Ingenieurfakultät Bau Geo Umwelt
Technische Universität München

The Finite Cell Method: towards engineering applications

Dr.-Ing. Stefan Kollmannsberger

Kumulative Habilitationsschrift

Commission	Prof. Dr. rer. nat Ernst Rank	Chariman
	Prof. Dr.-Ing. Kai-Uwe Bletzinger	Member
	Prof. Alessandro Reali (PhD)	Member

submitted: August 30, 2019

Involved Institutions

Ingenieurfakultät Bau Geo Umwelt
Technische Universität München
Arcisstraße 21
D-80333 München

Department of Civil Engineering and Architecture
University of Pavia
via Ferrata 3
27100 Pavia, Italy

Contents

Peamble: a personal note

This section will be filled in the final, published version. It will contain an honest, short and concise thank you to everyone who has contributed to the success of this thesis.

1 Introduction and basic formulation

The Finite Cell Method (FCM) is an immersed-boundary method based on higher order shape functions. The primary goal of the method is to avoid having to generate boundary-conforming meshes. Nevertheless, it provides high-order convergence rates.

Immersed-boundary methods have been in the focus of research in Computational Mechanics and Numerical Mathematics for decades. Even if the term 'immersed-boundary' seems to have appeared in [1] for the first time, similar approaches – such as embedded domain or fictitious domain methods – were proposed as early as in the 1960s, see, e.g. [2, 3]. Furthermore, XFEM (eXtended Finite Element Methods) [4] and EPUM (Extended Partition of Unity Methods), see, e.g. [5, 6, 7], are closely related. A review on immersed-boundary methods is given in e.g. [8, 9, 10]. Common to all these techniques is the idea of embedding a given domain of computation into a larger one with a simple shape (e.g. a square or a cube), which can be meshed trivially e.g. in a regular grid. Their difference lies in the underlying discretization of the unknown variables.

Whereas the large majority of immersed-boundary methods use low-order finite elements or, in some cases, spectral elements, the Finite Cell Method, which was introduced in [11, 12], combines the fictitious domain idea with high-order finite elements. To this end, the physical domain Ω_{phys} is embedded into the fictitious domain Ω_{fict}. The resulting $\Omega_{\cup}$ can then be discretized easily, thus preventing a complex mesh generation for Ω_{phys}. The original problem is recovered by multiplying the element integrands with the indicator function α of the domain $\Omega_{\cup}$, as depicted in fig. 1. For points located inside the domain Ω_{phy}, the indicator function equals $\alpha = 1$. Otherwise, it is set to $\alpha = 0$ or to very small positive values to avoid conditioning problems. The resulting non-boundary-conforming high-order elements are denoted as cells, giving the method its name. The benefits of this approach are (as for all immersed-boundary methods) a dramatically reduced effort for mesh generation and, specific to the FCM, high convergence rates due to the use of high-order Ansatz functions – which usually leads to excellent accuracy.

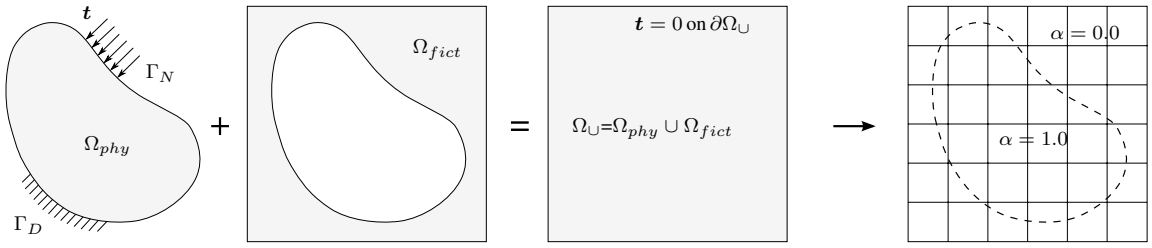


Figure 1: The idea of the Finite Cell Method: The physical domain Ω_{phys} is embedded into the fictitious domain Ω_{fict}. The resulting $\Omega_{\cup}$ can be discretized easily, preventing a complex mesh generation for Ω_{phys}. The original problem is recovered by the indicator function α (see [11, 12] for the original idea, while the picture above is taken from [13]).

The FCM rests on a sound mathematical basis and is able to deliver exponential convergence rates in the energy norm up to a modeling error proportional to $\sqrt{\alpha_{fict}}$, see [14] for a mathematical analysis of a one-dimensional elliptic problem. These favorable properties have also been verified numerically for a large number of different applications (two- and three-dimensional).

This treatise is structured into four main sections. This very brief introductory section is followed by two sections that are dedicated to the two key ingredients of the Finite Cell Method. Section 2 lays out the geometric models which can form the basic description of the physical domain under investigation, and section 3 gives an insight into the type of discretizations which were recently developed in the scope of the Finite Cell Method. Both of these sections start with an overview which is intended to give a broader picture and which puts the following subsections into context. In order to keep the extent of this treatise within reason, the subsections cover the related topics as an introduction would – but citing the relevant literature. The two main sections 2 and 3 then lead up to the final section 4, which is dedicated to engineering applications of the Finite Cell Method. For selected topics, and in the sense of a cumulative treatise, these extensions to the Finite Cell Method are presented in detail, which can be seen as the core of the author's scientific work in the scope of this treatise.

2 Geometric models

In the Finite Cell Method, the geometry of the object of interest is described by the function α, see also fig. 1. This function may have many origins, whereby the following classification is convenient. α may be:

- generated from explicit geometric models. These can be curves or surfaces of all imaginable types – such as explicit, implicit, or parametric functions. Their complexity ranges from simple surface triangulations, which are e.g. popular for visualization purposes on graphic cards, to geometrically smoother representations such as NURBS. The latter are just one popular representative among many other possible spline representations stemming from Computer Aided Design (CAD) models.
- given by volumetric imaging methods such as Computed Tomographic (CT) scans. Effectively, α is then represented by a set of discrete values organized in a spatial matrix.
- given in form of point clouds generated from images.
- an unknown function to be computed. Possible examples are topology optimization or generative manufacturing processes. Herein, the link to more classical phase field approaches is apparent.

The next section gives a general overview over a broad selection of geometric models for which the Finite Cell Method has been extended. Specific focus is then given to direct computations on constructive solid geometry (in section 2.2) as well as flawed (B-rep models section 2.3) by providing the corresponding journal publications. Section 2.4 then lays out some important details of how a direct computational analysis on point clouds can be carried out using the Finite Cell Method, for which a peer-reviewed journal publication is not yet available.

2.1 Overview

The generality with which geometry can be used in the Finite Cell Method was (naturally) not fully covered in the first publications on the FCM [11, 12], in which implicit analytical functions (plate with a hole) were used or in which the physical domain was described by voxels. First extensions were published in [15]. The underlying motivation for this extension was to apply the FCM to shells that were represented as thin solids. It is still quite challenging to generate three-dimensional meshes for classic high-order boundary-conforming finite elements, while two dimensions are, naturally, easier to handle. An extension from two to three dimensions for the special case of thin solids is published in [16], where a boundary-conforming meshing of a reference surface in the parameter domain of the geometric model is proposed. The mesh is then extruded to the third dimension. This proved to be a valuable approach, though limited to volumes based on extrusions as well as a fixed number of shell intersections. Furthermore, the methodology suffers from difficulties caused by strongly curved boundaries. These can lead to unnecessary local refinements due to involved mapping procedures in often badly parametrized geometries, and they can cause flipped or overlapping elements. As a remedy, the application of the FCM to thin solids is proposed in [15], where a coarse non-boundary conforming computational grid is utilized. This grid is a uniform subdivision of a rectangular patch in the parameter plane. The grid forms the finite cells, which are then blended[1] towards those B-spline patches of the CAD model which constitute the upper

[1] by quasi-regional blending

and lower surface of the geometric model. The physical geometry can then be recovered at integration level by using the trimming curves of the CAD model.

The FCM perfectly complements an alternative approach to integrate analysis and design: Isogeometric Analysis (IGA) [17, 18]. Its fundamental idea is to use the functions describing the geometry directly for its computational analysis. This makes sense in the line of its initial idea – under the assumption that the geometry is drawn in a water-tight boundary conforming way in which no trimming is necessary. However, trimming can be introduced into IGA by the FCM. This point is illustrated in fig. 2, which was e.g. presented in [19] to promote the idea of unifying FCM and IGA specifically for the computational analysis of thin solids.

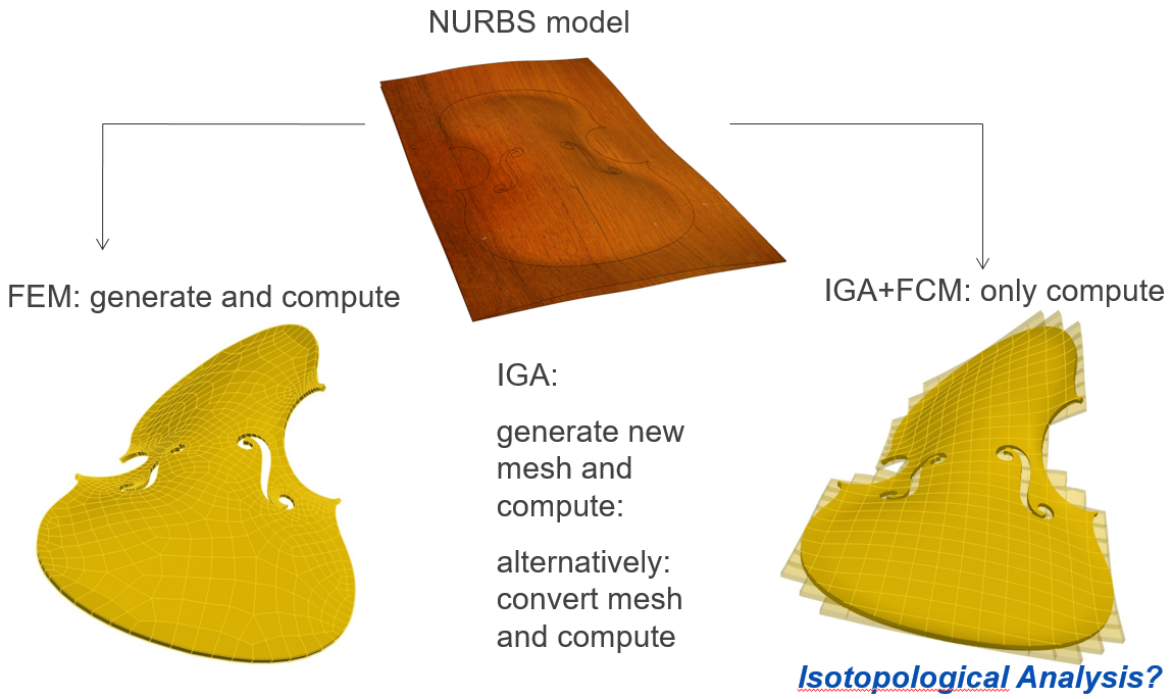


Figure 2: IGA and FCM in 2011: integration of CAD-CAM from [19]

However, its implication is more general because trimming is commonly used in CAD since it is inconvenient to compose every geometry out of conforming patches consisting of topological quadrilaterals or hexahedrals: A designer does not want to draw meshes. Instead, CAD systems mostly trim patches to represent complicated geometries: a process so natural to the FCM trimming was introduced to Isogeometric Analysis as a feature on the computational side of the CAM process. This unification was first addressed in [20]. To this end, B-spline basis functions spanned on a patch can be used directly as a discretization of the field variables in the IGA sense, and the true geometry of the system can be carved out using the trimming curves of the geometric model, see fig. 3.

