering, University of Minnesota, USA*
[2] *Department of Mechanical Engineering, Iowa State University, USA*
[3] *Aerospace Structures and Computational Mechanics, Delft University of Technology, The Netherlands*

SUMMARY

The finite cell method (FCM) is an immersed domain finite element method that combines higher-order non-boundary-fitted meshes, weak enforcement of Dirichlet boundary conditions, and adaptive quadrature based on recursive subdivision. Due to its ability to improve the geometric resolution of intersected elements, it can be characterized as an *immersogeometric* method. In this paper, we extend the FCM, so far only used with Cartesian hexahedral elements, to higher-order non-boundary-fitted tetrahedral meshes, based on a reformulation of the octree-based subdivision algorithm for tetrahedral elements. We show that the resulting TetFCM scheme is fully accurate in an immersogeometric sense, that is, the solution fields achieve optimal and exponential rates of convergence for h- and p-refinement, if the immersed geometry is resolved with sufficient accuracy. TetFCM can leverage the natural ability of tetrahedral elements for local mesh refinement in three dimensions. Its suitability for problems with sharp gradients and highly localized features is illustrated by the immersogeometric phase-field fracture analysis of a human femur bone. Copyright © 2000 John Wiley & Sons, Ltd.

KEY WORDS: Higher-order finite element methods; Immersogeometric analysis; Finite cell method; Adaptive tetrahedral meshes

---

*Correspondence to: Vasco Varduhn, Department of Civil, Environmental, and Geo-Engineering, University of Minnesota, 500 Pillsbury Drive S.E., Minneapolis, MN 55455, USA; Phone: +1 612 625 1807; Fax: +1 612 626 7750; E-mail: vvarduhn@umn.edu

Contents

1. INTRODUCTION

Immersed methods approximate the solution of boundary value problems using non-boundary-fitted discretizations. Their primary goal is to increase the geometric flexibility of discretization schemes with respect to their boundary-fitted counterparts and to alleviate meshing related obstacles that often appear for geometrically very complex domains. Instantiations of immersed methods have gained importance in many sub-disciplines, e.g., to resolve multi-phase flow interfaces in CFD [1, 2, 3], to deal with trimmed CAD surfaces in isogeometric analysis [4, 5, 6], to prevent mesh updating and mesh distortion effects in optimization [7, 8], or to handle fluid-structure interaction problems involving large displacements and contact [9, 10, 11, 12].

In the context of finite element analysis, immersed methods necessitate two additional critical capabilities that are not required in standard boundary-fitted FEA. First, they need a variationally consistent and accurate technique to impose boundary and interface conditions at surfaces that intersect elements. Over the last few years there has been significant progress in the weak enforcement of constraints (see e.g. [13, 14, 15, 16, 17, 18, 19]), with many applications outside the realm of immersed methods, e.g. for domain decomposition [20, 21] or boundary layer resolution [22, 23]. Second, immersed methods require an accurate quadrature technique to evaluate domain and surface integrals in intersected elements. Several studies have recently shown that inaccurate quadrature in intersected elements introduces a geometry error, which prevents higher-order accuracy [24, 25]. Influenced by isogeometric analysis [26, 27], where the importance of eliminating geometric errors has recently gained broader recognition, we follow Kamensky et al. [12] and denote methods that accurately represent the geometry of the immersed domain as *immersogeometric* methods. Immersogeometric analysis, combining the flexibility of variationally consistent weak boundary conditions with geometrically faithful quadrature of intersected elements, will guarantee higher-order accuracy.

An interesting precursor of the vision of high-fidelity immersogeometric analysis has been the finite cell method (FCM), introduced by Parvizian et al. [28] and Düster et al. [29]. At its present state of development, this technology combines the fictitious domain concept with higher-order basis functions for the approximation of solution fields, the weak imposition of unfitted Dirichlet boundary conditions, and the representation of the geometry by adaptive quadrature points [30]. The latter is based on the decomposition of each intersected element into sub-cells that can be efficiently organized in hierarchical tree data structures [31, 32]. Given sufficient resolution of the geometry, the FCM maintains optimal rates of convergence with mesh refinement and exponential rates of convergence with increasing polynomial degree [30]. The finite cell method can therefore be seen as an instantiation of an immersogeometric method. On the one hand, the FCM can operate with almost any geometric model, ranging from boundary representations in computer aided geometric design to voxel representations obtained from medical imaging technologies. On the other hand, the evaluation of the large number of quadrature points in intersected elements is computationally expensive.

Since its inception, the finite cell method has been further developed. Technical improvements include the weak imposition of boundary/coupling conditions [33, 34, 4], local refinement schemes [35, 36, 37, 38, 39], and improved quadrature rules for intersected elements [40, 24, 25]. In addition, the FCM has been successfully applied for large deformation analysis [41, 42, 30], thermoelasticity [43], homogenization [44], bone mechanics [45, 46], inelastic material behavior [47, 48], topology optimization [7], elastodynamics and wave propagation [49, 50, 51], and laminar and turbulent flows [52]. A concise summary of the FCM and

Copyright © 2000 John Wiley & Sons, Ltd.
Prepared using **nmeauth.cls**

Int. J. Numer. Meth. Engng 2000; **00**:1–6

applications can be found in the recent review article by Schillinger and Ruess [53]. In addition, there exists an open-source MATLAB code[†] that provides an instructive starting point for running numerical tests with the FCM [54].

In this paper, we extend the finite cell method, so far only used with Cartesian hexahedral elements, to higher-order non-boundary-fitted tetrahedral meshes. The tetrahedral finite cell method, or *TetFCM* in short, constitutes a change of paradigm with respect to Cartesian FCM, as it abandons the use of structured grids. We present an efficient workflow based on an octree based algorithm and the open-source mesh generator Netgen [55]. It is based on a cloud of h-values (i.e., the target edge length at each location) that is fed into Netgen as a basis for adaptive tetrahedral mesh generation. Since conformity to a simple embedding domain is the only geometric constraint, the discretization process is extremely fast, even for very large meshes. The mesher establishes the spatial adaptivity of the tetrahedral mesh by splitting elements until the closest h-value is reached. At the same time, it implicitly leverages advanced algorithms for mesh regularization and smoothing (available in any standard tet meshers) to ensure high-quality tetrahedral elements [56]. Tetrahedral meshes require an adaptation of the Cartesian decomposition scheme that generates adaptive quadrature points in intersected tetrahedral elements. We present a "bottom-up" approach based on adaptive tetrahedral sub-cells. Our algorithm first applies an inside/outside test to all quadrature points of the finest decomposition level. It then combines groups of fully-included tetrahedral sub-cells into larger tetrahedral sub-cells wherever possible, to reduce the final number of quadrature points. This involves a costly preprocessing step, but increases the geometric fidelity significantly with respect to "top-down" approaches [29, 53, 52].

Our article is organized as follows: In Section 2, we first provide a concise introduction to the finite cell method on Cartesian hexahedral elements. Section 3 briefly reviews fundamentals of the basis function technology on tetrahedra, with particular emphasis on higher-order warped integrated Legendre polynomials. Section 4 illustrates the automated generation of adaptive non-boundary-fitted tetrahedral meshes. We discuss the implementation of the technical components, in particular the reformulation of the octree based element decomposition, and their interaction with a standard mesh generator. Section 5 presents TetFCM results for two benchmarks and the femur bone. They demonstrate the performance of TetFCM in terms of accuracy vs. degrees of freedom, accuracy versus computing time and the efficiency of direct and iterative solvers. We also apply TetFCM based on adaptive immersogeometric tetrahedral meshes for the phase-field fracture analysis of a human femur bone. Section 6 summarizes the key aspects and draws conclusions.

2. FUNDAMENTALS OF THE FINITE CELL METHOD

We start with a concise summary of the technical components of the Cartesian finite cell method in the context of linear elasticity. We follow the presentation provided in the review paper by Schillinger and Ruess [53], which the interested reader is referred to for details.
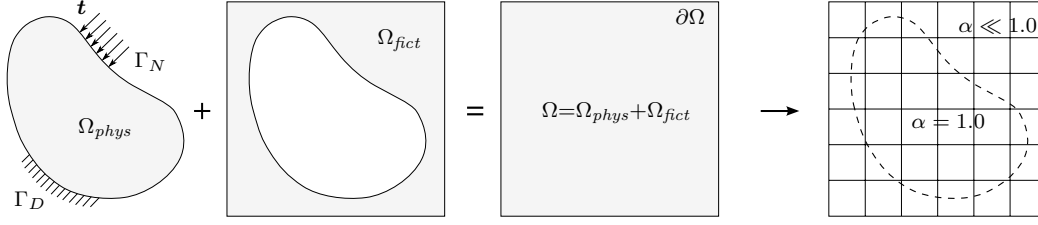

[†]http://fcmlab.cie.bgu.tum.de

Figure 1: *In the fictitious domain approach, the physical domain Ω_{phys} is extended by the fictitious domain Ω_{fict} into an embedding domain Ω that allows easy meshing. The original geometry is parameterized by a discontinuous indicator function α.*

2.1. The fictitious domain approach

Figure 1 illustrates the fictitious domain concept that lies at the heart of the finite cell method. The physical domain of interest Ω_{phys}, which can be geometrically complex, is extended by the fictitious domain Ω_{fict} to an embedding domain Ω, which is geometrically simple. Analogous to standard finite element methods, we consider a variational formulation, which is defined over the complete embedding domain Ω. For example, in linear elasticity, we use the principle of virtual work

$$\delta W\left(\boldsymbol{u}, \delta\boldsymbol{u}\right) = \int_\Omega \boldsymbol{\sigma} \,:\, \left(\nabla_{sym}\, \delta\boldsymbol{u}\right)\, dV - \int_\Omega \delta\boldsymbol{u}\cdot\boldsymbol{b}\, dV - \int_{\Gamma_N} \delta\boldsymbol{u}\cdot\boldsymbol{t}\, dA \;=\; 0 \qquad (1)$$

where $\boldsymbol{\sigma}$, $\boldsymbol{b}$, $\boldsymbol{u}$, $\delta\boldsymbol{u}$ and ∇_{sym} denote the Cauchy stress tensor, body forces, displacement vector, test function and the symmetric part of the gradient, respectively [57, 58, 59]. Neumann boundary conditions are specified over the boundary of the embedding domain $\partial\Omega$, where tractions are zero by definition, and over Γ_N of the physical domain, where tractions are given by vector $\boldsymbol{t}$ (see Fig. 1). The elasticity tensor $\boldsymbol{C}$ [57, 58, 59] relating stresses and strains

$$\boldsymbol{\sigma} = \alpha\boldsymbol{C} \,:\, \boldsymbol{\varepsilon} \qquad (2)$$

is complemented by a scalar discontinuous indicator function

$$\alpha\left(\boldsymbol{x}\right)\begin{cases} = 1.0 & \forall\boldsymbol{x}\in\Omega_{phys} \\ \ll 1.0 & \forall\boldsymbol{x}\in\Omega_{fict} \end{cases} \qquad (3)$$

which leaves the material parameters unchanged in the physical domain, but mitigates the contribution of the fictitious domain in (2). In Ω_{fict}, the value of the indicator function α should be chosen as small as possible, but large enough to prevent extreme ill-conditioning of the stiffness matrix [29, 28]. In our experience, α can be set to zero for moderately high polynomial degrees in the basis functions, e.g., quadratics or cubics. For high-order basis functions with $p > 4$ typical values of α range between 10^{-6} and 10^{-10}.

2.2. Higher-order non-boundary-fitted meshes

Using a non-boundary-fitted grid of higher-order elements (see Fig. 1) kinematic quantities are discretized as

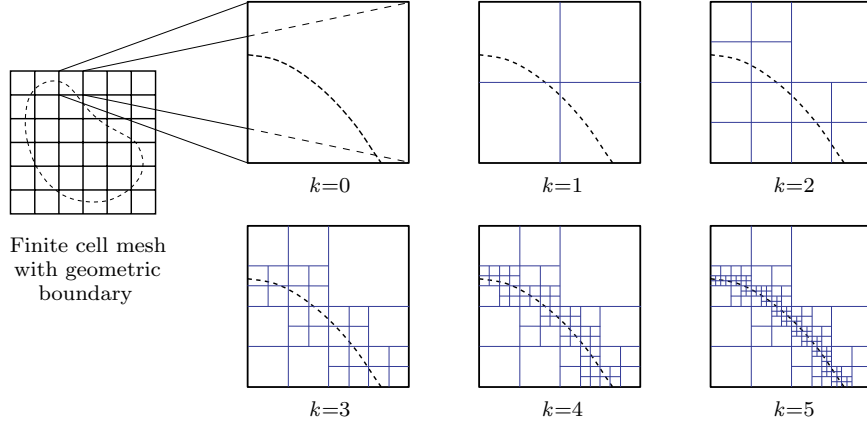$$\boldsymbol{u} = \sum_{a=1}^{n} N_a \boldsymbol{u}_a \qquad (4)$$

Figure 2: *2D sub-cell structure (thin blue lines) for adaptive integration of finite cells (bold black lines) intersected by a geometric boundary (dashed line).*

$$\delta\boldsymbol{u} = \sum_{a=1}^{n} N_a \delta\boldsymbol{u}_a \qquad (5)$$

The sum of N_a denotes a finite set of n higher-order basis functions, and $\boldsymbol{u}_a$ and $\delta\boldsymbol{u}_a$ are the corresponding vector-valued unknown coefficients [60, 58]. The discretized displacements (4) and virtual displacements (5) are defined over the complete embedding domain. It is important to identify basis functions with no support in the physical domain Ω_{phys} and to remove them from the discretization, since they do not contribute to the accuracy of the approximation in Ω_{phys}, but lead to rows and columns filled with only zeros in the case of $\alpha = 0$. Following the standard Bubnov-Galerkin approach [57, 58, 59], the substitution of (4) and (5) into the weak form (1) leads to a discrete set of equations.