While many other approaches to trimming in IGA were developed independently at the time, see e.g. [21], the publication [20] pointed out a further inconvenience in the transfer from CAD to CAM that was not covered by the initial versions of IGA: trimming of models leads to changes in the topology of the body, which in turn requires to generate a new computational mesh. A minimal example is given in fig. 4, where two simple bodies are depicted.

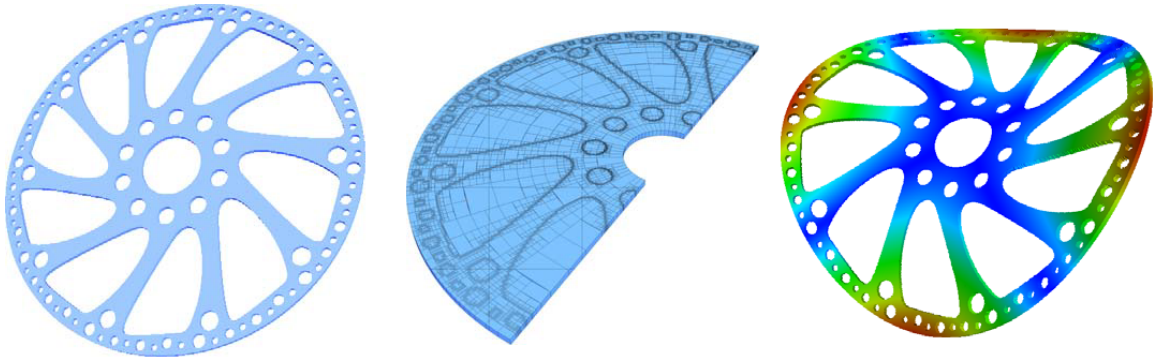


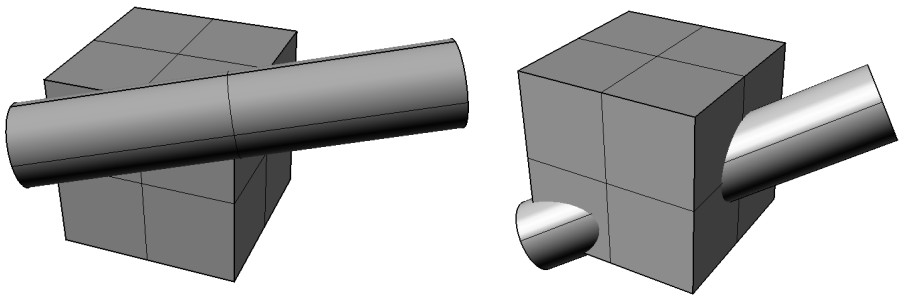
Figure 3: Trimmed FCM-IGA analysis: (left) NURBS model of the geometry, (center) integration by octree, (right) eigenvalue computation [20]

They are constructed of the same cylinder, substracted from the same cuboid by application of the identical corresponding Boolean operation of a difference $\setminus$ in constructive solid geometry (CSG). In this simple case, not even the trimming operation itself changes. The only difference is in the positions at which the bodies are located before trimming occurs. Yet, in a boundary conforming computational method such as IGA (or classic p-FEM), the resulting mesh topology must be completely different for the two bodies, i.e. a new mesh would have to be drawn. Instead, if the point of view of the CAD system is directly translated to computational analysis by means of the FCM, the required point membership test can be carried out directly on the same CSG-tree used for the Boolean operations to construct the CAD model.

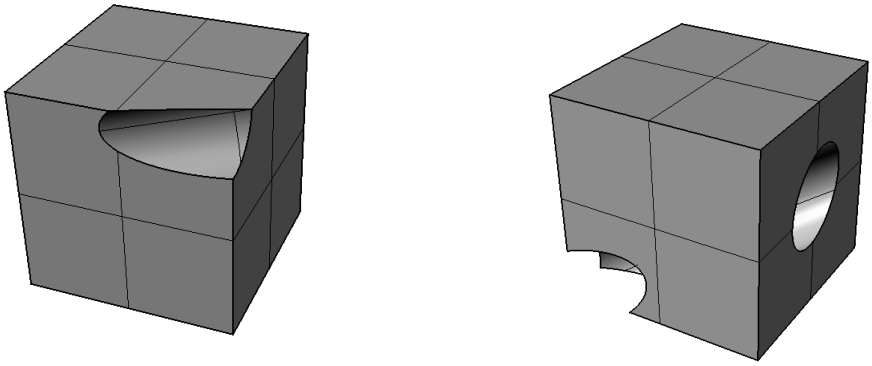
It is long known that boundary representations often lead to non water-tight geometric models. This flaw was frequently pointed out in several talks concerning IGA. Its most famous incarnation is the leaking teapot example, which exhibits a gap at the junction of the patch describing the vessel with the one forming the nozzle. This flaw, among with others, directly carries over to computational analysis, where gaps unintentionally lead to disconnected bodies that are unable to transfer fluxes such as forces. To the contrary, a pure CSG design is much less prone to this type of errors. Additionally, it is not necessary to stick to simple primitive objects such as basic elements in a CSG construction. More complicated primitives can naturally be allowed, see e.g. fig. 5, which are obtained by sweeps or lofts of two-dimensional objects along a path.

Robust point membership computations may be defined on these primitives as well, using ray-tests on their two-dimensional counterparts. The details of these extensions, published in [22], are laid out in section 2.2, where a water-tight FCM analysis based on extended CSG geometries is presented. Fully explicit representation of the volume of the geometric model are used in the entire process – from the design of the model to its computation. Indeed, the main point of the publication is to demonstrate that issues related to boundary representations may be avoided completely in the entire CAD to CAM chain by staying within the framework of constructive solid geometry. An example is depicted in fig. 6, where the results of a direct analysis of the extended CSG model defined in fig. 5 by FCM are provided.

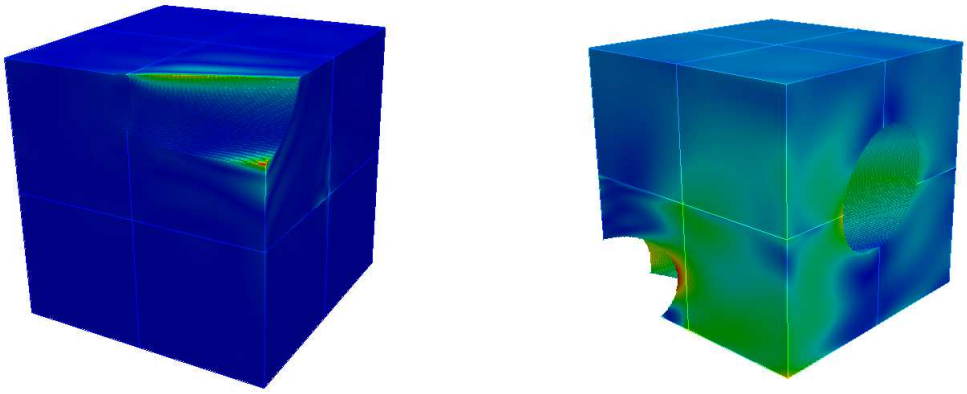
Unfortunately, this pure CSG approach has its limits: for example, CAD systems require a reconstruction of the surface of the body, simply for reasons of visualization, and an intersection (although 'dirty' and non-water tight) may be computed faster on explicitly described boundaries. For these and other reasons, CAD kernels usually maintain B-rep representations



(a) cylinder at two different positions



(b) CSG bodies after Boolean operation



(c) von Mises stresses

Figure 4: Completely different topologies resulting from identical CSG operations – only differing in the positioning of the bodies relative to each other [20]

in parallel, even if they are based on CSG. Thus, B-rep models continue to be widely used in the CAD community[2].

Despite the CAM community's efforts to unify CAD and CAM, both are still mostly disjoint. This is possible because CAM is a downstream application to CAD. It is, thus, rather difficult

[2]Moreover, most CAD users are not interested in a volumetric description of the body but rather in its superficial appearance. Extensions that combine volumetric modeling and surface descriptions in a more integrated manner have only recently gained attention, see [23]

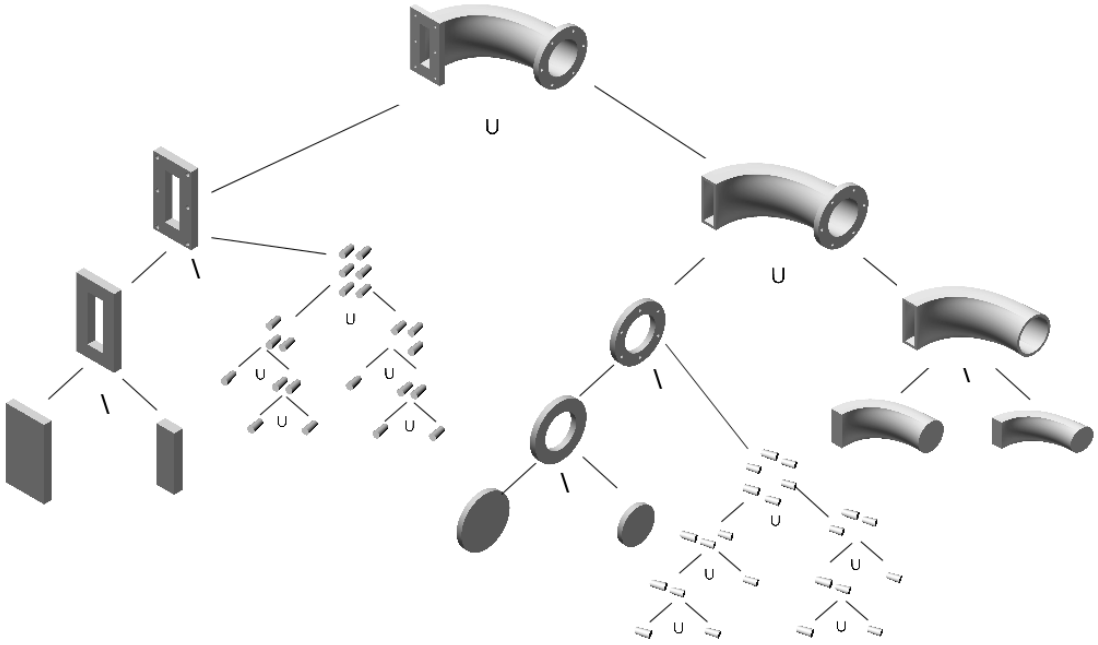


Figure 5: Extended CSG and FCM [22]

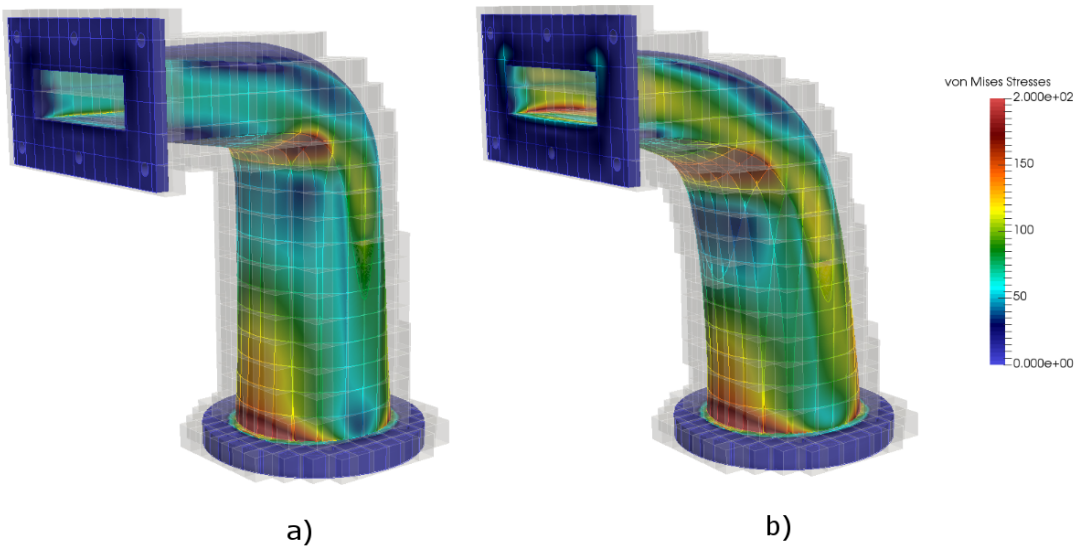
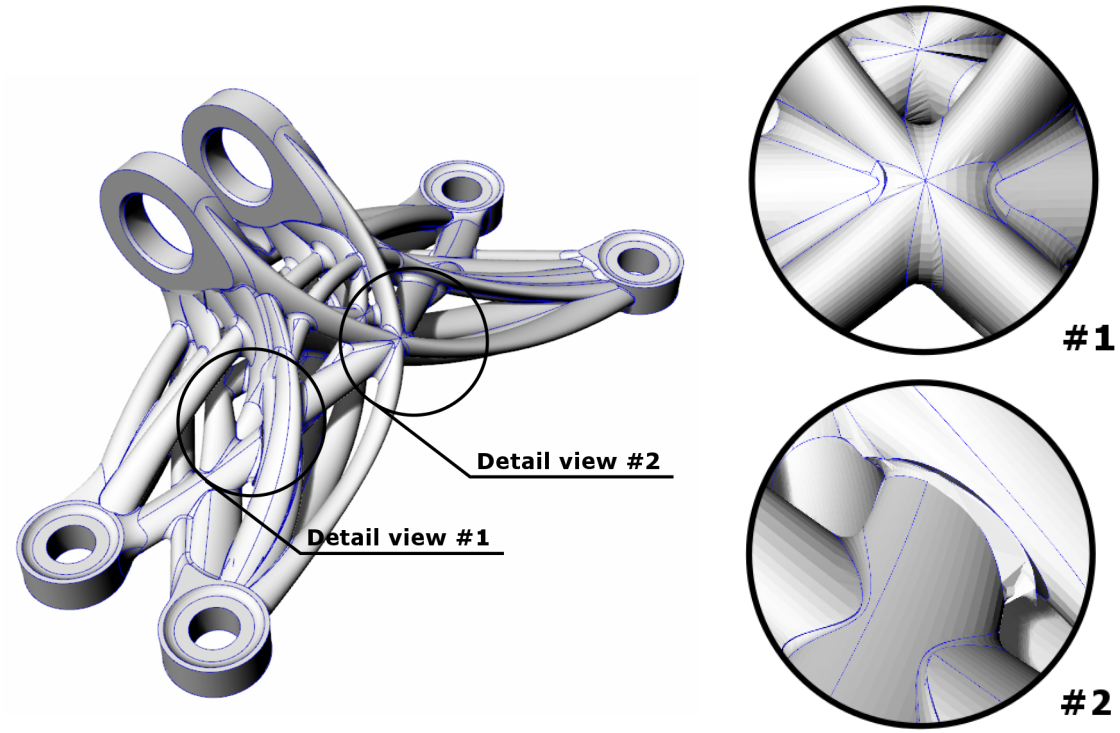


Figure 6: Stress computations on extended CSG models [22]

for CAD designers to immediately realize what consequences different modeling choices have for computational mechanics applications carried out on the same model. Direct connections between the parametrization of the geometry and the representation of the unknown field variables (such as displacements in a computational analysis) have further side issues beyond the problematic features of gaps. For example, singular points in the geometry, such as poles



(a) STL model **(b)** Details

Figure 7: Blue lines denote open edges: In many cases, the free edges are fairly close. However, as can be seen in the detailed views, some of the gaps and openings are quite large.

of a sphere, carry over their singularity to the computational analysis. Because of these and other issues, the term *analysis-aware design* [24] was coined.

To the contrary, the FCM offers the possibility to re-interpret analysis-aware design as *design-aware analysis*. Surely, the idea of working in CAM with what one obtains from CAD is, by itself, not exactly revolutionary. An obvious example is that the most common approach to FE-analysis is to generate a boundary-conforming mesh directly from the CAD geometry. The underlying problem therein is that, as a consequence, the creation of a mesh relies on the parametrization of the domain (boundary) itself and fails for flawed models. This dependency is broken by the FCM. Its idea is to radically disconnect the material description of the physical space from the description of its primal variables. All that is needed is a robust point membership test for all integration points in the computational domain.

The key observation is that it is possible to construct such a test even for flawed geometric models by adding a certain 'blindness' to the considered flaws, up to a controllable size within the point membership tests. This enables a computational analysis on 'dirty' (volumetric) CAD models without first having to heal defects in its topology. Consider the clamp depicted in fig. 7 as an example on which a classical simulation can not be carried out.

More than 337,000 triangles exhibit a free edge whose roots are gaps between spline patches. Additionally, a total number of 2,324 triangles are oriented in the wrong direction, and there are innumerable intersections. Due to the immense amount of flaws, (manual) healing is not applicable. However, using the approach presented in [25], it is possible to run a simulation

without healing a single flaw, see fig. 8, for a result of a direct computation on the flawed model depicted in fig. 7.

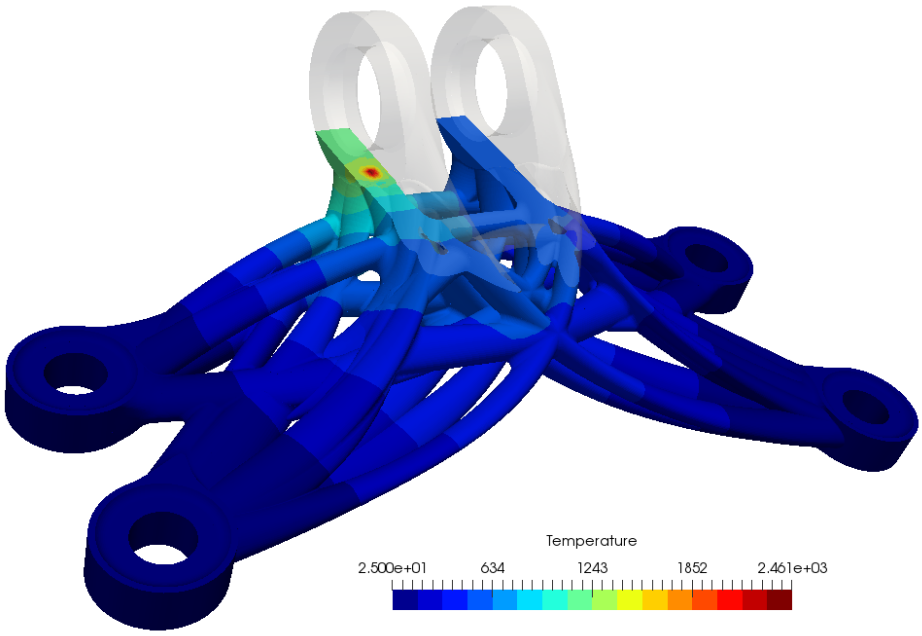


Figure 8: Temperature distribution in the bracket due to a point heat source.

Of course, the local accuracy of such computations is limited and proportional to the largest gap. However, those flaws might not even be situated in a structural engineer's region of interest – and the resulting overall accuracy might be good enough for a first design analysis. The point is that a computational analysis can actually be carried out at all. This is remarkable, especially because the model is actually invalid as no volume is correctly defined. However, it is only invalid up to the 'blindness' of the point membership test, which can be boiled down to a simple controllable parameter at least for gaps. Furthermore, the FCM can render the computational model to be absolutely insensitive to other topological flaws in the geometry, such as overlaps. A detailed discussion of how the Finite Cell Method enables a direct computational analysis of flawed geometries is given in the publication [26], provided in section 2.3.

All previously presented geometric models assumed a geometry to originate from a computer aided design i.e. a designer manually drew an analytic description of the object under investigation. This may at times be difficult, for example with regard to historic structures with complex shapes. In these cases, model acquisition may be carried out either by scanning the object under investigation or by photogrammetric methods. Both are able to provide oriented point clouds, i.e. a collection of points that represent discrete locations on the surface of the object. An outward-pointing vector is associated to each point of the object under investigation. This process is called *shape acquisition*. The most common path to computational analysis is then to derive a surface model from the point cloud using geometric segmentation and surface fitting methods. This process is called *surface reconstruction*. The resulting model is then stored using standardized geometric representation techniques such as STL, STEP, or IGES files. This surface description is then used to generate a boundary-conforming mesh.

Material properties as well as boundary conditions are then assigned, and a finite element analysis can be carried out.

However, point clouds already provide sufficient information to formulate a reliable point membership test. Therefore, a computational analysis is possible directly on the point cloud using the Finite Cell Method. This shortcut in the chain from image acquisition to computational analysis is depicted in fig. 9 and further discussed in section 2.4.

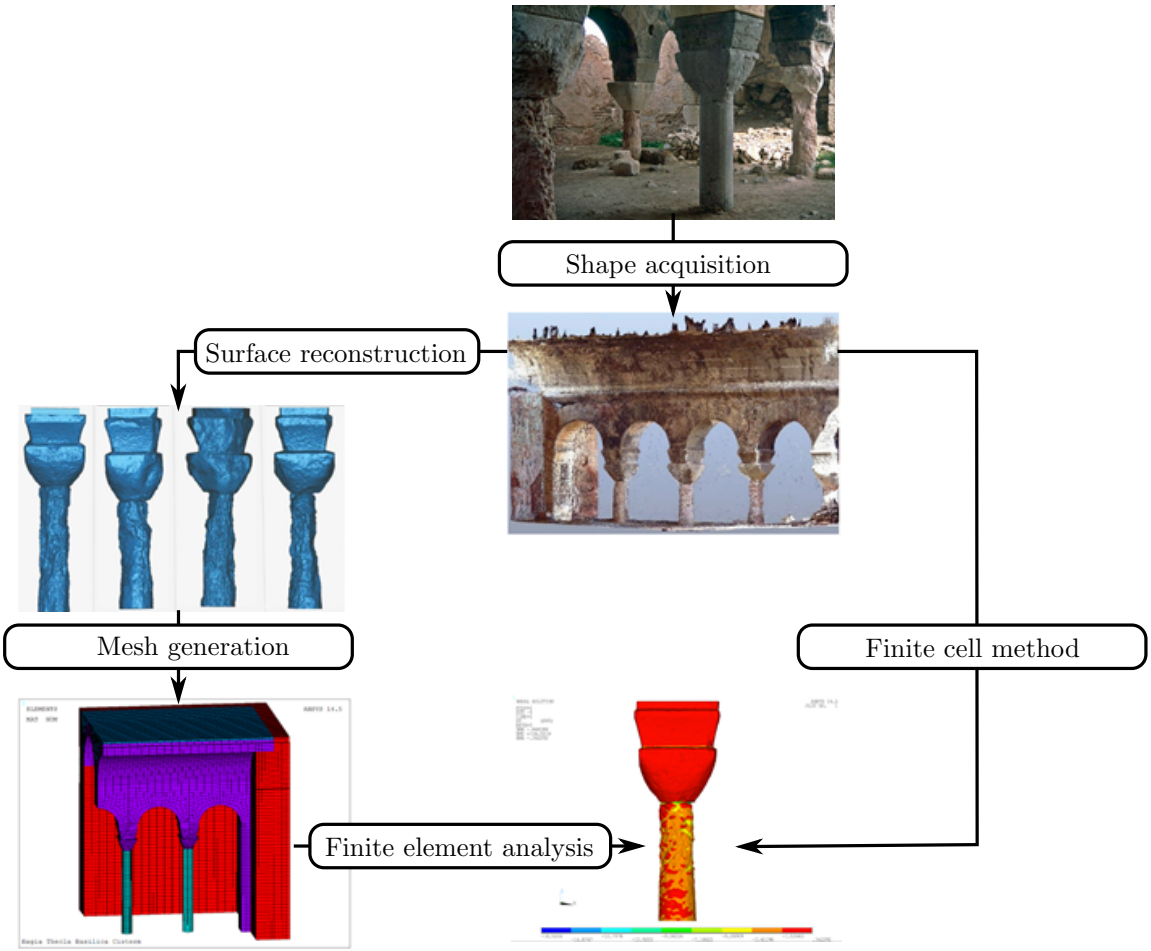


Figure 9: From point clouds to simulations: the classical path via surface reconstruction and mesh generation versus a direct computation on point clouds using the Finite Cell Method [27]