2.3. Adaptive quadrature based on recursive subdivision

The accuracy of numerical integration by Gauss quadrature [58, 61] assumes smoothness of the integrands that appear in the variational formulation (1). Standard Gauss quadrature can therefore not be employed for integrating finite cells that are intersected by the geometric boundary, since the discontinuous indicator function α (3) introduces a discontinuity via (2). The Cartesian finite cell method based on quadrilateral and hexahedral meshes uses composed Gauss quadrature to improve the integration accuracy in intersected elements, based on a hierarchical decomposition of each intersected element into integration sub-cells [29, 40].

We illustrate the sub-cell concept in Fig. 2 for the 2D case, where it can be implemented in the sense of a quadtree [31, 32]. On each sub-cell level, only those sub-cells intersected by the geometric boundary are further subdivided. Subdivision is repeated until a predefined maximum depth is reached. The quadtree approach can be easily adjusted to binary trees or octrees in 1D and 3D, respectively [29, 31, 32]. Finite elements are plotted in black and integration sub-cells are plotted in blue lines throughout this work (see Fig. 2) for clear distinction. The adaptive sub-cell scheme is easy to implement, but leads to an increased number of quadrature points, since in each sub-cell full Gauss quadrature is employed.

Interpreting the integration of intersected elements from a geometric point of view, adaptive quadrature enables the accurate representation of the geometry of the immersed domain. One can show that quadrature accuracy is equivalent to geometric accuracy, directly affecting the quality of the solution fields, an observation that was made for the first time in [24]. The ability of composed Gauss quadrature to accurately represent the geometry by increasing the number of sub-cells qualifies the FCM as an *immersogeometric* method [12].

2.4. Weak imposition of unfitted boundary conditions

The fictitious domain concept inherently satisfies Neumann boundary conditions of zero traction, since stresses cannot be transferred beyond Ω_{phys} due to the mitigation of the material stiffness with the indicator function (3) [29, 28]. Non-zero Neumann boundary conditions can be simply imposed by integrating over Γ_N (see Fig. 1), irrespective of whether the geometric boundary coincides with a cell boundary or not. Dirichlet boundary conditions defined along boundaries that intersect elements require an imposition in a weak sense. We prefer Nitsche's method (see e.g. [45, 42, 34, 4, 33]), since it does not introduce additional unknowns, preserves a symmetric, positive definite stiffness matrix and satisfies variational consistency.

In this paper we focus on linear elasticity, where Nitsche's method extends the weak form (1) by additional terms as follows

$$
\begin{aligned}
\delta W_{\boldsymbol{K}}\left(\boldsymbol{u}, \delta \boldsymbol{u}\right) \;=\; & \int_{\Omega} \boldsymbol{\sigma} : \left(\nabla_{sym}\, \delta \boldsymbol{u}\right) dV \;+\; \beta \int_{\Gamma_D} \boldsymbol{u} \cdot \delta \boldsymbol{u}\; dA \\
& -\int_{\Gamma_D} \delta\left(\boldsymbol{\sigma} \cdot \boldsymbol{n}\right) \cdot \boldsymbol{u}\; dA \;-\; \int_{\Gamma_D} \left(\boldsymbol{\sigma} \cdot \boldsymbol{n}\right) \cdot \delta \boldsymbol{u}\; dA
\end{aligned}
\tag{6}
$$

$$
\begin{aligned}
\delta W_{\boldsymbol{f}}\left(\boldsymbol{u}, \delta \boldsymbol{u}\right) \;=\; & \int_{\Omega_{phys}} \delta \boldsymbol{u} \cdot \boldsymbol{b}\; dV \;+\; \int_{\Gamma_N} \delta \boldsymbol{u} \cdot \boldsymbol{t}\; dA \\
& +\beta \int_{\Gamma_D} \hat{\boldsymbol{u}} \cdot \delta \boldsymbol{u}\; dA \;-\; \int_{\Gamma_D} \delta\left(\boldsymbol{\sigma} \cdot \boldsymbol{n}\right) \cdot \hat{\boldsymbol{u}}\; dA
\end{aligned}
\tag{7}
$$

where $\delta W_{\boldsymbol{K}} {=} \delta W_{\boldsymbol{f}}$. Function $\hat{\boldsymbol{u}}$ denotes the prescribed displacements along the Dirichlet boundary Γ_D, scalar β is a stabilization parameter, which can be chosen empirically or according to a generalized Eigenvalue problem [16, 34], and $\boldsymbol{n}$ is the outward unit normal vector on Γ_D. Discretization and evaluation of (7) and (8) leads to the stiffness matrix $\boldsymbol{K}$ and the force vector $\boldsymbol{f}$, respectively.

3. FUNDAMENTALS OF TETRAHEDRAL BASIS FUNCTION TECHNOLOGY

In the following, we employ two sets of tetrahedral basis functions in a FCM context. Lagrange polynomials on barycentric coordinates constitute the standard way of defining linear, quadratic and cubic tetrahedral elements. For higher-order basis functions, we map tensor-product integrated Legendre polynomials defined on a hexahedral element to a tetrahedron.

3.1. Nodal basis functions in barycentric coordinates

Nodal tetrahedral elements are based on Lagrange polynomials that are interpolatory at element nodes. Corresponding element basis functions can be conveniently defined

Copyright © 2000 John Wiley & Sons, Ltd.

Prepared using nmeauth.cls

Int. J. Numer. Meth. Engng 2000; **00**:1–6

in barycentric coordinates, which form a set of four dimensionless numbers denoted by $\zeta_1, \zeta_2, \zeta_3, \zeta_4$. The value of ζ_i is one at vertex i and zero at the other three vertices, including the entire opposite face. It varies linearly with distance as one traverses the distance from the corner to that face. The barycentric coordinates satisfy the following constraint

$$\zeta_1 + \zeta_2 + \zeta_3 + \zeta_4 = 1 \tag{8}$$

so that they constitute three independent variables suitable for describing 3D space.

Basis functions are intrinsically linked to the tetrahedral element geometry and therefore best expressed in barycentric coordinates. For each linear tetrahedral element e, the four nodal basis functions simply read

$$N_1^e = \zeta_1; \quad N_2^e = \zeta_2; \quad N_3^e = \zeta_3; \quad N_4^e = \zeta_4; \tag{9}$$

where the subscript index corresponds to the nodes located at the four vertices (see Fig. 3). Higher-order basis functions in terms of 10-node quadratic and 20-node cubic Lagrange polynomials in barycentric coordinates can be found for example in [59, 62].

Solution fields such as displacements, strains or stresses are expressed in physical coordinates $\{x, y, z\}$. To establish equations that pass from one coordinate system to the other, we interpolate the tetrahedral element geometry by the linear basis functions (9) in barycentric coordinates as

$$\begin{bmatrix} 1 \\ x \\ y \\ z \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ x_1 & x_2 & x_3 & x_4 \\ y_1 & y_2 & y_3 & y_4 \\ z_1 & z_2 & z_3 & z_4 \end{bmatrix} \begin{bmatrix} \zeta_1 \\ \zeta_2 \\ \zeta_3 \\ \zeta_4 \end{bmatrix} \tag{10}$$

It is easy to compute the inverse relation of (10) analytically, from which partial derivatives of each physical coordinate with respect to each barycentric coordinate can be established (see for example [62]). These partial derivatives can be used to map derivatives with respect to barycentric coordinates to derivatives with respect to physical coordinates. We note that in the scope of non-boundary-fitted meshes, we will always use tetrahedral elements with straight edges and planar faces, so that the element geometry can be exactly expressed with (10) also for higher-order elements (subparametric mapping).

3.2. Modal high-order basis functions

There exists several ways for generating higher-order basis functions of arbitrary polynomial degree. In general, higher-order approaches can be classified in terms of non-tensor-product and warped tensor-product expansions [63]. In this work, we follow the latter strategy, using integrated Legendre polynomials [64, 65, 66].

3.2.1. The concept of warped tensor-product expansions Warped tensor-product expansions on tetrahedra are generated as follows: We first define suitable tensor-product basis functions on the parametric hexahedral domain. We then establish a transformation to the parametric tetrahedral domain [67, 68], using the concept of collapsed coordinates [69] (also sometimes referred to as Duffy transformation). Let us assume that $\{\eta_1, \eta_2, \eta_3\}$ are the Cartesian coordinates that define the hexahedral parametric domain $(-1; 1)^3$, and that $\{\xi_1, \xi_2, \xi_3\}$ are the Cartesian coordinates, in which the tetrahedral parametric domain has vertices $(-1, -1, -1)$,
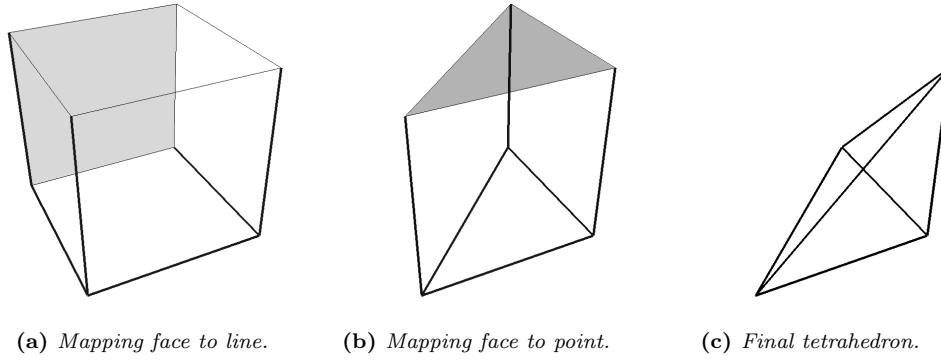
(a) *Mapping face to line.*   **(b)** *Mapping face to point.*   **(c)** *Final tetrahedron.*

Figure 3: *Using the concept of collapsed coordinates (grey faces), tetrahedral coordinates are derived from the hexahedral element.*

$(+1, -1, -1)$, $(-1, +1, -1)$, and $(-1, -1, +1)$. The transformation rule that maps tetrahedral coordinates to hexahedral coordinates is defined as

$$\eta_1 = -\frac{2(1 + \xi_1)}{\xi_2 + \xi_3} - 1; \qquad \eta_2 = \frac{2(1 + \xi_2)}{1 - \xi_3} - 1; \qquad \eta_3 = \xi_3 \tag{11}$$

and the corresponding Jacobian determinant is given as

$$j = \frac{4}{(\xi_2 + \xi_3)(\xi_3 - 1)} \tag{12}$$

The transformation is illustrated in Fig. 3. Substituting the transformation rule (11) into the basis functions given in terms of $\{\eta_1, \eta_2, \eta_3\}$ on the hexahedral parametric domain yields the warped basis functions in terms of $\{\xi_1, \xi_2, \xi_3\}$ on the tetrahedral parametric domain.

3.2.2. Basis functions based on integrated Legendre polynomials In this work, we adopt a set of basis functions originally introduced by Wassouf [66]. He used a tensor-product basis generated with integrated Legendre polynomials and transformed it with (11) to the tetrahedral domain. The resulting basis functions are hierarchic, in the sense that increasing the polynomial degree can be achieved by adding a number of basis functions to the original set rather than changing the complete set of functions. Hierarchic basis functions are also known as modal basis functions, since they are typically classified in terms of vertex, edge, face and internal modes [64, 66]. In the following, we briefly outline the generation of modal basis functions for each class of modes. We use hexahedral coordinates, since there exists no closed-form analytical representation in tetrahedral coordinates. The numbering of vertices, edges and faces corresponds to the convention shown in [62, 66].

Vertex modes represent nodal basis functions each of which is interpolatory at one of the four element vertices v_i. They are defined as

$$\Psi_{v1}^e = \frac{1}{8}(1 - \eta_1)(1 - \eta_2)(1 - \eta_3) \tag{13a}$$

$$\Psi_{v2}^e = \frac{1}{8}(1 + \eta_1)(1 - \eta_2)(1 - \eta_3) \tag{13b}$$

$$\Psi_{v3}^e = \frac{1}{4}(1 + \eta_2)(1 - \eta_3) \tag{13c}$$

$$\Psi_{v4}^e = \frac{1}{2}(1 + \eta_3) \tag{13d}$$

It is straightforward to check that inserting (11) into (13) yields the standard basis functions (9) in tetrahedral coordinates.

Edge modes represent basis functions that are zero at all vertices and at all but one element edges e_i. They are defined as

$$\Psi_{e1} = \hat{p}_j^0(\eta_1) \left(\frac{1 - \eta_2}{2}\right)^j \left(\frac{1 - \eta_3}{2}\right)^j \qquad j = 2, \ldots, p \tag{14a}$$

$$\Psi_{e2} = \hat{p}_j^0(\eta_2) \left(\frac{1 + \eta_1}{2}\right) \left(\frac{1 - \eta_3}{2}\right)^j \qquad j = 2, \ldots, p \tag{14b}$$

$$\Psi_{e3} = \hat{p}_j^0(\eta_2) \left(\frac{1 - \eta_1}{2}\right) \left(\frac{1 - \eta_3}{2}\right)^j \qquad j = 2, \ldots, p \tag{14c}$$

$$\Psi_{e4} = \hat{p}_j^0(\eta_3) \left(\frac{1 - \eta_1}{2}\right) \left(\frac{1 - \eta_2}{2}\right) \qquad j = 2, \ldots, p \tag{14d}$$

$$\Psi_{e5} = \hat{p}_j^0(\eta_3) \left(\frac{1 + \eta_1}{2}\right) \left(\frac{1 - \eta_2}{2}\right) \qquad j = 2, \ldots, p \tag{14e}$$

$$\Psi_{e6} = \hat{p}_j^0(\eta_3) \left(\frac{1 + \eta_2}{2}\right) \qquad j = 2, \ldots, p \tag{14f}$$

where $\hat{p}_j^0(\cdot)$ denotes the one-dimensional integrated Legendre polynomial in the corresponding direction. Index j denotes the order that is increased up to the desired overall polynomial degree p of the set of basis functions. We emphasize that the resulting basis functions on the tetrahedron are polynomials again. To this end, some of the tensor-product terms carry exponents to eliminate rational terms that appear due to (11).