Another recent extension of the Finite Cell Method is the possibility of using different geometric models (CSG, B-rep, and Voxel Models) *simultaneously* in one computation. Consider fig. 10 where two screws are driven into a vertebra: This situation appears in an orthopedic procedure used to stabilize a spinal segment by fusing two vertebrae. The model of the screw is given by a boundary representation model in form of an STL-file, while the lower thoracic vertebra is defined by a CT-scan. On the computational mechanics side, each model is discretized by a non-conforming mesh. The boundary of the physical domain is only taken into account at the integration level. The FCM discretizations are glued at the intersection of the physical boundaries using a penalty approach. This work was published in [28]. This procedure has the interesting additional aspect that the otherwise difficult problem of computationally representing embedded interfaces, as addressed in section 3.4.1, is

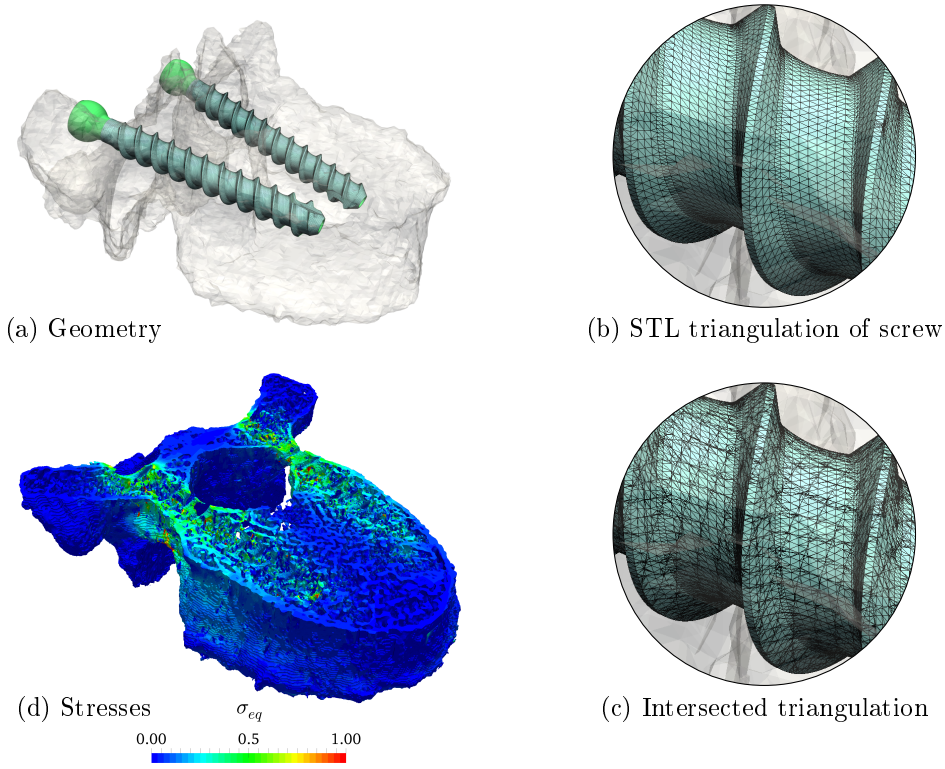


Figure 10: FCM on multiple geometric models

resolved accurately. While the underlying methodology is very general, it was first designed for mechanobiological applications. Therefore, it is presented in section 4.3 in form of a corresponding journal publication [28].

2.2 Constructive Solid Geometry and the Finite Cell Method

The extensions of the Finite Cell Method to constructive solid geometry are thoroughly laid out in the following publication.

Journal Publication

title:	From geometric design to numerical analysis: A direct approach using the Finite Cell Method on Constructive Solid Geometry
authors:	B. Wassermann, S. Kollmannsberger, T. Bog, E. Rank
published at:	*Computers & Mathematics with Applications*
publisher:	Elsevier
year:	2017
volume:	74
pages:	1703–1726
doi:	https://doi.org/10.1016/j.camwa.2017.01.027

From geometric design to numerical analysis: A direct approach using the Finite Cell Method on Constructive Solid Geometry

Benjamin Wassermann [*1], Stefan Kollmannsberger[†1], Tino Bog[‡1], and Ernst Rank[§1,2]

[1]Chair for Computation in Engineering, Technische Universität München, Arcisstr. 21, 80333 München, Germany
[2]Institute for Advanced Study, Technische Universität München, Lichtenbergstr. 2a, 85748 Garching, Germany

Abstract

During the last ten years, increasing efforts were made to improve and simplify the process from Computer Aided Design (CAD) modeling to a numerical simulation. It has been shown that the transition from one model to another, i.e. the meshing, is a bottle-neck. Several approaches have been developed to overcome this time-consuming step, e.g. Isogeometric Analysis (IGA), which applies the shape functions used for the geometry description (typically B-Splines and NURBS) directly to the numerical analysis. In contrast to IGA, which deals with boundary represented models (B-Rep), our approach focuses on parametric volumetric models such as Constructive Solid Geometries (CSG). These models have several advantages, as their geometry description is inherently watertight and they provide a description of the models interior. To be able to use the explicit mathematical description of these models, we employ the Finite Cell Method (FCM). Herein, the only necessary input is a reliable statement whether an (integration-) point lies inside or outside of the geometric model. This paper mainly discusses such point-in-membership tests on various geometric objects like sweeps and lofts, as well as several geometric operations such as filleting or chamfering. We demonstrate that, based on the information of the construction method of these objects, the point-in-membership-test can be carried out efficiently and robustly.

Keywords: CSG, Constructive Solid Geometry, Finite Cell Method, Spline Ray Casting, Point Membership Classification, Procedural Modeling, Sweep, Loft

DOI: 10.1016/j.camwa.2017.01.027

[*]benjamin.wassermann@tum.de, Corresponding Author
[†]stefan.kollmannsberger@tum.de
[‡]tino.bog@tum.de
[§]ernst.rank@tum.de

Preprint accepted by *Computers & Mathematics with Applications* February 16, 2017

Journal publication: https://doi.org/10.1016/j.camwa.2017.01.027

Contents

Journal publication: https://doi.org/10.1016/j.camwa.2017.01.027

1. Introduction

Computer aided engineering in general requires an iterative process to find an optimal design. This iterative process consists of a modelling phase followed by a numerical simulation and an analysis phase.
Modern CAD tools mainly use two different techniques to create 3D models. A classic method, which is still commonly used, is boundary representation (B-Rep)[1]. B-Rep describes a body implicitly as a topological model via its faces, edges, and nodes. Geometric information is then assigned to faces and edges, often using B-Spline-, or NURBS surfaces and curves. A more recent and natural approach is Procedural Modeling (PM), which is strongly related to Constructive Solid Geometry (CSG), but extends this concept by providing additional operations and primitives. Both CSG and PM describe a complex model as a combination of simple or complex primitives and Boolean operations (union, intersection, difference). Procedural modeling and B-Rep each have advantages and disadvantages, which are often complementary in such a way that, nowadays, many CAD systems use a hybrid representation combining B-Rep and PM [2]. In this context, the B-Rep model provides information necessary e.g. for visualization purposes. PM serves as an underlying model that can easily be used for parametric and feature-based design [3], for which a description of the construction history, the dependencies, and the constraints is mandatory. It is noteworthy that it is always possible to derive a B-Rep model from a PM model, but not the other way round. This is due to the loss of information in the conversion from PM to B-Rep. In addition, B-Rep cannot provide information about the structure of the interior of the model. However, this information can be crucial, for example in cases of heterogeneous materials or to describe additive manufacturing processes. Interestingly, fully three-dimensional-computational mechanical analyses mostly draw the geometrical information of the computational domain from B-Rep models which are then explicitly converted into a volumetric description by a meshing process. Moreover, in the finite element method (FEM), elements are required to conform with the physical boundaries of the model, which often requires a flawless B-Rep description. A practical consequence of these requirements is the often huge engineering effort to 'clean' a CAD model or to 'heal' a finite element mesh before a numerical analysis can start. At Sandia National Laboratories [4], an estimation of the relative time required for a representative design process showed that more than 80 % of the engineering effort is allotted to the transition from geometric models to simulation models that are suitable for analysis.

Various methodologies have been developed to overcome the difficulties involved in this transition process. The most prominent method in the Computational Mechanics Community is the recently introduced Isogeometric Analysis (IGA) as proposed by Hughes et al. [5]. IGA aims at bridging the gap between the CAD model and computational analysis by a closer mathematical interconnection between the two worlds. To this end, the same B-Splines and NURBS representations used to describe CAD are applied as both geometry and Ansatz functions in FEM. These functions offer several desirable properties such as the possibility of straightforward refinements in grid size and polynomial degree, as well as the possibility to control the continuity within a patch. Most importantly, they guarantee a precise description of the geometry, in contrast to classical FEM, where only an approximation can be obtained by meshing into tetrahedra or hexahedra. Furthermore, as B-Splines and NURBS are functions of higher order, they offer the potential to deliver high convergence rates if the underlying problem possesses smooth solutions. Concerning the modeling processes, IGA was first applied to B-Reps which consisted of several conforming two-dimensional B-Splines or NURBS patches. More complicated topologies are usually generated by trimming, which may lead to non-watertight geometric models. Remedies for this problem range from classic re-parametrization [6] to the use of T-Splines [7].

An alternative, designed to overcome the problems of B-Rep descriptions, are V-Reps, recently proposed by Gershon et al. [8]. They consist of trimmed trivariate NURBS patches which directly describe the volume under consideration. A related approach was presented by Zuo et al. [9], who proposed to treat CSG primitives separately as volumes using IGA and to trim and glue them by using the Mortar Method [10] at their intersection surface. However, apart from some special numerical pitfalls inherent to domain sewing techniques, this poses the additional difficulty that an explicit boundary representation needs to be set up *for all* inter-subdomain boundaries. Another related approach was presented much earlier by Natekar et al. [11] who proposed to combine spline-based element formulations with two-dimensional CSG model descriptions.

Journal publication: https://doi.org/10.1016/j.camwa.2017.01.027

However, this approach is also based on heavy use of explicit boundary representations as well as a decomposition into sub-domains. The same holds also for the design-through analysis procedure presented in [12], which uses a B-Rep description and relies on a 3D ray-casting test to describe the volume of the model.

This strong reliance on the explicit description of coupling interfaces – or, more generally, surface descriptions in the analysis process – poses a drawback to parametric modeling approaches: Even though a change of parameters or constraints hardly has any impact on the general structure of the CSG model itself, it often triggers a complete reconstruction of the entire corresponding B-Rep model. Together with the observation that a CSG or a procedural model is intrinsically watertight and directly provides information about the interior, we conclude that a desirable simulation technique would have to use the explicit description of volumes by CSG as often as possible, and its B-Rep representation as little as possible.

To this end, we propose a combination of CSG and the Finite Cell Method for volume orientated modeling and numerical analysis. We denote this approach as a direct modeling-to-analysis method as it allows, like IGA, a very close interaction of the (geometric) design process and the (numerical) analysis, where an engineer can immediately investigate consequences of a variation of the geometric design on the mechanical behavior of a structural object.

The Finite Cell Method (FCM) [13], which represents the core of this approach, is a high-order fictitious domain method that embeds an arbitrary complex geometry into an extended domain which can easily be meshed by a Cartesian grid. The complexity of the geometry is handled only on the level of integration of element matrices and load vectors. This makes the method very flexible, because the only information the FCM needs from the CAD model is a reliable and robust point-in-membership test, i.e. whether an integration point lies inside or outside of the physical model. This point-in-membership test is directly provided by the CSG model description. The interplay between CSG and FCM was already investigated for simple primitives, and it proved to be an accurate and efficient method to analyze trimmed NURBS patch structures [14]. The goal of the present paper is to extend the combination of the FCM and the CSG to more complex geometric models as well as to solid construction processes of industrial relevance.

This paper is organized as follows: In section 2, a short overview on geometric representations and the Finite Cell Method is given. In section 3, the relevant methods for the combination of CSG and FCM are presented. Section 4 provides examples showing the relevance and potential for practical applications before conclusions are drawn in section 5.

Journal publication: https://doi.org/10.1016/j.camwa.2017.01.027

2. From Geometric Design to Numerical Analysis

2.1. Geometric modelling

In the field of Computer Aided Design (CAD), several different schemes are available to model 3D geometric objects. Nowadays, 3D CAD systems are usually based on either (i) boundary representation or (ii) procedural modeling with solid primitives. Next, the two schemes will be outlined – followed by a short section about the conversion of one into another.

2.1.1. Boundary Representation

Classically, objects are defined by a Boundary Representation (B-Rep), where only the objects' surfaces with their corresponding edges and nodes are stored (see fig. 1) [15]. This is motivated by the requirements for visualization, in the scope of which 3D objects are displayed via their surfaces.

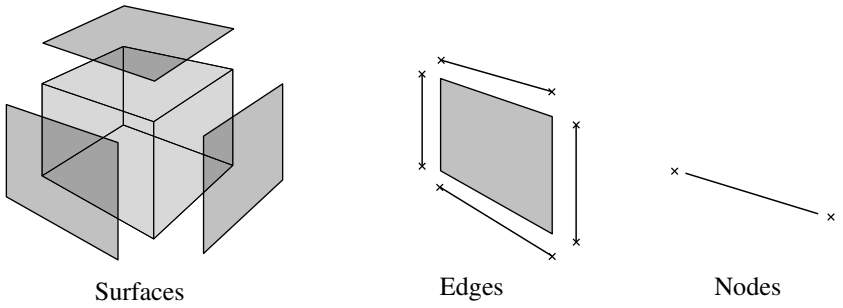


Figure 1: Structure of B-Rep.

Although B-Rep has several advantages, for example the direct access to surfaces, it has also some disadvantages, especially with respect to a subsequent numerical simulation. B-Rep models are not necessarily watertight, which means that a point-in-membership test on these potentially corrupted solids may not be sufficient to clearly distinguish whether a point lies outside or inside the domain.

Another disadvantage are defective topological descriptions, such as multiple nodes, or edges. Although such errors do not disturb the visualization, they may render numerical simulations difficult or even impossible. These 'dirty geometries' are among to the major reasons for the considerable effort that often goes with cleaning up a geometric model. This preparation work is necessary to be able to mesh a model into a consistent finite element model.

2.1.2. CSG and procedural modelling

Alternatively, a 3D object can be described as a procedural model that is strongly related to Constructive Solid Geometry (CSG) [16]. In CSG, a 3D object is created from a set of primitives, such as cubes, cylinders, cones, spheres etc. These primitives are combined by the three basic boolean operations: union, intersection, and difference. The resulting CSG object is stored implicitly in a CSG tree (see fig. 2).

Journal publication: https://doi.org/10.1016/j.camwa.2017.01.027

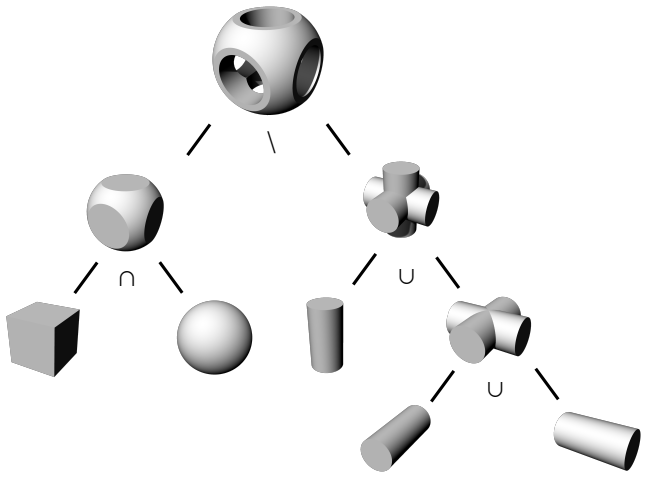


Figure 2: CSG Tree using the three boolean operations: union ∪, intersection ∩, difference \ on primitives.

In contrast to CSG, a procedural model stores the construction steps in chronological order as a construction history. These two different storage schemes, the CSG tree and the construction history, can be converted one to another. Procedural modeling also comes along with a richer set of operations and primitives, in the following referred to as extended operations and extended primitives. Extended operations include chamfer, fillet, drilling a hole, and draft. A closer look reveals that they are in fact just a sequence of the original three boolean operations – union, difference, and intersection – which are summarized for convenience (see fig. 3).

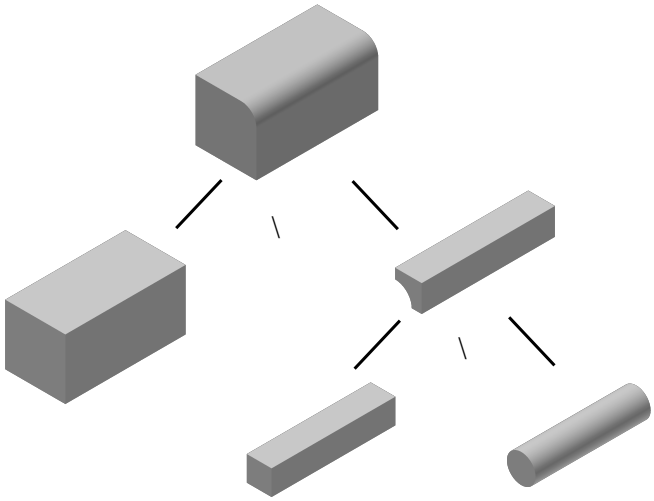


Figure 3: Extended operations can be expressed by the classical boolean operations: union, intersection, difference. The example shows filleting an edge.

However, extended primitives such as extrusions, sweeps, lofts, and solids of revolution can be regarded as a true extension to the CSG primitive set (see fig. 4).

Journal publication: https://doi.org/10.1016/j.camwa.2017.01.027

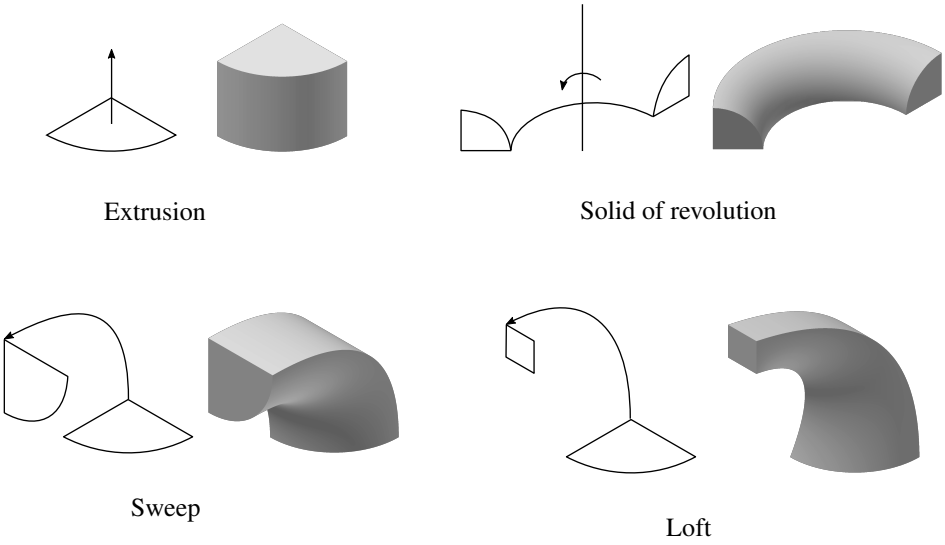


Figure 4: Extended primitives: extrusion, solid of revolution, sweep, loft.

The construction processes of these primitives are all strongly related. On an arbitrary plane, a 2-dimensional closed contour line is drawn. This is extruded along a sweep path. Depending on the shape of this path, either an extrusion, a solid of revolution, or a sweep is obtained. Only the construction of a loft differs slightly. Here, the initial sketch is not just extruded, but is blended along the path into another (final) sketch.

Regarding the aspect of numerical simulation, one big advantage of the procedural model over the B-Rep model is that it is inherently watertight. All primitives are explicitly described by their volume and always form a valid closed 3D object. CSG operations combine these valid primitives, and their combination again results in a valid model.

2.1.3. Conversion between explicit and implicit models

We would like to stress the fact that a conversion from a B-Rep model to a CSG model is usually not possible, as B-Rep models carry less information. Thus, it is desirable to use an explicit volume description such as CSG directly for a simulation.

The drawback of the CSG description is that it only provides indirect access to the models' surfaces, which might be needed e.g. for visualization. Fortunately, it is always possible to derive an approximate surface description from the CSG, e.g. by the marching cubes algorithm [17]. This algorithm only needs the information whether a point is inside or outside of the model at any point, which is readily available via the CSG construction tree. Another (practical) possibility is to gather the B-Rep information directly from the CAD software if needed. This is often possible because many CAD systems maintain a B-Rep model concurrently to a CSG model and provide direct access to its surface. These derived surface models, even if they are not perfectly watertight, are a sufficient basis to impose boundary conditions in an analysis by the Finite Cell Method, which will be described in the next section.

2.2. Finite Cell Method

The finite cell method (FCM) is a fictitious domain approach using high order finite elements. It relies on an explicit description of the volume of the physical domain and is able to deliver high accuracy [18].