Face modes represent basis functions that are zero at all vertices and edges and at all but one element faces f_i. They are defined as

$$\Psi_{f1} = \hat{p}_j^0(\eta_1)\, \hat{p}_k^0(\eta_2) \left(\frac{1 - \eta_2}{2}\right)^j \left(\frac{1 - \eta_3}{2}\right)^{j+k} \tag{15a}$$

$$\Psi_{f2} = \hat{p}_j^0(\eta_1)\, \hat{p}_k^0(\eta_3) \left(\frac{1 - \eta_2}{2}\right) \left(\frac{1 - \eta_3}{2}\right)^j \tag{15b}$$

$$\Psi_{f3} = \hat{p}_j^0(\eta_2)\, \hat{p}_k^0(\eta_3) \left(\frac{1 + \eta_1}{2}\right) \left(\frac{1 - \eta_3}{2}\right)^j \tag{15c}$$

$$\Psi_{f4} = \hat{p}_j^0(\eta_2)\, \hat{p}_k^0(\eta_3) \left(\frac{1 - \eta_1}{2}\right) \left(\frac{1 - \eta_3}{2}\right)^j \tag{15d}$$

$$j = 2, \ldots, p; \quad k = 2, \ldots, p - j$$

where the terminology corresponds to the previous paragraph. Some of the tensor-product terms and their exponents eliminate rational terms that appear due to (11).

Copyright © 2000 John Wiley & Sons, Ltd.
Prepared using **nmeauth.cls**

Int. J. Numer. Meth. Engng 2000; **00**:1–6

Internal modes represent basis functions that are zero at all vertices, edges and faces. They are defined as

$$\Psi_b \;=\; \hat{p}_j^0(\eta_1)\; \hat{p}_{k+1}^0(\eta_2)\; \hat{p}_{l+1}^0(\eta_3)\; \left(\frac{1-\eta_2}{2}\right)\; \left(\frac{1-\eta_3}{2}\right)^{j+k-1} \tag{16}$$

$$j = 2, \ldots, p-2; \quad k = 2, \ldots, p-j; \quad l = 2, \ldots, p-j-k$$

The extra terms in the tensor-product eliminate all rational terms that appear due to (11).

The set of all modal basis functions can exactly represent a complete polynomial of degree p over the tetrahedron, that is, they span a space S that consists of the following monomials

$$S: \quad \xi_1^j \, \xi_2^k \, \xi_3^l \qquad \text{for all } \{i, j, k\} \text{ s.t. } 0 \le j + k + l \le p \tag{17}$$

on each tetrahedron. It follows from (17) that the total number of basis functions in each element is $1/6\,(p+1)(p+2)(p+3)$, the maximum degree of each monomial is p, and hence the modal basis is linearly independent.

3.3. Symmetry, continuity, and hierarchy

Symmetry is satisfied when basis functions are rotationally symmetric with respect to element vertices, edges and faces. In case of symmetry, tetrahedral elements can be connected in an arbitrary fashion, and each element basis function that is non-zero over an element boundary will be in direct correspondence with other basis functions in the neighboring elements. Therefore, a symmetric basis directly satisfies the requirement of C^0 continuity over element boundaries. If basis functions are formulated in terms of barycentric coordinates, such as Lagrange polynomials of Section 3.1, they satisfy symmetry automatically due to the rotational symmetry of the barycentric coordinate system. The warped modal basis is not symmetric, since some of the vertices, edges and faces of the original hexahedral domain are collapsed into a single vertex or edge of the tetrahedral domain. To guarantee C^0 continuity automatically on arbitrary tetrahedral meshes, we organize the basis functions on each element according to the global numbering of the element vertices [70, 71]. This principle ensures that edge and face modes in neighboring elements coincide, since they are defined according to the same rising sequence of vertex points (two for each edge, three for each face). The hierarchy of the modal basis can be exploited to increase the computational efficiency, if multiple computations with different resolutions on the same mesh are required. Following the concept of p-adaptivity [72, 73], subsequent computations can use part of the stiffness matrix of previous computations, and only those entries need to be computed that involve the higher-order basis functions added in the current p-refinement step.

4. THE TETRAHEDRAL FINITE CELL METHOD

In the following, we present the tetrahedral finite cell method as an adaptation of the Cartesian FCM reviewed in Section 2. Particular emphasis is placed on the generation of adaptive unfitted tetrahedral meshes and adaptive quadrature points in intersected elements. The latter includes the presentation of algorithms for a "bottom-up" approach that guarantees accurate geometry resolution irrespective of the form of the intersection.
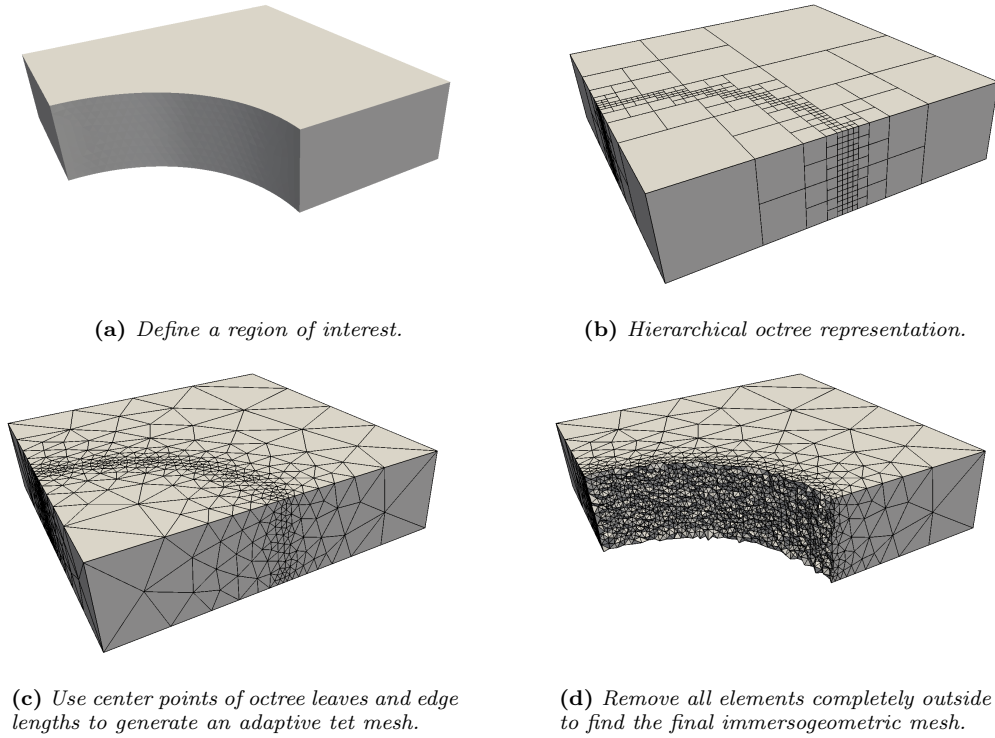
(a) *Define a region of interest.*

(b) *Hierarchical octree representation.*

(c) *Use center points of octree leaves and edge lengths to generate an adaptive tet mesh.*

(d) *Remove all elements completely outside to find the final immersogeometric mesh.*

Figure 4: *Generation of a adaptive immersogeometric meshes.*

4.1. Generating adaptive tetrahedral meshes

Tetrahedral elements have the capability of local refinement in 3D without introducing hanging nodes. This makes them especially suitable for applications where adaptivity is important to control local accuracy without introducing a prohibitively large number of degrees of freedom through global refinement. In the following, we develop a pipeline for the generation of adaptive tetrahedral meshes that allows a flexible specification of regions with local refinement. In a first step, we use a hierarchical data structure based on the octree concept [31, 32] to generate a cloud of adaptive spatial points. Each point specifies the local element width h in its vicinity, providing a measure of local mesh density. In a second step, we feed the cloud of points into Netgen [55] (or any other standard tetrahedral meshing tool), where it serves as the basis for the generation of an adaptive non-boundary-fitted tetrahedral mesh. Many advanced tetrahedral mesh generators make use of built-in efficient mesh smoothing and regularization algorithms [56] that ensure high-quality undistorted elements with well-behaved angles.

Figure 4 illustrates this process for a quarter of a thick plate with a circular hole. We assume that we want to resolve the boundary region around the circular hole adaptively for higher accuracy. We first refine the region up to a predefined depth, using a hierarchical octree. Its hierarchical representation directly provides the point cloud for the generation of the adaptive target mesh. The leaf nodes of the octree define a non-overlapping complete partitioning of the domain with the finest resolution in the boundary region. The center of each leaf defines one
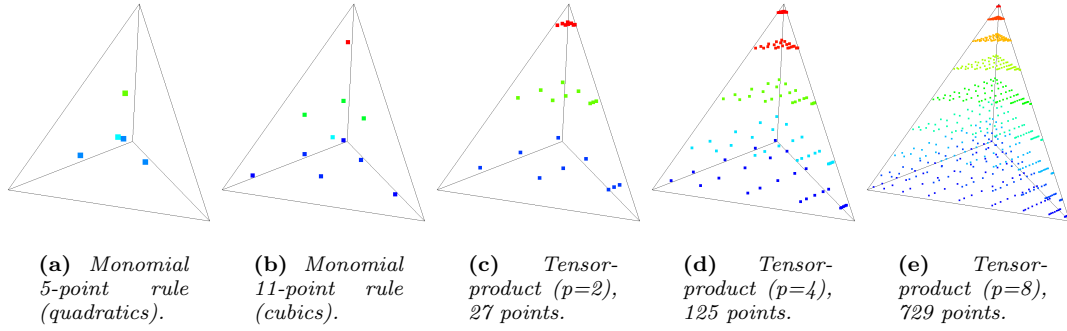
Copyright © 2000 John Wiley & Sons, Ltd.
Prepared using **nmeauth.cls**

Int. J. Numer. Meth. Engng 2000; **00**:1–6

(a) *Monomial 5-point rule (quadratics).* **(b)** *Monomial 11-point rule (cubics).* **(c)** *Tensor-product (p=2), 27 points.* **(d)** *Tensor-product (p=4), 125 points.* **(e)** *Tensor-product (p=8), 729 points.*

Figure 5: *Quadrature points for monomial and tensor-product rules used for nodal and modal basis functions of different p, respectively. The color encodes their spatial coordinate in vertical direction.*

point in the cloud and the corresponding local element length can be simply computed from the diameter of the leaf. The resulting cloud of points and local edge lengths completely specify the target mesh that is generated in Netgen. Removing all elements with no contribution in the physical domain yields the final immersogeometric mesh that is independent of the geometric boundaries of the immersed object and its local features.

4.2. Quadrature rules on tetrahedral elements

For numerical integration over tetrahedra we use two different approaches depending on the type of the basis functions.

4.2.1. Quadrature rules for nodal elements

Following [74] we employ a five-point quadrature rule for the integration over quadratic elements and an eleven-point quadrature rule for integration over cubic elements. The corresponding quadrature points are illustrated in Figs. 5a and 5b, respectively. We note that we can use any other monomial rule that yields the desired accuracy (see for example [75, 76, 62, 77]).

4.2.2. Quadrature rules for modal elements

Following Wassouf [66] and Hillion [78, 79] we can derive a quadrature rule for polynomials of arbitrary degree defined over tetrahedra. Using affine transformations, the standard 3D integration domain can be transformed to a tetrahedron. The corresponding integral expression to be evaluated reads

$$\int_{-1}^{1} \int_{-1}^{-\xi_1} \int_{-1}^{-\xi_1-\xi_2} f(\xi_1,\xi_2,\xi_3) d\xi_3 d\xi_2 d\xi_1 = \sum_{i=1}^{n} \sum_{j=1}^{n} \sum_{k=1}^{n} w_i^* w_j^* w_k^* f(\eta_1^{*i}, \eta_2^{*j}, \eta_3^{*k}) \qquad (18)$$

for which we can use three-dimensional quadrature points

$$\eta_1^{*i} = \eta_1^i \tag{19a}$$

$$w_1^{*i} = \frac{1}{2} w_1^i (1 - \eta_1^{*i}) \tag{19b}$$

$$\eta_2^{*j} = \frac{1}{2}(-\eta_1^i \eta_2^j + \eta_2^j - \eta_1^i - 1) \tag{19c}$$

(a) *four corner cells*　　　　　　　　**(b)** *an octahedron is split into four cells*
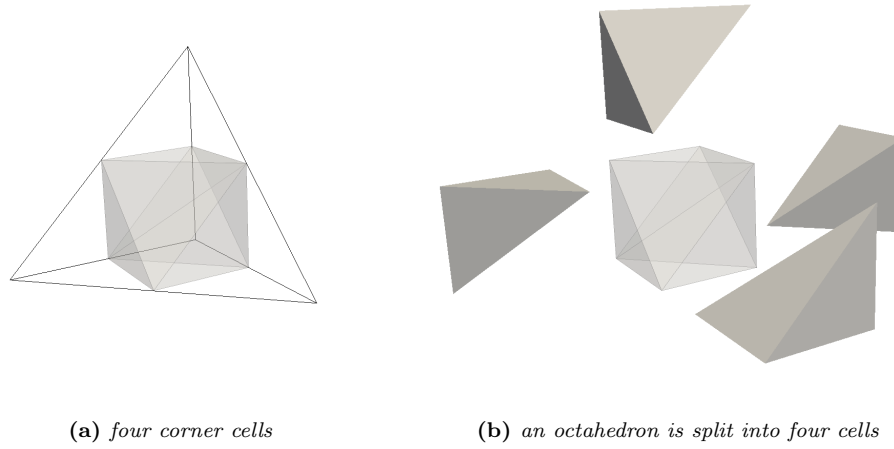
Figure 6: *One element is split into 8 subcells which are formed by the four corner cells and a uniform split of the remaining octahedron into four cells.*

$$w_2^{*j} = -\frac{1}{2} w_2^j (\eta_1^{*i} + \eta_2^{*j}) \tag{19d}$$

$$\eta_3^{*k} = \frac{1}{4}(-\eta_1^i \eta_3^k + \eta_1^i \eta_2^j \eta_3^k - \eta_2^j \eta_3^k + \eta_3^k + \eta_1^i \eta_2^j - \eta_2^j - 3) \tag{19e}$$

$$w_3^{*k} = w_3^k \tag{19f}$$

where $\{\eta^{*i}, w_i^*\}$ denotes the i-th quadrature point with the corresponding weight. The resulting quadrature rules with $p+1$ points in each parametric direction of the untransformed hexahedron are illustrated for different polynomial degrees in Figs. 5c to 5e.