Journal publication: https://doi.org/10.1016/j.camwa.2017.01.027

2.2.1. Classical finite elements

Consider a linear-elastic problem on a physical domain Ω_{phy} with the boundary $\delta\Omega$ divided into Dirichlet and Neumann boundary parts Γ_{D} and Γ_{N}. By applying the principle of virtual work, the weak form of an elliptic partial differential equation reads [19]

$$\mathcal{B}(\mathbf{u},\mathbf{v}) = \mathcal{F}(\mathbf{v}) \qquad \begin{matrix}\forall \mathbf{u} \in \mathcal{S}(\Omega_{\text{phy}}) \\ \forall \mathbf{v} \in \mathcal{V}(\Omega_{\text{phy}})\end{matrix} \tag{1}$$

with

$$\mathcal{B}(\mathbf{u},\mathbf{v}) = \int_{\Omega_{\text{phy}}} \nabla\mathbf{v} : \mathbb{C} : \nabla\mathbf{u}\, \mathrm{d}\Omega_{\text{phy}} \tag{2}$$

$$\mathcal{F}(\mathbf{v}) = \int_{\Omega_{\text{phy}}} \mathbf{b}\cdot\mathbf{v}\, \mathrm{d}\Omega_{\text{phy}} + \int_{\Gamma_{\text{N}}} \hat{\mathbf{t}}\cdot\mathbf{v}\, \mathrm{d}\Gamma_{\text{N}} \tag{3}$$

where $\mathbf{u}$ is the displacement, $\mathbf{v}$ the test function, and $\mathbb{C}$ the elasticity tensor. $\mathbf{b}$ and $\hat{\mathbf{t}}$ denote the body load and the prescribed boundary traction applied on the Neumann boundary, respectively. $\mathcal{S}(\Omega_{\text{phy}})$ is the trial function space, which is constructed such that $\mathbf{u}$ satisfies the prescribed Dirichlet boundary conditions $\hat{\mathbf{u}}$

$$\mathcal{S}(\Omega_{phy}) = \left\{\mathbf{u} \;\middle|\; \mathbf{u} \in H^1(\Omega_{\text{phy}}), \mathbf{u} = \hat{\mathbf{u}} \;\; \forall\, \mathbf{x} \in \Gamma_{\text{D}}\right\}, \tag{4}$$

whereas $\mathcal{V}(\Omega_{\text{phy}})$ denotes the space of all admissible test functions that satisfy homogeneous Dirichlet boundary conditions

$$\mathcal{V}(\Omega_{\text{phy}}) = \left\{\mathbf{v} \;\middle|\; \mathbf{v} \in H^1(\Omega_{phy}), \mathbf{v} = 0 \;\; \forall\, \mathbf{x} \in \Gamma_{\text{D}}\right\}. \tag{5}$$

H^1 denotes the Sobolev space [20] of first order. Both $\mathbf{u}$ and $\mathbf{v}$ are discretized to yield an approximate solution using a linear combination of Ansatz functions $\{N_1, N_2, ..., N_n\}$

$$\mathbf{u}_h = \tilde{\mathbf{N}}\mathbf{u}, \tag{6}$$

where $\tilde{u}_i$ is the degree of freedom of the related Ansatz function N_i. Following the Bubnov - Galerkin approach [21], the test functions $\boldsymbol{\nu}$ are represented by the same basis as the trial functions $\mathbf{v} \in \mathcal{S}(\Omega_{\text{phy}})$. This approach leads to the following system of linear equations:

$$\tilde{\mathbf{K}}\mathbf{u} = \mathbf{f} \tag{7}$$

where $\mathbf{K}$ is the stiffness matrix and $\mathbf{f}$ the load vector.

2.2.2. Concept of FCM

The original idea of the finite cell method is to extend the physical domain Ω_{phy} by a fictitious domain Ω_{fict} such that the resulting domain $\Omega_{\cup}$ has a simple shape, which can be meshed easily (see fig. 5)[18, 22].

The weak formulation is modified by extending integrals over the domain $\Omega_{\cup}$. Additionally, the virtual work terms are multiplied by a scalar field $\alpha(x)$:

$$\mathcal{B}_e(\mathbf{u},\mathbf{v}) = \int_{\Omega_{\cup}} \nabla\mathbf{v} : \alpha\mathbb{C} : \nabla\mathbf{u}\, d\Omega_{\cup} \tag{8}$$

$$\mathcal{F}_e(\mathbf{v}) = \int_{\Omega_{\cup}} \alpha\mathbf{b}\cdot\mathbf{v}\, d\Omega_{\cup} + \int_{\Gamma_N} \hat{\mathbf{t}}\cdot\mathbf{v}\, \Gamma_N \tag{9}$$

Journal publication: https://doi.org/10.1016/j.camwa.2017.01.027

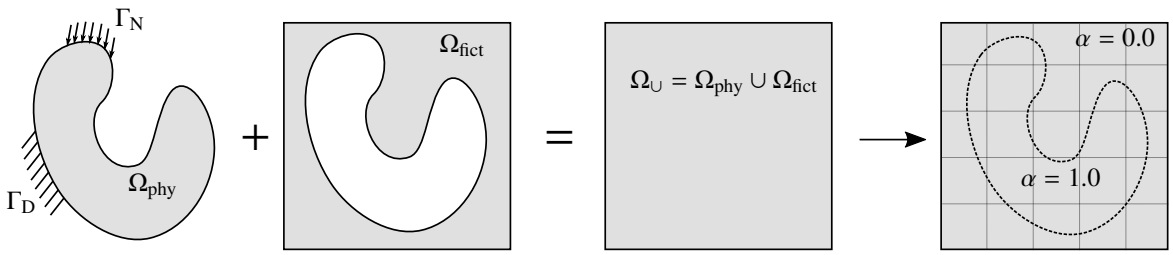


Figure 5: Concept of Finite Cell Method.

with α defined as:

$$\alpha = \begin{cases} 1 & \forall \mathbf{x} \in \Omega_{phy} \\ 10^{-q} & \forall \mathbf{x} \in \Omega_{fict} \end{cases}, \tag{10}$$

where, ideally $q \to \infty$. In practical applications, it is usually sufficient to choose $q = 8$ *to* 10. In essence, the discontinuous indicator function α now represents the geometric description of the domain. The convergence of this scheme is mathematically proven in [23] where it is also shown that the influence of a non-infinite q is proportional to a (controllable) modeling error.

After discretizing the extended domain $\Omega_\cup$ into a Cartesian grid, high-order finite elements can be used for the computation of the displacement field. Several different Ansatz functions for high-order elements are available, such as integrated Legendre polynomials [22, 24], B-Splines [25, 26], and Lagrange polynomials [27, 28].

The discontinuity of α necessitates an adaptive integration of the element matrices and load vectors, see e.g. [29, 30] for a recent overview of possible schemes. The simplest (although not most efficient) choice is a composed integration by means of an octree in 3D or a quadtree in 2D (see fig. 6) which is used in all examples presented in section 4.

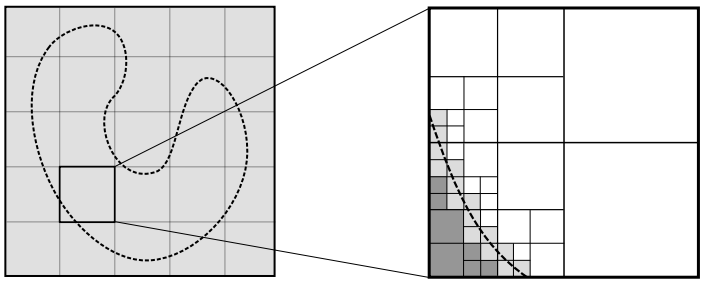


Figure 6: Quadtree partitioning of a 2D cell with a partitioning depth of 4. Coloring of quadtree leaves: white – completely inside, light gray – cut, dark gray – completely outside)

2.2.3. Boundary conditions

In FCM the boundaries of the physical domain Ω_{phy} typically do not coincide with the boundaries of the cells in the extended domain $\Omega_\cup$. In that case boundary conditions are enforced weakly.

Journal publication: https://doi.org/10.1016/j.camwa.2017.01.027

Homogeneous Neumann boundary conditions need not to be treated in a special manner since they are naturally included in the weak formulation (see equation (8)). Inhomogeneous Neumann conditions can be handled by integrating the prescribed traction forces $\hat{\mathbf{t}}$ along the boundary Γ_N (see equation (3)).
Dirichlet boundary conditions, i.e. applying a prescribed displacement $\mathbf{u} = \hat{\mathbf{u}} \quad \forall \mathbf{x} \in \Gamma_D$, must be applied in a weak sense using, e.g., Lagrangian multipliers, the penalty method, or Nitsche's method, see e.g. [31] [32].

In short, an explicit description of the boundary must only be available where forces or displacements are imposed (see section 2.1.3). The geometric and topological requirements of these surfaces are much lower than for mesh-generation as they only need to fulfill conditions for a sufficiently accurate numerical integration. Therefore, these integration meshes neither have to be conforming or watertight, as it would be necessary for the basis for a finite element computation.

3. FCM and CSG

3.1. Point-in-membership test

The geometric description of the physical domain is provided by the function α which is explicitly given by a point-in-membership test, i.e. if a point lies inside the physical domain Ω_{phy} or inside the fictitious domain Ω_{fict}. This test can be carried out on a B-Rep model by ray casting. To this end, a ray is sent out from the integration point into an arbitrary direction, and all intersections with the surfaces are counted. If the number of intersections is odd, then the starting point lies inside Ω_{phy}, otherwise it lies in Ω_{fict} (see fig. 7). This simple test, however, can fail for non-water-tight models. Moreover, this approach can become quite expensive for highly resolved models or models with a complex surface. Even though there are methods available to reduce the amount of ray castings. [1]

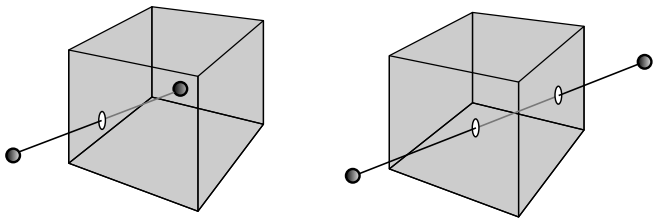


Figure 7: Ray casting on a B-Rep model.

In contrast, a point-in-membership test can be performed much faster on a CSG tree of classical primitives. In our implementation the CSG tree is a full binary-tree, i.e. each node has either exactly two or zero child nodes. There are two different types of nodes: (a) primitives which are always leaf nodes, and (b) bifurcation nodes which combine their two children with one of the three boolean operations: intersection, union, difference (see fig. 8). It is also possible to add elements, carrying only one child like unary operations such as negations, or transformations to the CSG tree. Whereas negations have hardly any application in geometric modeling of solid mechanics (in contrast to exterior problems, such as fluid mechanics), transformations are used intensively. To this end, simple transformations, such as translation, scaling and rotation can be handled directly on the tree nodes, whereas more complex operations, like mirroring, would be preferably represented by a node, containing the operation and only the respective child to which this operation should be applied.
For a point-in-membership test, first the root node representing the final construction is queried. In the un-

[1] For example hierarchical bounding boxes, which divide the surface triangles into small chunks. Only if the ray intersects a bounding box the query is forwarded to the respective next level of bounding boxes and finally to a the relevant chunk(s) of triangles.

Journal publication: https://doi.org/10.1016/j.camwa.2017.01.027

likely case that the root node is also a leaf node and, hence, a primitive, this test is carried out directly. In all other cases, the root element is a bifurcation node. Then, the request is forwarded recursively to its children until it reaches a primitive, i.e. a leaf node. The pairing leaf node is tested as well. For simple primitives (treated in section 3.2), this test can be carried out very fast, as an analytical solution is available. Both results are then combined with the logical operation defined by the parent bifurcation node.
The algorithm can be sped up considerably by the following considerations: Due to the recursive property at a bifurcation node, the entire branch of the first child is evaluated before the second child is queried. This can be used with the knowledge that, in case of an intersection or a difference, it is often not necessary to test the entire tree. Considering a difference

$$A \backslash B := \mathbf{P} \in A \wedge \mathbf{P} \notin B \tag{11}$$

i.e. if point **P** is not in body A, body B needs not to be queried. Hence, entire branches of the CSG tree can be omitted during the query. The same holds for the intersection. Therefore, it is useful to perform an intersection test with the bounding box of computationally expensive branches first. This test can be introduced on each level in the CSG tree.

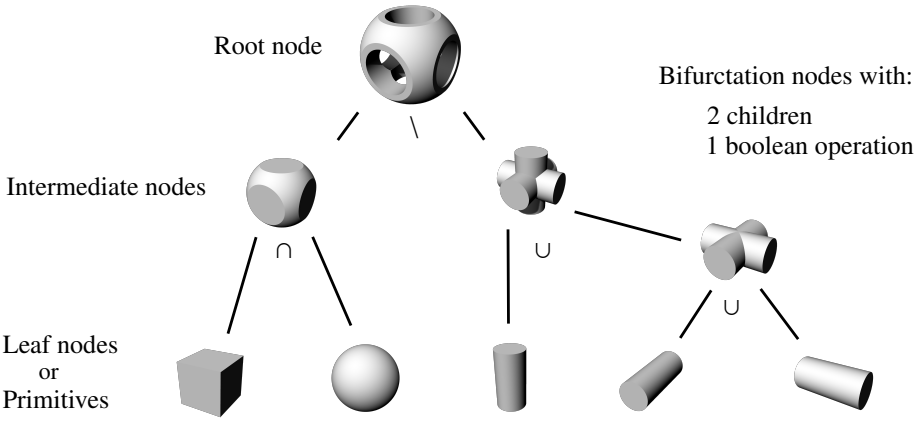


Figure 8: CSG-tree consisting of (a) primitives at leaf nodes and (b) bifurcation nodes at an intermediate or root node with two children and a boolean operation.

3.2. Point-in-membership test on simple primitives

For classical primitives, a simple analytical description is available. Hence, a point-in-membership test can be carried out very efficiently. Consider a primitive $\mathbf{B}_i$ which is created axis-aligned on the $x-y$ plane, and assume that we define each primitive as a closed body, i.e. the boundary is included in the body. The test whether a point of interest $\mathbf{P} = \{x, y, z\}$ is inside a primitive reads as follows for a:

- **Sphere** with center point $\mathbf{C}_{\text{Shpere}}$ and radius r_0

$$\mathbf{P} \in \mathbf{B}_{\text{Sphere}} \quad \text{iff} \quad \|\overline{\mathbf{PC}}_{\text{Sphere}}\|_2 \leq r_0, \tag{12}$$

- for a **Cuboid** defined by two corner points lying on its diagonal $P_{\text{start}} = [x_s, y_s, z_s]$ and $P_{\text{end}} = [x_e, y_e, z_e]$

$$\mathbf{P} \in \mathbf{B}_{\text{Cuboid}} \quad \text{iff} \quad x \in [x_s, x_e] \wedge y \in [y_s, y_e] \wedge z \in [z_s, z_e], \tag{13}$$

- for a **Cylinder** defined by its center point $\mathbf{C}_{\text{Cylinder}} = \{x_c, y_c, z_c \equiv 0\}$, radius r_0, and height h_0

$$\mathbf{P} \in \mathbf{B}_{\text{Cylinder}} \quad \text{iff} \quad \|\tilde{\overline{\mathbf{PC}}}_{\text{Cylinder}}\|_2 \leq r_0 \wedge z \in [z_c, z_c + h_0] \tag{14}$$

Journal publication: https://doi.org/10.1016/j.camwa.2017.01.027

where point $\tilde{\mathbf{P}} = \{x, y, 0\}$ is the projection of point $\mathbf{P}$ onto the $x - y$ plane,

- a **Cone Frustum** with the same set up as for the cylinder whose bottom and tip circles are concentric with radii r_0 and r_1

$$\mathbf{P} \in \mathbf{B}_{\text{Cone}} \quad \text{iff} \quad \|\overline{\tilde{\mathbf{P}}\mathbf{C}}_{\text{Cone}}\|_2 \leq r(z) \wedge z \in [z_c, z_c + h_0] \tag{15}$$

with

$$r(z) = \frac{r_1 - r_0}{h_0} z + r_0. \tag{16}$$

If the radius r_1 is chosen to be zero, a complete cone is obtained.

- and a **Pyramid Frustum** with a corresponding set up to that of the cone frustum. The rectangular bounding box at the bottom $[x_{s0}, x_{e0}]$, $[y_{s0}, y_{e0}]$ and on the top $[x_{s1}, x_{e1}]$, $[y_{s1}, y_{e1}]$ have the same center point.

$$\mathbf{P} \in \mathbf{B}_{\text{Pyramid}} \quad \text{iff} \quad x \in [x_s(z), x_e(z)] \wedge y \in [y_s(z), y_e(z)] \wedge z \in [z_c, z_c + h_0] \tag{17}$$

with

$$x_s(z) = \frac{x_{s1} - x_{s0}}{h_0} z + x_{s0} \tag{18}$$

and $x_e(z), y_s(z), y_e(z)$ correspondingly. If $x_{s1} = x_{e1}$ and $y_{s1} = y_{e1}$ the pyramid frustum becomes a complete pyramid.

There are also fast analytical solutions for other primitives - such as wedges, four-sided pyramids, or tori – available.
In general these primitives are not constructed axis-aligned to the $x - y$ plane. At a suitable position, a local orthonormal coordinate system $\mathbf{A}(\mathbf{A}_1, \mathbf{A}_2, \mathbf{A}_3)$ is thus constructed, where $\mathbf{A}_i$ denotes the respective base vectors. The oriented primitive can be constructed on the local $\mathbf{A}_1$-$\mathbf{A}_2$ plane,. To perform a point-in-membership test, the point of interest $\mathbf{P}$ needs to be mapped from the Cartesian coordinate system $\mathbf{E}(\mathbf{E}_1, \mathbf{E}_2, \mathbf{E}_3)$ to the local base $\mathbf{A}$.

$$\tilde{\mathbf{P}} = \mathbf{T}\,\mathbf{P} + \mathbf{v} \tag{19}$$

with $\mathbf{v}$ the translation vector between the origins of the Cartesian $\mathbf{E}$ and local basis system $\mathbf{A}$

$$\mathbf{v} = \mathbf{O}_A - \mathbf{O}_E = \mathbf{O}_A \tag{20}$$

and the transformation matrix

$$\mathbf{T} = \begin{bmatrix} A_{1x} & A_{2x} & A_{3x} \\ A_{1y} & A_{2y} & A_{3y} \\ A_{1z} & A_{2z} & A_{3z} \end{bmatrix} \quad . \tag{21}$$

3.3. Point-in-membership test on extended primitives

Point-in-membership tests are more complex, if the primitives are generated by sweeps or lofts, where no analytical tests are available in general. Nevertheless, it is possible to perform a fast, reliable test on these geometries as well. The basic idea is to reduce the dimension of the problem, taking advantage of the fact that these extended primitives are constructed by moving 2D sketches along a curve.We remark that these sketches are planar which corresponds to the options available in common CSG modeling tools such as Autodesk® Inventor® and Siemens NX®. The fallback to two dimensional point in membership tests does not compromise robustness because it is much easier to construct such tests in two- than in three dimensions. In the

Journal publication: https://doi.org/10.1016/j.camwa.2017.01.027

subsequent sections we present a fast and robust point in membership test based on ray casting. Other robust alternatives such as a classic winding number test [33] are possible as well.

3.3.1. Coordinate systems

In the following two local basis systems are introduced. First, the local basis $\mathbf{A}(\xi)$, which belongs to the curve, e.g. the Frenet base, and changes according to the (sweep or loft) path variable ξ. Second, the local basis $\mathbf{B}(\xi)$ of the sketch. This basis $\mathbf{B}(\xi)$ and so the sketch change accordingly to the curve basis system $\mathbf{A}(\xi)$ and thus depends also on ξ.

$$\mathbf{A}(\xi)\Big(\mathbf{A}_1(\xi), \quad \mathbf{A}_2(\xi), \quad \mathbf{A}_3(\xi)\Big) \tag{22a}$$

$$\mathbf{B}(\xi)\Big(\mathbf{B}_1(\xi), \quad \mathbf{B}_2(\xi), \quad \mathbf{B}_3(\xi)\Big) \tag{22b}$$

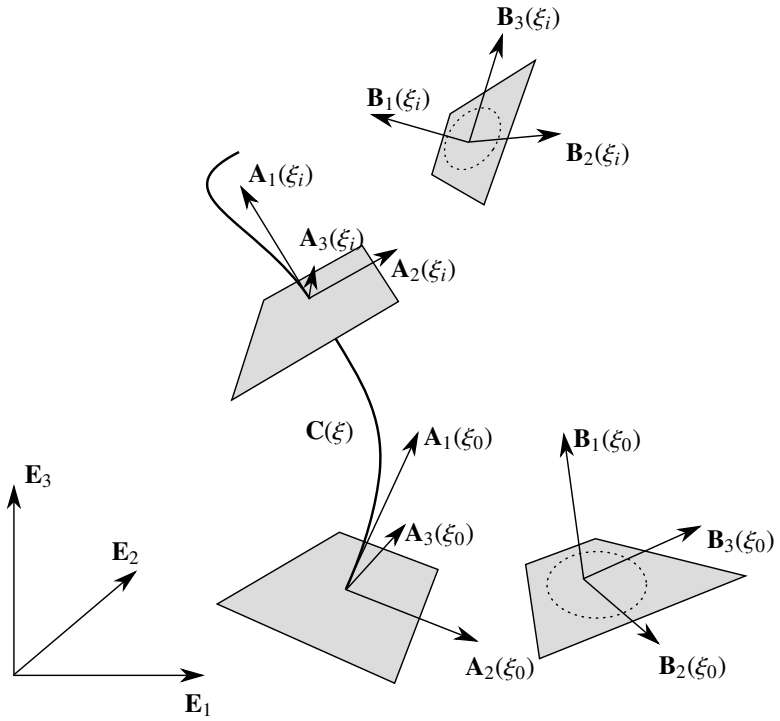


Figure 9: Coordinate systems: (a) Cartesian basis $\mathbf{E}$. (b) Local coordinate system $\mathbf{A}(\xi)$ of the sweep path $\mathbf{C}(\xi)$ (c) Local coordinate system of the sketch $\mathbf{B}(\xi)$.

3.3.2. Sweeps

As depicted in fig. 4 most extended primitives can be ascribed to sweeps. Thereby a 2-dimensional sketch is 'swept' along a 3-dimensional path, forming a 3D object. $\mathbf{C}(\xi)$ denotes the point on the sweep path $\mathbf{C}$ at the local path coordinate ξ. The following steps describe a point-in-membership test for a point of interest $\mathbf{P}$ and a sweep for the special case that (i) the local basis $\mathbf{A}(\xi)$ equals the basis of the sketch $\mathbf{B}(\xi)$ and (ii) the local basis system $\mathbf{A}(\xi)$ follows the tangent of the path with a suitable description, e.g. the Frenet base. For the general case, see appendix A.1.