4.3. Adaptive quadrature of intersected tetrahedra based on octree subdivision

The accuracy of FCM solution fields depends on how accurately the geometry inside each intersected element is represented by the quadrature rule. We introduce an adaptive quadrature method based on octree subdivision for tetrahedral elements that adapts the recursive subdivision concept applied for hexahedral cells (see Section 2.3) to the tetrahedral case. The general idea is based on splitting intersected elements into tetrahedral sub-cells as shown in Fig. 6. Each cell is decomposed into eight sub-cells which consist of four corner sub-cells and a uniform split of the remaining octahedron into four cells. Following the octree approach for hexahedrals used in Cartesian FCM [29, 53], this procedure is repeated recursively for each sub-cell intersected by the geometric boundary until a predefined maximum level of sub-cells is reached. We emphasize again that splitting is performed on the integration level only and does not affect the basis functions, which are still defined on the original tetrahedral element. For each of the sub-cells, the same integration rule is applied. This keeps the amount of quadrature points per sub-cell constant and allows an easy calculation of the weights and local coordinates of the recursive quadrature points. The weights of the quadrature points in each sub-cell are scaled with the volume of the sub-cell.

From an algorithmic viewpoint, this idea is implemented in a "bottom-up" fashion. Instead of building up the octree in the regular "top-down" approach, we first refine the complete tetrahedron by generating the quadrature points of all possible leaves at the maximum tree depth. We then check each sub-cell whether it is intersected by the geometric boundary. An important consideration is how to determine best whether a sub-cell is intersected. Instead of checking whether a face or edge is intersected or vertices are located on different sides of the geometric boundary, our intersection test solely relies on an inside/outside test for each quadrature point. If we detect that quadrature points of one sub-cell are located on different sides of the geometric boundary, we mark it as intersected. Based on this result, we start building up the octree from the bottom up by combining sets of non-intersected leaves into one leaf of higher level. This pruning procedure is repeated recursively until we reach the root at the top, that is, the original finite element. The major advantage of the bottom-up approach over the top-down approach is a significantly increased geometric accuracy, since elements that are intersected in such a way that only a small portion of their domain is cut are captured reliably. The bottom-up procedure is computationally more expensive than the top-down procedure, but is eminently suited for parallelization. A detailed algorithmic description is provided in Algorithm 1.

We illustrate the algorithm for the embedded cube in Fig. 7, which shows sub-cells for different maximum octree depths. Our approach exhibits the following two advantages. First, we do not subdivide elements with very small cuts. In the top-down approach, they would be subdivided, but without effect, since the corresponding quadrature points are all either completely inside or outside of the physical domain. The bottom-up approach recombines all sub-cells to the original element, thus reducing the number of quadrature points and saving computation time. Second, we automatically exclude all elements that have only very small portions of their element domain in the physical domain (created, e.g., by chopping off most of the domain such that only one vertex of the tetrahedron is located within the physical domain). Such elements have no contribution to the system matrix, since no quadrature point is located in the physical domain. If kept in the mesh, they either lead to a singular system matrix or negatively influence the condition of the system matrix, when a small value $\alpha \ll 1$ is applied, see indicator function (3).

4.4. Voxel quadrature

Image based geometric models that emanate from medical imaging technologies such as quantitative computed tomography (qCT) scans are the most prominent data source for patient-specific simulations in biomedical applications. They are made up of a rasterized voxel structure, where each voxel contains a color value that can be associated with a physical property, e.g., material density. If the tetrahedral finite cell method is applied for the analysis of image based geometric models, the concept of intersected elements and the recursive resolution of the geometry by adaptive quadrature does not apply, as there exists no clearly defined boundary of the physical domain. Instead, we suggest a quadrature approach that follows two principles. First, tetrahedral elements that are completely located outside the physical domain, that is, the color value of all voxel located within this element are below or above the predefined threshold, are removed from the mesh. Second, we subdivide each tetrahedral element into sub-cells. The level of sub-cells is the same for each cell and throughout the complete mesh. The sub-cell resolution is chosen such that the density of the resulting quadrature points

Data: element e, maximal integration tree depth d;

function generateAdaptiveQuadrature(e, d)

   % *resolve subcell tree of element completely*
   generateChildren(e.cell, d);

   % *prune tree from cells which have quadrature points only inside or only outside the physical domain*
   pruneTree(e.cell, d);

end

function generateChildren(c, d)

   % *generate quadrature points of cell*
   c.qp = qudaraturePoints(c.coordinates);

   % *resolve subcell tree completely*
   if *d != 0* **then**

      % *split cell into eight cells*
      c.children = splitTo8(c);

      for *i_ cell = 1:8* **do**

         % *recursively process subcells*
   generateChildren(c.children[i_cell], d-1);
      end

   end

end

function pruneTree(c, d)

   % *process tree bottom-up*
   if *d != 0* **then**

      for *i_ cell = 1:8* **do**

         % *recursively process subcells*
         pruneTree(c.children[i_cell], d-1);
      end

   end

   % *count subcells with all quadrature points inside or all outside the physical domain*
   int inside = 0, outside = 0;
   for *i_ cell = 1:8* **do**

      % *all quadrature points inside*
      if *allQuadraturepointsInside(c.children[i_ cell])* **then**
         ++inside;
      end
      % *all quadrature points outside*
      if *allQuadraturepointsOutside(c.children[i_ cell])* **then**
         ++outside;
      end

   end

   if *inside == 8 OR outside == 8* **then**
      % *element is not intersected, clear subcells*
      c.children.clear();
   else
      % *element is intersected, clear quadrature points*
      c.qp.clear();
   end

end

Algorithm 1: Compute integration tree for subcells of an element e up to depth d.
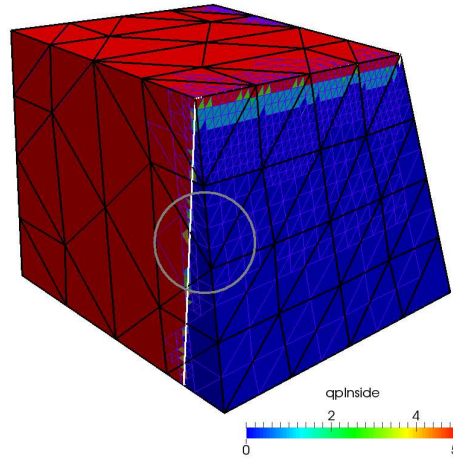
(a) *Non-boundary-fitted mesh of a cube (elements in black, sub-tetrahedra in blue, boundary in white).*

(b) *level $d = 0$*   **(c)** *level $d = 1$*   **(d)** *level $d = 2$*   **(e)** *level $d = 3$*   **(f)** *level $d = 4$*
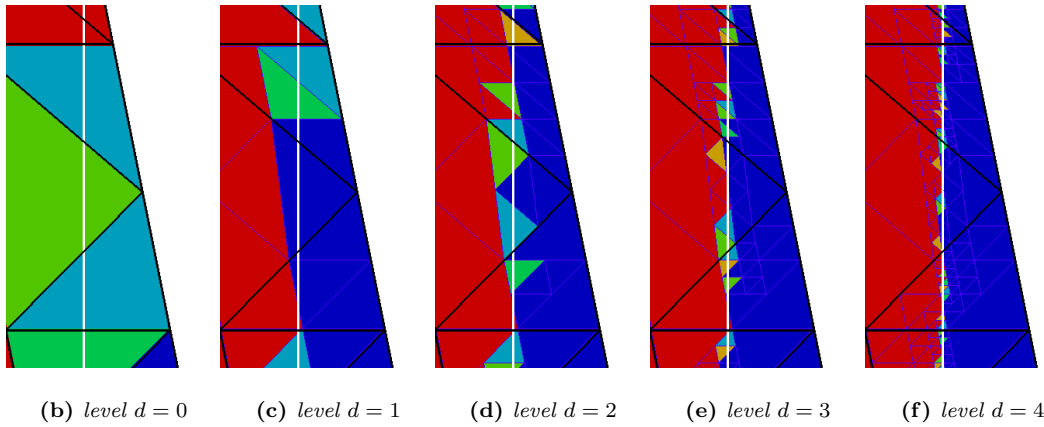
Figure 7: *By building the tree from the bottom up, we ensure that any small cut that can be resolved by the finest level d of sub-cells is captured. The color indicates for each sub-cell how many quadrature points are located inside the cube domain.*

approximately corresponds to the voxel density. As a consequence, each quadrature point can be approximately associated with one voxel. We emphasize that a finer resolution of quadrature points should be avoided, as it could resolve sharp interfaces between single voxels, which are an artifact of the geometric model.

5. NUMERICAL EXAMPLES

In this section we examine the accuracy and computational efficiency of the tetrahedral finite cell method for several benchmark problems. In particular, we illustrate the ability of high-

order modal basis functions to achieve exponential convergence rates and the advantages of quadratic and cubic nodal basis functions in terms of reasonable conditioning and fast iterative solution of large systems. We also highlight the ability of immersogeometric tetrahedral meshes to locally refine the solution fields in three dimensions.

5.1. Thick plate with a circular hole

As a first benchmark, we consider a thick plate with a circular hole under uniform tension shown in Fig. 8. We make use of the symmetry of the problem, reducing the system to one octant of the original domain. Symmetry boundary conditions are applied in a weak sense by using Nitsche's method. Figure 9 illustrates the basic steps of the immersogeometric discretization procedure using the TetFCM scheme. First, we generate an unfitted tetrahedral mesh of the embedding domain, using the mesh generator Netgen. Second, we employ octree subdivision described in Section 4.3 to generate integration sub-cells for performing adaptive quadrature in intersected elements.

Figure 10a plots the von Mises stress distribution in the boundary region close to the circular boundary. We observe that the stress field is smooth, the concentration at the lower boundary can be captured accurately and no unphysical interference of the boundary can be detected. We also compute the relative error in strain energy norm defined as [58, 64, 59]

$$e_r = \sqrt{\frac{|U_{num} - U_{ref}|}{U_{ref}}} \tag{20}$$

where U_{num} represents the numerical strain energy obtained for a specific discretization, and U_{ref} is a reference strain energy computed with an overkill discretization. Figure 10b plots the energy error versus the total number of degrees of freedom. We employ a series of uniformly refined Cartesian meshes with quadratic and cubic Lagrange basis functions and a coarse mesh based on integrated Legendre basis functions, where we increase the polynomial degree p at fixed element size. The geometry in intersected elements is resolved by adaptive quadrature with six levels of hierarchical sub-cells. We observe that for h-refinement with quadratic and cubic basis functions, we achieve optimal rates of convergence, which correspond to the polynomial degree p. For p-refinement, the TetFCM achieves exponential rates.

The stress accuracy that can be achieved directly on the immersed boundary in intersected elements is of particular interest in many situations, e.g., for stress analysis, where maximum stresses mostly occur on the surface, or in coupled multi-physics problems, where surface quantities need to be exchanged between different solvers. Figures 11a to 11c plot the
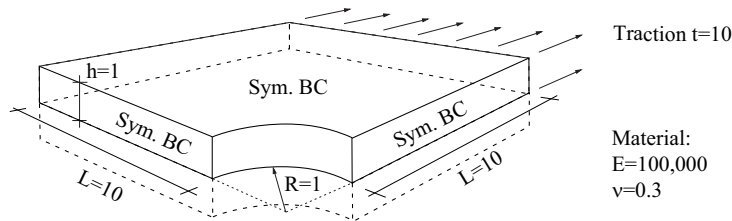

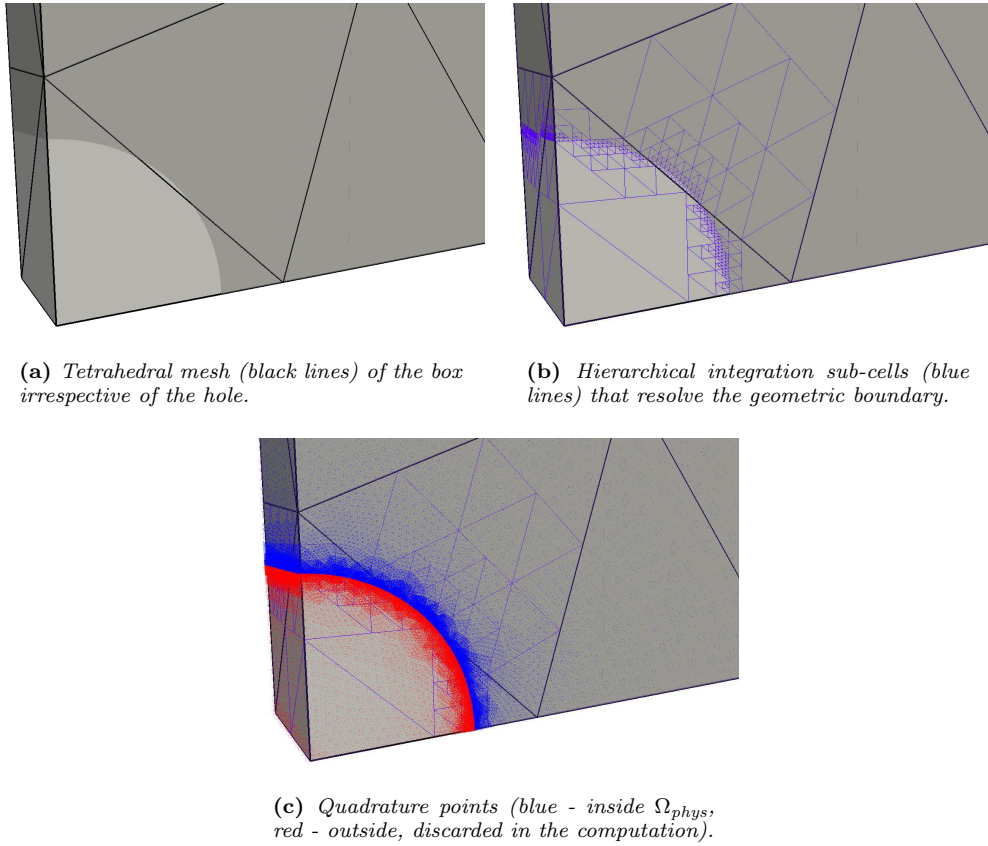**Figure 8:** *Three-dimensional thick plate with a circular hole.*

(a) *Tetrahedral mesh (black lines) of the box irrespective of the hole.*

(b) *Hierarchical integration sub-cells (blue lines) that resolve the geometric boundary.*

(c) *Quadrature points (blue - inside Ω_{phys}, red - outside, discarded in the computation).*

Figure 9: *TetFCM discretization procedure for the thick plate benchmark.*

circumferential stress on the circular boundary of the hole obtained with the same mesh of approximately 100 cubic tetrahedral elements, but different maximum levels of sub-cells. We observe that the accuracy of the stress solution critically depends on the accuracy of the adaptive quadrature rule. If we resolve intersected elements poorly, the stress solution largely deviates from the reference. If we accurately resolve the geometry by choosing a sufficiently large sub-cell level, the stresses are in excellent agreement. Figure 11 clearly demonstrates the importance of a faithful representation of the geometry in an immersogeometric sense to obtain accurate solution fields.

5.2. Voxelized cube with inhomogeneous stiffness

Image-based geometric models generated with medical imaging technologies often represent the spatial distribution of highly inhomogeneous material properties. Before looking at a voxel model of a real bone, we demonstrate the accuracy of the tetrahedral finite cell method for the analysis of voxel models with a strong variation of Young's modulus. To this end, we consider the unit cube whose domain is given by $\Omega = \big\{(x,y,z)|-0.5 \leq x, y \leq 0.5, 0.0 \leq z \leq 1.0\big\}$. The
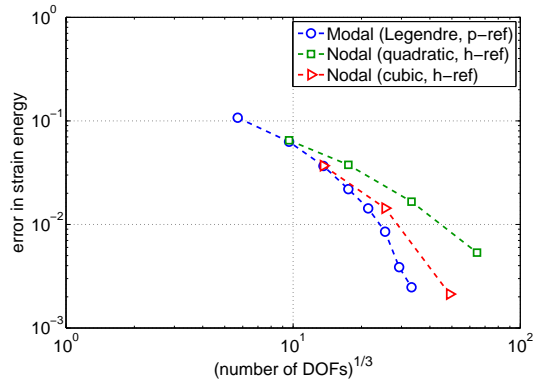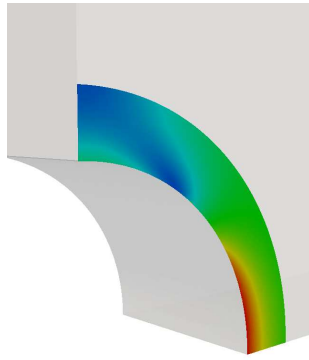
(a) *Von Mises stress near the hole.*

(b) *Convergence of the relative error in strain energy.*

Figure 10: *Accuracy of the TetFCM for the thick plate benchmark.*
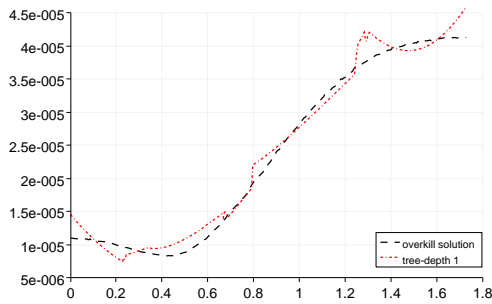
(a) *One level of sub-cells.*
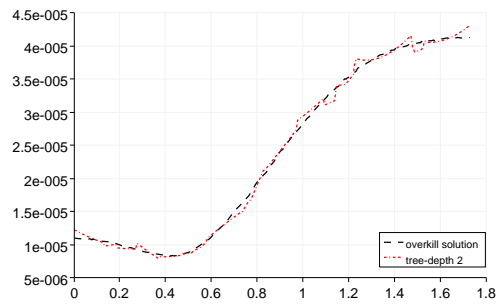
(b) *Two level of sub-cells.*
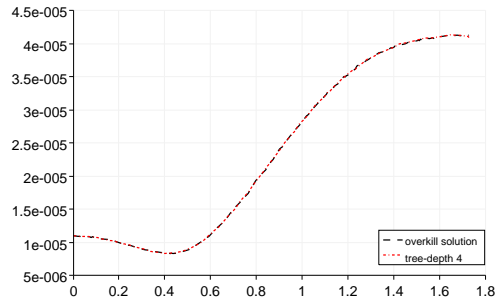
(c) *Four level of sub-cells.*

Figure 11: *Influence of the geometry resolution on the accuracy of the circumferential stress, plotted along the circular boundary ($\theta = 0...\pi/2$). Note that we use the same mesh and cubic basis functions.*

inhomogeneous Young's modulus is given by the function

$$E(x, y, z) = 3^x(10 + \sin(5y))(50 + \cos(10z)) \tag{21}$$

and Poisson's ratio is $\nu = 0.3$. We fix displacements normal to the faces $x = -0.5$, $y = -0.5$ and $z = 0.0$, and apply a unit traction normal to the face $y = 0.5$. Starting from the smooth function (21) plotted in Fig. 12a, we transfer the distribution of Young's modulus into a discrete voxel representation by partitioning the cube into voxel grids of different size and sampling (21) at each voxel center. Two examples are shown in Figs. 12b and 12c.

In the next step, we embed the cube into a larger domain that we discretize with tetrahedral elements using Netgen. Figures 13a and 13b show the embedding domain and the corresponding initial mesh that we use for uniform h-refinement with quadratic and cubic Lagrange basis functions and for p-refinement with warped integrated Legendre polynomials, respectively. Both embedding domains are designed in such a way that intersected tetrahedral elements, in particular with unfavorable cuts, are present in all meshes. Figures 14a and 14b plot the corresponding quadrature points. We observe that the geometry of intersected elements is resolved by the aggregation of quadrature points. When performing h-refinement with nodal elements, we only consider the points inside the cube for the formation of the stiffness matrix. When performing p-refinement, we consider all quadrature points, where contributions from points outside of the cube are penalized by $\alpha < 10^{-10}$ in the sense of (2) and (3). For large p, this is required to prevent extreme ill-conditioning of the stiffness matrix.

Figure 15a illustrates the convergence in strain energy, if we use the smooth description of Young's modulus (21). The curves confirm optimal rates of convergence for uniform mesh refinement of quadratic and cubic meshes and exponential rates of convergence for p-refinement on meshes with modal basis functions. In particular, the convergence behavior is unaffected by the presence of the fictitious domain extensions. Figure 15b illustrates the convergence in strain energy for the same discretizations, but based on discrete descriptions of Young's modulus with two differently sized rasterized voxel grids. We observe that if the error due to the approximation of the solution fields falls below the error due to the discrete description of Young's modulus, the convergence curve flattens. We conclude that the minimum error level that can be achieved depends on the granularity of the voxel resolution. We note that
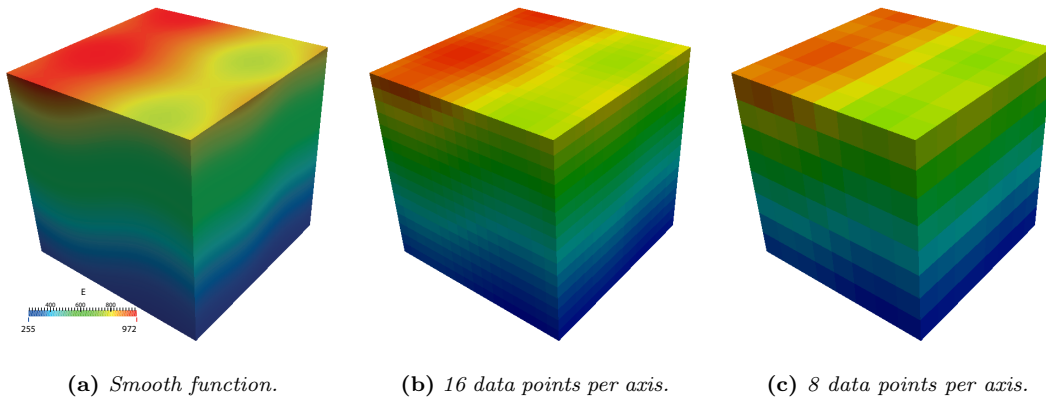

(a) Smooth function.   (b) 16 data points per axis.   (c) 8 data points per axis.

Figure 12: Cube with varying Young's modulus E - continuous vs. discrete voxel representations.

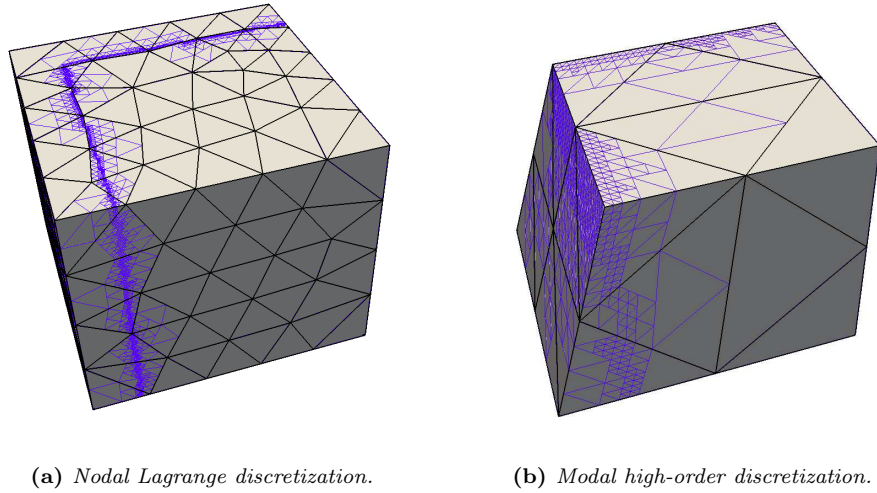**(a)** *Nodal Lagrange discretization.* **(b)** *Modal high-order discretization.*

Figure 13: *Tetrahedral mesh of the embedding domain (black lines) and resolution of the cube geometry with sub-cells (blue lines).*

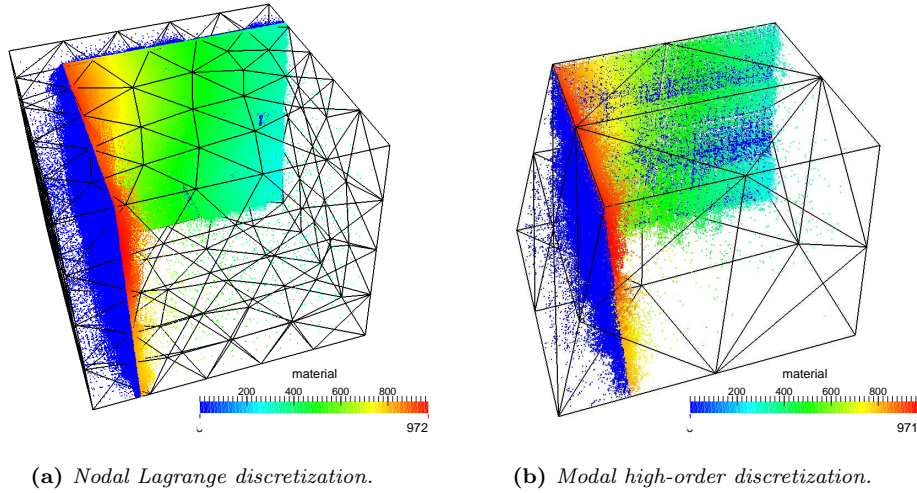**(a)** *Nodal Lagrange discretization.* **(b)** *Modal high-order discretization.*

Figure 14: *Corresponding quadrature points. Blue points are located outside the cube.*

in practical applications, we cannot control the granularity, as it is given by the image-based geometric model, which depends on the resolution of the available imaging technology. We can only ensure that our methods deliver the best possible result for the given voxel resolution.

Figure 16 illustrates the evolution of the condition number under h- and p-refinement for the cube example. Considering standard boundary-fitted meshes in Fig. 16a first, we observe that the condition number increases linearly when the mesh is refined, but exponentially when the polynomial degree p is increased. In the TetFCM scheme, the presence of intersected elements additionally affects the conditioning of the system matrix. However, Figure 16b indicates
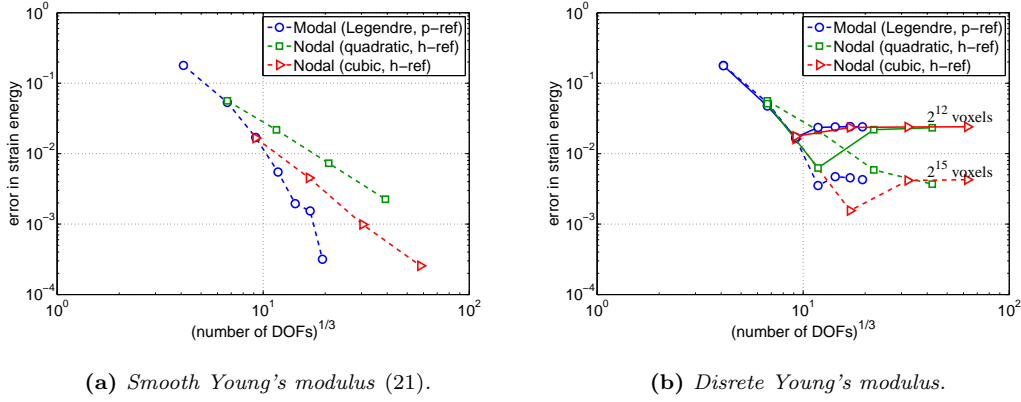
(a) *Smooth Young's modulus* (21).   **(b)** *Disrete Young's modulus.*

Figure 15: *Convergence of the relative error in strain energy.*

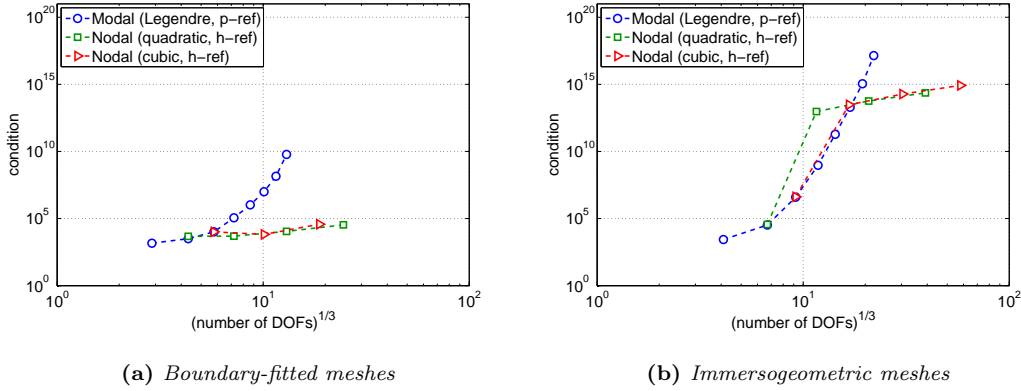**(a)** *Boundary-fitted meshes*   **(b)** *Immersogeometric meshes*

Figure 16: *Evolution of the condition number with increasing mesh size (nodal) or increasing polynomial degree (modal).*

that this effect is bounded, so that the condition number still increases linearly with mesh refinement, just at a higher level. Due to the exponential increase under p-refinement, the condition number increases exponentially when p is large, irrespective of whether the mesh is body-fitted or immersogeometric.

The conditioning of the system drastically affects the efficiency of iterative solvers. The performance of iterative and direct solvers for the cube example is illustrated in Fig. 17. We report computing times that were measured from the same C++ code based on the library framework Trilinos [80] that was run in serial using a single thread on a Intel(R) Xeon(R) E7-4830 @ 2.13GHz with 512 GB of RAM. The direct solver is Intel's version of *Pardiso* provided as part of their math kernel library (MKL), and the iterative solver is a preconditioned conjugate gradient (PCG) method provided as part of the Trilinos package AztecOO. We observe that if we use the direct solver, increasing the polynomial degree of high-order modal basis functions is attractive, as this strategy achieves a specified level of accuracy faster than refining the
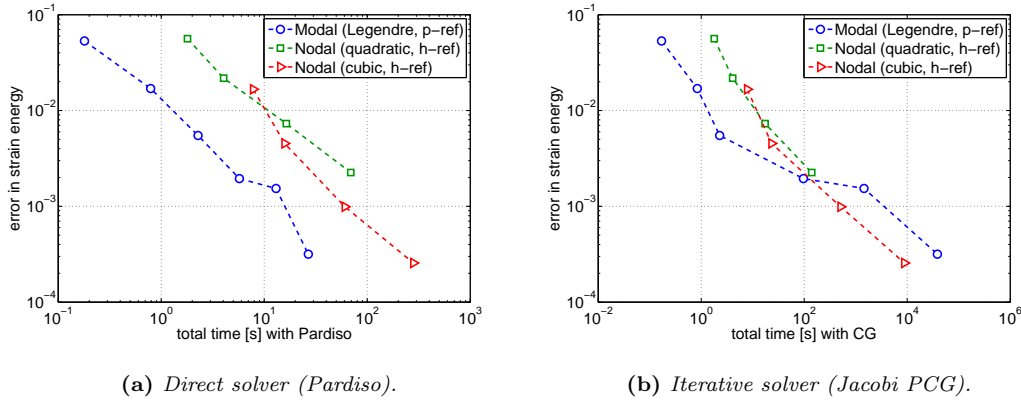
(a) *Direct solver (Pardiso).*   **(b)** *Iterative solver (Jacobi PCG).*

Figure 17: *Total computing times, including formation and solution with different solver types.*

mesh. However, if we use the iterative solver, the computing times for high-order modal basis functions grow dramatically, so that refining the mesh with quadratic and cubic basis functions is faster. This is a direct consequence of the severe ill-conditioning of the system for large p, which leads to a prohibitive number of iterations to achieve convergence in the iterative solution process. These results indicate that for large-scale computations, where small shared memory and the need for parallelization preclude the application of direct solvers, the best option for efficient TetFCM analysis are basis functions of moderately high degree that can be successfully applied with standard iterative solution technology.

We observed that for both quadratics and cubics a simple and inexpensive Jacobi preconditioner based on the inverse of the diagonal of the stiffness matrix works very well. In particular the Jacobi preconditioner worked better than more expensive approaches such as the incomplete LU factorization (ILU). The PCG method uses the fact that for a symmetric preconditioner $C = K K^T$ the linear equation system $A x = b$ can be transformed to $K^T A K y = K^T b$ with $y = K^{-1} x$. It conserves the definiteness of the matrix and the conditioning of the matrix is improved [81, 82]. As the Jacobi preconditioner uses the main diagonal of the global system matrix for constructing C it is obviously very fast to compute. The system matrix is the assembly of the element stiffness matrices. They are all similar to the reference element, but vary depending on the inhomogeneous material and the element volume. Our assumption is that the scaling of the Jacobi preconditioner with the diagonal entries corresponds to the scaled composed structure of the system matrix which leads to the good results for quadratics and cubics. For computations with $p \geq 4$ neither the Jacobi nor the ILU preconditioner gave satisfactory results.

5.3. Phase-field fracture analysis of a femur bone

Patient-specific stress and fracture analysis is based on quantitative computed tomography (qCT) scans of an individual bone that provide a 3D image of its mineral density, from which bone strength at each image point can be inferred [83, 84]. Standard simulation tools require labor-intensive image segmentation to separate the dense cortical shell at the bone's surface

from the foam-like trabecular bone in the interior. The finite cell method provides a framework that seamlessly integrates qCT data into automatic stress and fracture analysis [53, 45].

5.3.1. Phase-field model for brittle fracture   Our test problem is based on a phase-field model for brittle fracture [85, 86, 87, 88, 89], which is represented in variational form for quasistatic conditions by the following coupled equations

$$\int \left(\frac{4l_0 \psi_0^+}{\mathcal{G}_c} + 1 \right) c\, q\, d\Omega \; + \int 4l_0^2 \, \nabla c \, \nabla q \, d\Omega = \int q \, d\Omega \tag{22}$$

$$\int \left(\boldsymbol{\sigma}^+ + \boldsymbol{\sigma}^- \right) : \nabla \dot{\boldsymbol{w}} \, d\Omega = \int \boldsymbol{b} \cdot \boldsymbol{w} \, d\Omega + \int \boldsymbol{t} \cdot \boldsymbol{w} \, d\partial\Omega \tag{23}$$

The pairs $\{\boldsymbol{u}, \boldsymbol{w}\}$ and $\{c, q\}$ represent the displacement and phase-field solutions and corresponding test functions, $\mathcal{G}_c$ and l_0 are the energy release rate and a length scale, and λ and μ are the Lamé parameters. The tensile and compressive parts of the stress tensor read

$$\boldsymbol{\sigma}^+ := c^2 \left(\lambda \left\langle \text{tr}(\boldsymbol{\varepsilon}) \right\rangle^+ \boldsymbol{I} + 2\mu \, \boldsymbol{\varepsilon}^+ \right) \tag{24}$$

$$\boldsymbol{\sigma}^- := \lambda \left\langle \text{tr}(\boldsymbol{\varepsilon}) \right\rangle^- \boldsymbol{I} + 2\mu \, \boldsymbol{\varepsilon}^- \tag{25}$$

which are based on an additive split of the strain tensor. The phase-field part (22) requires homogeneous Neumann boundary conditions, the elasticity part (23) the usual traction and displacement constraints.

The basic idea of the phase-field fracture model (22) and (23) is to represent cracks by a continuous scalar field c that has a value of one away from the crack and is zero at the crack location. The phase-field serves as a multiplication factor to tensile energy components in (24) such that it locally penalizes the capability of the material to carry tensile stress at the crack location. In this sense, the phase-field idea is conceptually very similar to the fictitious domain approach applied in the finite cell method. The diffusiveness of the crack approximation is controlled by the length-scale parameter l_0. The diffusive approximation of the crack by a continuous phase-field eliminates the need for explicit discontinuities in the mesh. Cracks can be represented independently from the mesh and its topology by the solution of an additional differential equation that completely determines crack nucleation and propagation. The phase-field fracture approach has proven to accurately and robustly capture crack behavior in two and three dimensions for quasi-static fracture [90, 91, 92, 93, 87], dynamic crack propagation [94, 92, 95, 96, 97, 98], at finite strains [99], for fracture in piezo- and ferroelectric materials [100, 101, 102] and for cohesive fracture [103, 104].

5.3.2. TetFCM with local refinement   As the phase-field approximation exhibits sharp local gradients near the crack and is otherwise constant, local refinement is mandatory for an efficient phase-field analysis technology. In the following, we demonstrate the advantages of the tetrahedral finite cell method in terms of highly graded meshes that adaptively refine the crack region in the context of image based stress analysis of a human femur bone. Our analysis is based on a qCT scan of the bone, which can be transferred into the distribution of Young's modulus over the bone [83], and a homogeneous Poisson's ratio $\nu = 0.3$. Figure 18a shows a
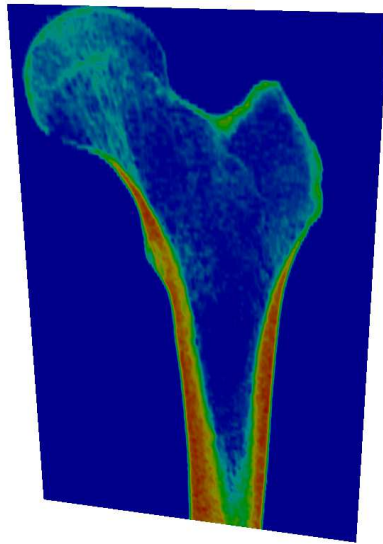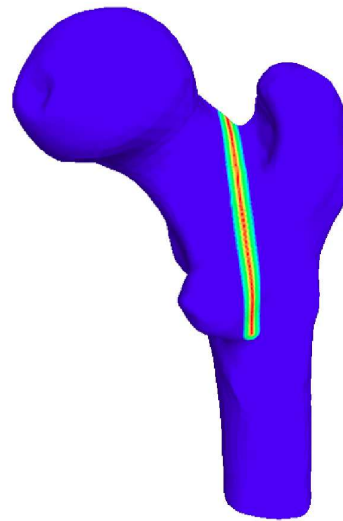
(a) *Slice of the discrete voxel model. The color information refers to Young's modulus (blue - soft, red - very stiff).*

(b) *Immersogeometric tetrahedral mesh of finite cells that locally refine the fracture region.*
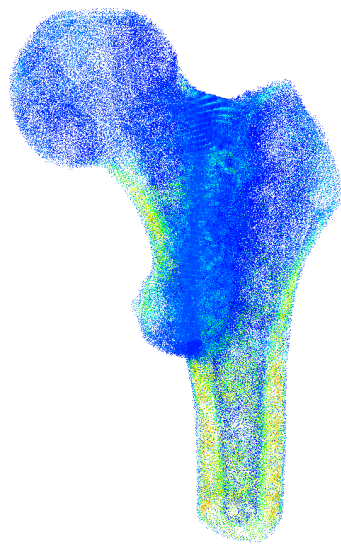
(c) *Quadrature points.*

(d) *Phase-field (blue - 1, red - 0).*

Figure 18: *Image based bone geometry and phase-field approximation of the crack.*

slice of the discrete voxel model[‡]. We use the strain energy based method shown in [88, 89] to impose a crack through the central shaft of the bone. We generate a cloud of h-values using the octree based approach of Section 4.1, where we make use of the assumed local strain energy to drive the depth of the octree. Based on the cloud of h-values we generate an adaptive immersogeometric tetrahedral mesh of the bone, shown in Fig. 18b. The target element size close to the crack location is twice the length parameter l_0 of the phase-field model. Further away from the crack, we allow a significantly larger element size. As described in Section 4.4, we remove all elements that do not have at least one voxel with Young's modulus above a minimum threshold within their support. The quadrature points are shown in Fig. 18c.

Homogeneous Neumann boundary conditions in the phase-field part (22) over the bone surface are automatically imposed without surface quadrature in the TetFCM scheme. In the elasticity part (23), we apply a load of 1000 N on the bone head over a circular area. Displacement boundary conditions at the bone's distal face are weakly enforced with Nitsche's method. To perform quadrature for the formation of matrix and vector components, we triangulate these surfaces. Figure 18d plots the phase-field solution that approximates the sharp crack. A uniform tetrahedral mesh of the bone with an equivalent resolution of the phase-field would yield 250 million degrees of freedom, as compared to approximately 500,000

[‡]Courtesy of Prof. Zohar Yosibash, Dept. of Mechanical Engineering, Ben-Gurion University, Beer-Sheva, Israel; http://www.bgu.ac.il/∼zohary/

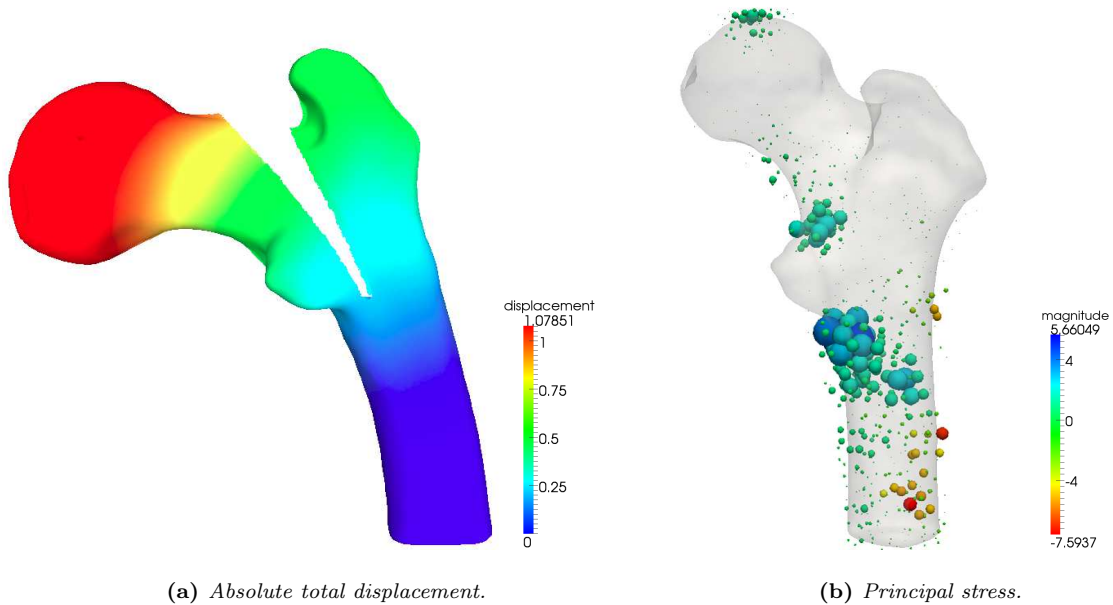(a) *Absolute total displacement.*   (b) *Principal stress.*

Figure 19: *Solution fields for the fractued bone. The color scale varies from blue (small) to red (large) for the displacements and from blue (compression) to red (tension) for the principal stress. The bone surface is reconstructed from the CT data.*

degrees of freedom of the present adaptive mesh. We then solve the elasticity problem, using the phase-field solution in (24). In Figs. 19a and 19b, we plot the total displacement and the principal stress distribution. We observe that maximum stresses appear due to the bending effect in the area where the crack reduces the effective height.

6. SUMMARY AND CONCLUSIONS

In this paper, we extended the finite cell method, so far only used with Cartesian hexahedral elements, to higher-order non-boundary-fitted tetrahedral meshes. With respect to Cartesian FCM, the TetFCM scheme requires three basic adaptations. First, the notion of a Cartesian mesh is abandoned and replaced by the more flexible notion of a general unstructured tetrahedral mesh of the embedding domain. We encouraged the use of open-source meshing tools such as Netgen, in particular for exploiting advanced algorithms for mesh regularization and smoothing that ensure high-quality tetrahedral elements. According to our experience, meshing is extremely fast, even for very large meshes, since there are no geometric constraints other than the simple boundaries of the embedding domain. We also outlined an efficient workflow based on an octree based algorithm for obtaining locally refined tetrahedral meshes. Second, tetrahedral basis functions, for example based on standard quadratic and cubic Lagrange polynomials or based on high-order warped integrated Legendre polynomials, are used. Third, we presented a modification of the Cartesian sub-cell quadrature scheme that achieves accurate integration of intersected tetrahedral elements by increasing quadrature points near the geometric boundary. In particular, we demonstrated that building the tree "from the bottom up" automatically guarantees a high fidelity resolution of the geometry with the finest level of sub-cells available.

Using a series of 3D numerical examples with smooth solutions, we demonstrated that TetFCM yields optimal rates of convergence, when the mesh is refined, and exponential rates, when the polynomial degree of the basis is increased. To illustrate the fundamental importance of accurate geometry resolution, we plotted the stress solution over the immersed boundary for different levels of the integration sub-cells. Using the same tetrahedral mesh with the same basis functions, an analysis with poor geometry resolution of intersected elements resulted in significant errors in boundary stresses, while an analysis with several levels of adaptive sub-cells resulted in very accurate boundary stresses that were indistinguishable from the reference solution. Furthermore, our numerical tests indicated that p-refinement based on the increase of the polynomial degree is computationally more efficient than analysis based on mesh refinement at a fixed polynomial degree, when we use a direct solver. However, our numerical tests also indicated that the decay in conditioning of the discrete system that occurs due to unfavorably cut elements is bounded for mesh refinement at moderate polynomial degrees, but deteriorates without bounds for p-refinement. As a consequence, we could successfully apply an iterative PCG solver for quadratic and cubic meshes, but encountered prohibitive numbers of iterations for high polynomial degrees. The TetFCM is a suitable tool for problems where local refinement is mandatory for efficiency. As an example, we presented an image based phase-field fracture analysis of a human femur bone, where we adaptively resolved sharp gradients in the diffuse phase-field approximation of the crack.

In our view, TetFCM constitutes another opportunity for immersogeometric analysis. It provides access to the advantages of adaptive tetrahedral meshes in the FCM context. From

an analysis point of view, TetFCM does not perform better than Cartesian FCM, and it is merely the choice and preference of the analyst, which FCM scheme is used. TetFCM could also bring us closer to the adoption of FCM capabilities into a commercial FEA package, since adaptive higher-order tetrahedral elements and tetrahedral mesh generators are already available in most commercial production codes.

ACKNOWLEDGEMENTS

REFERENCES

1. S. Haeri and J.S. Shrimpton. On the application of immersed boundary, fictitious domain and body-conformal mesh methods to many particle multiphase flows. *International Journal of Multiphase Flow*, 40:38–55, 2012.
2. A. Calderer, S. Kang, and F. Sotiropoulos. Level set immersed boundary method for coupled simulation of air/water interaction with complex floating structures. *Journal of Computational Physics*, 277:201–227, 2014.
3. B. Avci and P. Wriggers. Direct numerical simulation of particulate flows using a fictitious domain method. In *Numerical Simulations of Coupled Problems in Engineering*, pages 105–127. Springer, 2014.
4. M. Ruess, D. Schillinger, A.I. Özcan, and E. Rank. Weak coupling for isogeometric analysis of non-matching and trimmed multi-patch geometries. *Computer Methods in Applied Mechanics and Engineering*, 269:46–71, 2014.
5. A.P. Nagy and D.J. Benson. On the numerical integration of trimmed isogeometric elements. *Computer Methods in Applied Mechanics and Engineering*, 284:165–185, 2015.
6. M. Breitenberger, A. Apostolatos, B. Philipp, R. Wüchner, and K.-U. Bletzinger. Analysis in computer aided design: Nonlinear isogeometric b-rep analysis of shell structures. *Computer Methods in Applied Mechanics and Engineering*, 284:401–457, 2015.
7. J. Parvizian, A. Düster, and E. Rank. Topology optimization using the finite cell method. *Optimization and Engineering*, 13:57–78, 2012.
8. J. Benk, H.-J. Bungartz, M. Mehl, and M. Ulbrich. Immersed boundary methods for fluid-structure interaction and shape optimization within an FEM-based PDE toolbox. In *Advanced Computing*, pages 25–56. Springer, 2013.
9. I. Borazjani, L. Ge, and F. Sotiropoulos. Curvilinear immersed boundary method for simulating fluid structure interaction with complex 3D rigid bodies. *Journal of Computational Physics*, 227(16):7587–7620, 2008.
10. M.-C. Hsu, D. Kamensky, Y. Bazilevs, M.S. Sacks, and T.J.R. Hughes. Fluid–structure interaction analysis of bioprosthetic heart valves: significance of arterial wall deformation. *Computational Mechanics*, 54(4):1055–1071, 2014.
11. F. Sotiropoulos and X. Yang. Immersed boundary methods for simulating fluid–structure interaction. *Progress in Aerospace Sciences*, 65:1–21, 2014.
12. D. Kamensky, M.-C. Hsu, D. Schillinger, J. A. Evans, A. Aggarwal, Y. Bazilevs, M. S. Sacks, and T. J. R. Hughes. An immersogeometric variational framework for fluid–structure interaction: application to bioprosthetic heart valves. *Computer Methods in Applied Mechanics and Engineering*, 284:1005–1053, 2015.
13. B.I. Wohlmuth. Discretization techniques based on domain decomposition. *Lecture Notes in Computational Science and Engineering, Vol. 17*, 2001.
14. A. Hansbo and P. Hansbo. An unfitted finite element method, based on Nitsche's method, for elliptic interface problems. *Computer Methods in Applied Mechanics and Engineering*, 191:537–552, 2002.
15. S. Fernández-Méndez and A. Huerta. Imposing essential boundary conditions in mesh-free methods. *Computer Methods in Applied Mechanics and Engineering*, 193:1257–1275, 2004.
16. A. Embar, J. Dolbow, and I. Harari. Imposing Dirichlet boundary conditions with Nitsche's method and spline-based finite elements. *International Journal for Numerical Methods in Engineering*, 83:877–898, 2010.
17. C. Annavarapu, M. Hautefeuille, and J.E. Dolbow. A robust Nitsche's formulation for interface problems. *Computer Methods in Applied Mechanics and Engineering*, 225:44–54, 2012.
18. I. Harari and E. Grosu. A unified approach for embedded boundary conditions for fourth-order elliptic problems. *International Journal for Numerical Methods in Engineering*, accepted for publication, 2014.
19. Y. Guo and M. Ruess. Nitsche's method for a coupling of isogeometric thin shells and blended shell structures. *Computer Methods in Applied Mechanics and Engineering*, 284:881–905, 2015.
20. R. Becker, P. Hansbo, and R. Stenberg. A finite element method for domain decomposition with non-matching grids. *Mathematical Modelling and Numerical Analysis*, 37(2):209–225, 2003.
21. A. Apostolatos, R. Schmidt, R. Wüchner, and K.-U. Bletzinger. A Nitsche-type formulation and comparison of the most common domain decomposition methods in isogeometric analysis. *International Journal for Numerical Methods in Engineering*, 97(7):473–504, 2014.
22. Y. Bazilevs and T.J.R. Hughes. Weak imposition of Dirichlet boundary conditions in fluid mechanics. *Computers & Fluids*, 36:12–26, 2007.
23. M.C. Hsu, I. Akkerman, and Y. Bazilevs. Wind turbine aerodynamics using ALE-VMS: validation and the role of weakly enforced boundary conditions. *Computational Mechanics*, 50:499–511, 2012.
24. A. Stavrev. *The role of higher-order geometry approximation and accurate quadrature in NURBS based immersed boundary methods*. Master Thesis, Technische Universität München, 2012.

25. L. Kudela. *Highly Accurate Subcell Integration in the Context of The Finite Cell Method*. Master Thesis, Technische Universität München, 2013.

26. T.J.R. Hughes, J.A. Cottrell, and Y. Bazilevs. Isogeometric analysis: CAD, finite elements, NURBS, exact geometry and mesh refinement. *Computer Methods in Applied Mechanics and Engineering*, 194:4135–4195, 2005.

27. J.A. Cottrell, T.J.R. Hughes, and Y. Bazilevs. *Isogeometric analysis: Towards Integration of CAD and FEA*. John Wiley & Sons, 2009.

28. J. Parvizian, A. Düster, and E. Rank. Finite cell method: *h*- and *p*- extension for embedded domain methods in solid mechanics. *Computational Mechanics*, 41:122–133, 2007.

29. A. Düster, J. Parvizian, Z. Yang, and E. Rank. The finite cell method for three-dimensional problems of solid mechanics. *Computer Methods in Applied Mechanics and Engineering*, 197:3768–3782, 2010.

30. D. Schillinger. *The p- and B-spline versions of the geometrically nonlinear finite cell method and hierarchical refinement strategies for adaptive isogeometric and embedded domain analysis*. Dissertation, Technische Universität München, http://d-nb.info/103009943X/34, 2012.

31. H. Samet. *The design and analysis of spatial data structures*, volume 199. Addison-Wesley Reading, MA, 1990.

32. H. Samet. *Foundations of Multidimensional and Metric Data Structures*. Morgan Kaufmann Publishers, 2006.

33. N. Zander. *The Finite Cell Method for linear thermoelasticity*. Master Thesis, Technische Universität München, 2011.

34. M. Ruess, D. Schillinger, Y. Bazilevs, V. Varduhn, and E. Rank. Weakly enforced essential boundary conditions for NURBS-embedded and trimmed NURBS geometries on the basis of the finite cell method. *International Journal for Numerical Methods in Engineering*, 95(10):811–846, 2013.

35. D. Schillinger and E. Rank. An unfitted *hp* adaptive finite element method based on hierarchical B-splines for interface problems of complex geometry. *Computer Methods in Applied Mechanics and Engineering*, 200(47–48):3358–3380, 2011.

36. D. Schillinger, A. Düster, and E. Rank. The *hp-d* adaptive finite cell method for geometrically nonlinear problems of solid mechanics. *International Journal for Numerical Methods in Engineering*, 89:1171–1202, 2012.

37. D. Schillinger, L. Dede', M.A. Scott, J.A. Evans, M.J. Borden, E. Rank, and T.J.R. Hughes. An isogeometric design-through-analysis methodology based on adaptive hierarchical refinement of NURBS, immersed boundary methods, and T-spline CAD surfaces. *Computer Methods in Applied Mechanics and Engineering*, 249-250:116–150, 2012.

38. M. Joulaian and A. Düster. Local enrichment of the finite cell method for problems with material interfaces. *Computational Mechanics*, 52:741–762, 2013.

39. N. Zander, T. Bog, S. Kollmannsberger, D. Schillinger, and E. Rank. Multi-Level hp-Adaptivity: High-Order mesh adaptivity without the difficulties of constraining hanging nodes. *Computational Mechanics*, accepted for publication, 2015.

40. Z. Yang, M. Ruess, S. Kollmannsberger, A. Düster, and E. Rank. An efficient integration technique for the voxel-based finite cell method. *International Journal for Numerical Methods in Engineering*, 91:457–471, 2012.

41. D. Schillinger, S. Kollmannsberger, R.-P. Mundani, and E. Rank. The finite cell method for geometrically nonlinear problems of solid mechanics. *IOP Conference Series: Material Science and Engineering*, 10:012170, 2010.

42. D. Schillinger, M. Ruess, N. Zander, Y. Bazilevs, A. Düster, and E. Rank. Small and large deformation analysis with the *p*- and B-spline versions of the Finite Cell Method. *Computational Mechanics*, 50(4):445–478, 2012.

43. N. Zander, S. Kollmannsberger, M. Ruess, Z. Yosibash, and E. Rank. The Finite Cell Method for linear thermoelasticity. *Computers & Mathematics with Applications*, 64(11):3527–3541, 2012.

44. A. Düster, H.-G. Sehlhorst, and E. Rank. Numerical homogenization of heterogeneous and cellular materials utilizing the finite cell method. *Computational Mechanics*, 50:413–431, 2012.

45. M. Ruess, D. Tal, N. Trabelsi, Z. Yosibash, and E. Rank. The finite cell method for bone simulations: Verification and validation. *Biomechanics and Modeling in Mechanobiology*, 11(3):425–437, 2012.

46. C.V. Verhoosel, G.J. van Zwieten, B. van Rietbergen, and R. de Borst. Image-based goal-oriented adaptive isogeometric analysis with application to the micro-mechanical modeling of trabecular bone. *Computer Methods in Applied Mechanics and Engineering*, 284:138–164, 2015.

47. A. Abedian, J. Parvizian, A. Düster, and E. Rank. The finite cell method for the J_2 flow theory of plasticity. *Finite Elements in Analysis and Design*, 69:37–47, 2013.

48. M. Ranjbar, M. Mashayekhi, J. Parvizian, A. Düster, and E. Rank. Using the finite cell method to predict crack initiation in ductile materials. *Computational Materials Science*, 82:427 – 434, 2014.

49. C. Willberg, S. Duczek, J.M. Vivar Perez, D. Schmicker, and U. Gabbert. Comparison of different higher

Copyright © 2000 John Wiley & Sons, Ltd.

Prepared using nmeauth.cls

Int. J. Numer. Meth. Engng 2000; **00**:1–6

order finite element schemes for the simulation of Lamb waves. *Computer Methods in Applied Mechanics and Engineering*, 241–244:246–261, 2012.

50. S Duczek, M Joulaian, A Düster, and U Gabbert. Numerical analysis of lamb waves using the finite and spectral cell methods. *International Journal for Numerical Methods in Engineering*, 99(1):26–53, 2014.

51. M. Joulaian, S. Duczek, U. Gabbert, and A. Düster. Finite and spectral cell method for wave propagation in heterogeneous materials. *Computational Mechanics*, 54(1):661–675, 2014.

52. F. Xu, D. Schillinger, D. Kamensky, V. Varduhn, C. Wang, and M.-C. Hsu. The tetrahedral finite cell method for fluids: Immersogeometric analysis of turbulent flow around complex geometries. *Submitted to Computer & Fluids*, 2015.

53. D. Schillinger and M. Ruess. The Finite Cell Method: A review in the context of higher-order structural analysis of CAD and image-based geometric models. *Archives of Computational Methods in Engineering*, doi:10.1007/s11831-014-9115-y, 2014.

54. N. Zander, T. Bog, M. Elhaddad, R. Espinoza, H. Hu, A.F. Joly, C. Wu, P. Zerbe, A. Düster, S. Kollmannsberger, J. Parvizian, M. Ruess, D. Schillinger, and E. Rank. FCMLab: A finite cell research toolbox for MATLAB. *Advances in Engineering Software*, 74:49–63,, 2014.

55. Netgen Mesh Generator, developed by j. schoeberl, http://sourceforge.net/projects/netgen-mesher/, 2015.

56. Joachim Schöberl. NETGEN. An advancing front 2D/3D-mesh generator based on abstract rules. *Computing and Visualization in Science*, 1(1):41–52, 1997.

57. K.-J. Bathe. *Finite Element Procedures*. Prentice-Hall, 1996.

58. T.J.R. Hughes. *The Finite Element Method: Linear Static and Dynamic Finite Element Analysis*. Dover Publications, 2000.

59. O.C. Zienkiewicz and R.L. Taylor. *The Finite Element Method – The Basis*, volume 1. Butterworth-Heinemann, 6th edition, 2005.

60. J. Bonet and R. Wood. *Nonlinear Continuum Mechanics for Finite Element Analysis*. Cambridge University Press, 2008.

61. E. Süli and D.F. Mayers. *An introduction to numerical analysis*. Cambridge University Press, 2003.

62. C. Felippa. Advanced finite element methods. *Course notes, available online at http://www.colorado.edu/engineering/CAS/courses.d/AFEM.d/Home.html*.

63. C. Canuto, M.Y. Hussaini, A. Quarteroni, and T.A. Zang. *Spectral Methods: Fundamentals in Single Domains*. Springer, 2006.

64. B. Szabó and I. Babuška. *Finite Element Analysis*. Wiley, 1991.

65. A. Düster, H. Bröker, and E. Rank. The *p*-version of the finite element method for three-dimensional curved thin walled structures. *International Journal for Numerical Methods in Engineering*, 52:673–703, 2001.

66. Z. Wassouf. *Die Mortar Methode für Finite Elemente hoher Ordnung.* Dissertation, Technische Universität München, 2010.

67. M. Dubiner. Spectral methods on triangles and other domains. *Journal of Scientific Computing*, 6(4):345–390, 1991.

68. S.J. Sherwin and G.E. Karniadakis. A new triangular and tetrahedral basis for high-order (hp) finite element methods. *International Journal for Numerical Methods in Engineering*, 38(22):3775–3802, 1995.

69. G.E. Karniadakis and S.J. Sherwin. *Spectral/hp Element Methods for Computational Fluid Dynamics*. Oxford University Press, 2005.

70. T.C. Warburton, S.J. Sherwin, and G.E. Karniadakis. Basis functions for triangular and quadrilateral high-order elements. *SIAM Journal on Scientific Computing*, 20(5):1671–1695, 1999.

71. Timothy Warburton. Spectral/hp methods on polymorphic multidomains: Algorithms and applications, 1999.

72. B.A. Szabó, A. Düster, and E. Rank. The p-version of the Finite Element Method. In E. Stein, R. de Borst, and T. J. R. Hughes, editors, *Encyclopedia of Computational Mechanics*, volume 1, chapter 5, pages 119–139. John Wiley & Sons, 2004.

73. Zohar Yosibash. *p*-fems in biomechanics: Bones and arteries. *Computer Methods in Applied Mechanics and Engineering*, 249:169–184, 2012.

74. P. Keast. Moderate-degree tetrahedral quadrature formulas. *Computer Methods in Applied Mechanics and Engineering*, 55:339–348, 1986.

75. A. Stroud. *Approximate Calculation of Multiple Integrals*. Prentice Hall, 1971.

76. R. Cools. Monomial cubature rules since "Stroud": A compilation - Part 2. *Journal of Computational and Applied Mathematics*, 112(1):21–27, 1999.

77. D. Schillinger, S.J. Hossain, and T.J.R. Hughes. Reduced Bézier element quadrature rules for quadratic and cubic splines in isogeometric analysis. *Computer Methods in Applied Mechanics and Engineering*, 277:1–45, 2014.

78. P. Hillion. Numerical integration on a triangle. *International Journal for Numerical Methods in*

Engineering, 11:797–815, 1977.

79. P. Hillion. Numerical integration on a tetrahedron. *Calcolo*, 18(2):117–130, 1979.

80. Trilinos Version 11.12, Sandia National Laboratories, http://trilinos.org/, 2015.

81. Y. Saad. *Iterative methods for sparse linear systems*. Society for Industrial and Applied Mathematics, 2003.

82. G.H. Golub and C. Van Loan. *Matrix Computations*. Johns Hopkins University Press, 1996.

83. J.H. Keyak and Y. Falkinstein. Comparison of in situ and in vitro CT scan-based finite element model predictions of proximal femoral fracture load. *Medical Engineering & Physics*, 25(9):781–787, 2003.

84. N. Trabelsi, Z. Yosibash, and C. Milgrom. Validation of subject-specific automated p-fe analysis of the proximal femur. *Journal of Biomechanics*, 42(3):234–241, 2009.

85. G.A. Francfort and J.-J. Marigo. Revisiting brittle fracture as an energy minimization problem. *Journal of the Mechanics and Physics of Solids*, 46(8):1319–1342, 1998.

86. B. Bourdin, G.A. Francfort, and J.-J. Marigo. The variational approach to fracture. *Journal of Elasticity*, 91(1-3):5–148, 2008.

87. C. Miehe, M. Hofacker, and F. Welschinger. A phase field model for rate-independent crack propagation: Robust algorithmic implementation based on operator splits. *Computer Methods in Applied Mechanics and Engineering*, 199(45):2765–2778, 2010.

88. M.J. Borden, M.A. Scott, J.A. Evans, and T.J.R. Hughes. Isogeometric finite element data structures based on Bézier extraction of NURBS. *International Journal for Numerical Methods in Engineering*, 87:15–47, 2011.

89. D. Schillinger, M.J. Borden, and H.K. Stolarski. Isogeometric collocation for phase-field fracture models. *Computer Methods in Applied Mechanics and Engineering*, 284:583–610, 2015.

90. G. Lancioni and G.F. Royer-Carfagni. The variational approach to fracture mechanics. A practical application to the French Panthéon in Paris. *Journal of Elasticity*, 95(1-2):1–30, 2009.

91. C. Kuhn and R. Müller. A continuum phase field model for fracture. *Engineering Fracture Mechanics*, 77(18):3625–3634, 2010.

92. M.J. Borden, C.V. Verhoosel, M.A. Scott, Hughes T.J.R., and C.M. Landis. A phase-field description of dynamic brittle fracture. *Computer Methods in Applied Mechanics and Engineering*, 217–220:77–95, 2012.

93. C. Miehe, F. Welschinger, and M. Hofacker. Thermodynamically consistent phase-field models of fracture: Variational principles and multi-field FE implementations. *International Journal for Numerical Methods in Engineering*, 83(10):1273–1311, 2010.

94. M.J. Borden, T.J.R. Hughes, C.M. Landis, and C.V. Verhoosel. A higher-order phase-field model for brittle fracture: Formulation and analysis within the isogeometric analysis framework. *Computer Methods in Applied Mechanics and Engineering*, 273:100 – 118, 2014.

95. B. Bourdin, C.J. Larsen, and C.L. Richardson. A time-discrete model for dynamic fracture based on crack regularization. *International Journal of Fracture*, 168(2):133–143, 2011.

96. M. Hofacker and C. Miehe. Continuum phase field modeling of dynamic fracture: variational principles and staggered FE implementation. *International Journal of Fracture*, 178(1-2):113–129, 2012.

97. M. Hofacker and C. Miehe. A phase field model of dynamic fracture: Robust field updates for the analysis of complex crack patterns. *International Journal for Numerical Methods in Engineering*, 93(3):276–301, 2013.

98. A. Schlüter, A. Willenbücher, C. Kuhn, and R. Müller. Phase field approximation of dynamic brittle fracture. *Computational Mechanics*, 2014.

99. C. Miehe and L.-M. Schänzel. Phase field modeling of fracture in rubbery polymers. part i: Finite elasticity coupled with brittle failure. *Journal of the Mechanics and Physics of Solids*, 65:93–113, 2014.

100. C. Miehe, F. Welschinger, and M. Hofacker. A phase field model of electromechanical fracture. *Journal of the Mechanics and Physics of Solids*, 58(10):1716–1740, 2010.

101. Z.A. Wilson, M.J. Borden, and C.M. Landis. A phase-field model for fracture in piezoelectric ceramics. *International Journal of Fracture*, 183(2):135–153, 2013.

102. A. Abdollahi and I. Arias. Phase-field modeling of fracture in ferroelectric materials. *Archives of Computational Methods in Engineering*, doi:10.1007/s11831-014-9118-8, 2014.

103. C.V. Verhoosel and R. de Borst. A phase-field model for cohesive fracture. *International Journal for Numerical Methods in Engineering*, 96(1):43–62, 2013.

104. J. Vignollet, S. May, R. de Borst, and C.V. Verhoosel. Phase-field models for brittle and cohesive fracture. *Meccanica*, doi:10.1007/s11012-013-9862-0, 2014.

Copyright © 2000 John Wiley & Sons, Ltd.

Prepared using nmeauth.cls

Int. J. Numer. Meth. Engng 2000; **00**:1–6