Journal publication: https://doi.org/10.1016/j.camwa.2017.01.027

- The closest point $\mathbf{C}(\xi_{cp})$ on the sweep path with respect to the corresponding point of interest $\mathbf{P}$, is computed either analytically, or, if this is not possible, using Newton's method to find a root of the function

$$f(\xi) = \dot{\mathbf{C}}(\xi) \cdot (\mathbf{P} - \mathbf{C}(\xi)) \stackrel{!}{=} 0. \tag{23}$$

$f(\xi)$ is the dot product between the tangent vector $\dot{\mathbf{C}}(\xi)$ and the vector pointing from a curve point $\mathbf{C}(\xi)$ to the point of interest $\mathbf{P}$. Provided that a suitable starting value is available Newton's method then iteratively delivers the corresponding coordinates ξ_{cp}^{j+1} of ever closer points $\mathbf{C}(\xi_{cp}^{j+1})$:

$$\xi_{cp}^{j+1} = \xi_{cp}^{j} - \frac{f(\xi_{cp}^{j})}{\dot{f}(\xi_{cp}^{j})} = \xi_{cp}^{j} - \frac{(\dot{\mathbf{C}}(\xi_{cp}^{j}) \cdot (\mathbf{P} - \mathbf{C}(\xi_{cp}^{j}))}{\ddot{\mathbf{C}}(\xi_{cp}^{j}) \cdot (\mathbf{P} - \mathbf{C}(\xi_{cp}^{j})) + |\dot{\mathbf{C}}(\xi_{cp}^{j})|^2}, \tag{24}$$

where $\dot{\mathbf{C}}$ and $\ddot{\mathbf{C}}$ denote the first and second derivative of the sweep path $\mathbf{C}$. The minimal requirement for continuity of the sweep path is C^1.
Remarks:

(i) In our implementation the initial values are found by evaluating the distance to the points of an approximation polygon.

(ii) It is possible that eq. (23) has multiple solutions. Among the corresponding points on the curve $\mathbf{C}(\xi_i)$ the one with the smallest distance to the point $\mathbf{P}$ has to be chosen.

(iii) There is the unlikely possibility that eq. (23) delivers multiple solutions $\mathbf{C}(\xi_i)$ with all the same distance to the point $\mathbf{P}$. For sweeps it does not matter which of theses points is selected to be $\mathbf{C}(\xi_{cp})$. This, however, does not hold for lofts (for a more detailed explanation refer to fig. 10).

(iv) Non-linear cases are possible, where C^2 can not be provided, i.e. the curvature is not continuous, or worse cannot even be evaluated (e.g. knot-multiplicity for splines). For that reason we suggest to compute the first derivative $\dot{f}$ (see eq. (24)) by finite differences.

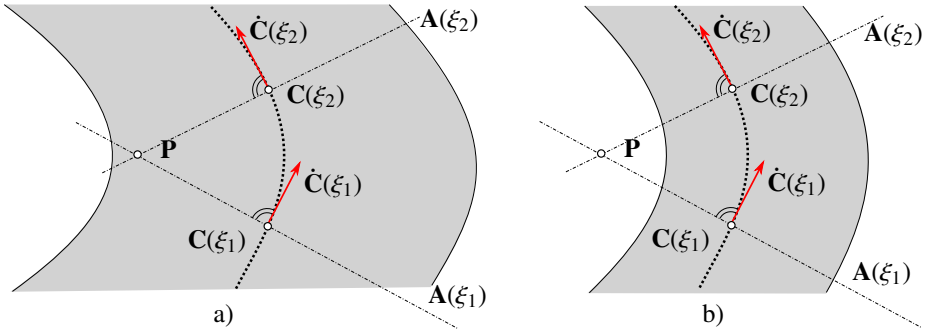


Figure 10: Multiple solutions for eq. (23) $\mathbf{C}(\xi_i)$ with the same distance to $\mathbf{P}$. For sweeps the sketch on plane $\mathbf{A}(\xi_1)$ coincides with the sketch on plane $\mathbf{A}(\xi_2)$. Hence, it does not matter on which plane the point-in-membership test is carried out. However, for lofts (see section 3.3.3) the intermediate sketches are interpolated according to the arc length. In this case, the sketches at $\mathbf{A}(\xi_1)$ and $\mathbf{A}(\xi_2)$ will be different. Consequently all solutions for eq. (23) must be evaluated. The point is defined to lie inside if one sketch delivers this result. However, these cases occur only in the unusual cases, where the thickness of the body is large compared to the curvature (see case a)). More likely is case b) where all ambiguous solutions for eq. (23) lie outside.

- The closest point $\mathbf{C}(\xi_{cp})$ forms the origin of the newly created local coordinate system $\mathbf{A}(\xi_{cp})$. To this end, the tangent vector of the curve is evaluated at $\mathbf{C}(\xi_{cp})$ and provides the z-axis of the local system $\mathbf{A}(\xi_{cp})$ using e.g. the Frenet base. Other alternatives are possible and presented in appendix A.1.

Journal publication: https://doi.org/10.1016/j.camwa.2017.01.027

- The point of interest **P** is mapped into the local coordinate system $\mathbf{A}(\xi_{cp})$ to get $\tilde{\mathbf{P}}$ where $\tilde{P}_z = 0$ (see eq. (19)).
- For the case that the local system $\mathbf{A}(\xi)$ coincides with $\mathbf{B}(\xi)$, a point-in-membership test can be performed with the mapped point $\tilde{\mathbf{P}}$ and the sketch contour line as the boundary. To this end, 2D ray casting is used.

  Remark: A considerable speedup, especially for spline contour lines, can be achieved if the sketch is modeled as quadtree of certain accuracy (similar to fig. 6) for each extended primitive. This can be carried out once at the beginning of the analysis phase.

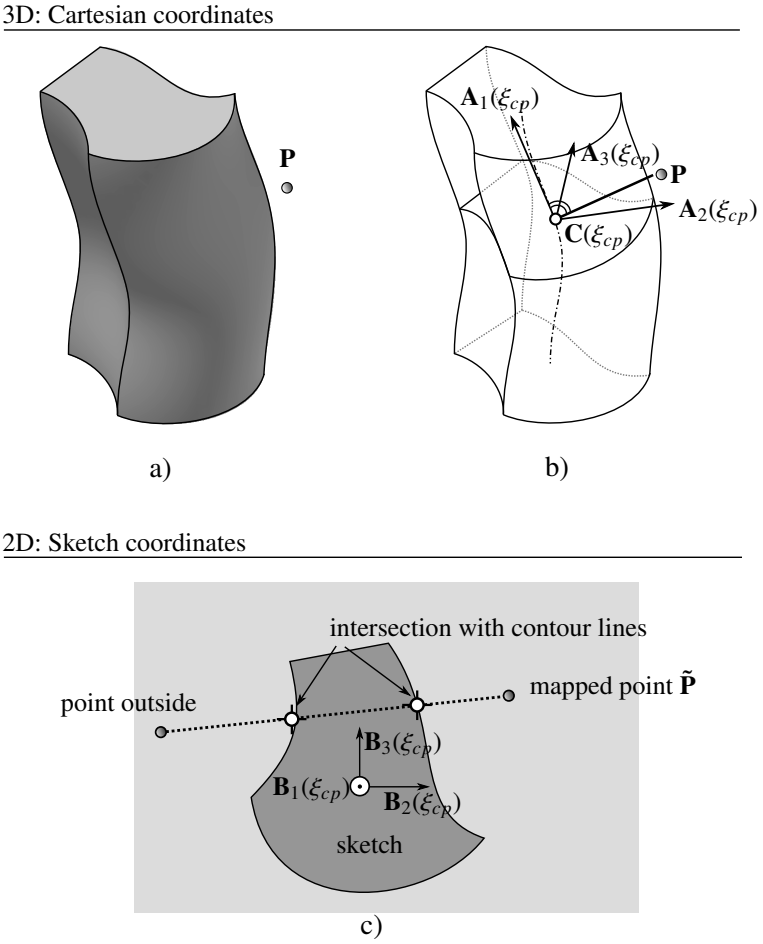


Figure 11: Point-in-membership test of a sweep whose local basis **A** coincides with the sketch basis system **B**: (a) Sweep with point of interest. (b) Form local basis **A** at closest point $\mathbf{C}(\xi_{cp})$ and (c) perform a point-in-membership test on the 2D sketch plane – here, by ray casting with the contour line of the sketch.

3.3.3. Lofts

Lofts can be treated very similarly to sweeps. Again, for simplicity, the loft path is considered to be orthogonal to the starting S_0 and ending S_{end} sketch. For a loft, in contrast to sweeps, the 2D point-in-membership test

Journal publication: https://doi.org/10.1016/j.camwa.2017.01.027

must be performed on both the starting as well as the ending sketch. Additionally, the smallest distances to both contour lines d_0 and d_{end} are calculated. If the mapped point $\tilde{\mathbf{P}}$ lies inside one sketch and outside the other, the distance to the intermediate contour line d_{cp} is interpolated according to the arclength l_{cp} of the closest point on the loft path (see fig. 12) and the overall length of the loft path l_{all}. In our implementation, linear interpolation is used. The distance to the point outside is set negative and, thereby, according to the sign of the interpolated distance d_{cp}, the point is outside for negative and inside for positive values. For the linear interpolation, the point-in-membership test reads

$$\mathbf{P} \in B_{\text{Loft}} \quad \text{iff} \qquad d_i \geq 0 \tag{25}$$

with

$$d_{cp} = d_0 + \frac{(d_{end} - d_0)}{l_{all}} l_{cp}. \tag{26}$$

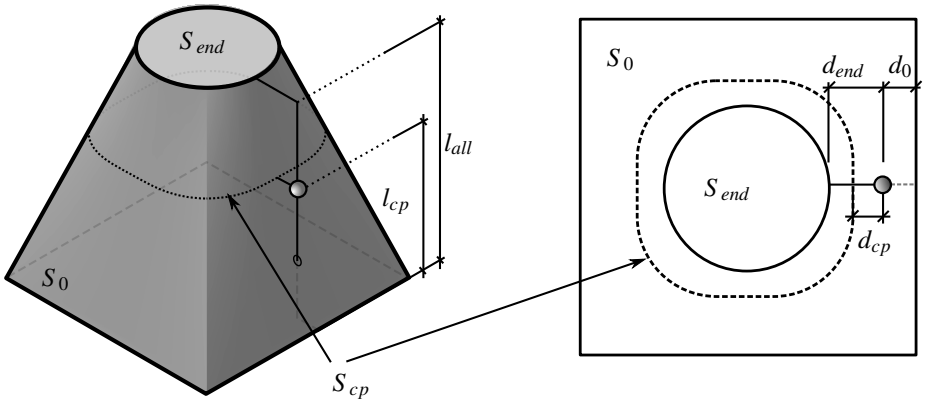


Figure 12: Point-in-membership test on intermediate sketch S_{cp} of a loft.

3.4. Ray-casting with Splines (in 2D)

As presented in fig. 11, a point-in-membership test can be performed with ray-casting on a dimensionally reduced model, i.e. in 2D. This is by far simpler than carrying out ray-casting in 3D. On the two-dimensional intermediate sketch, a ray from a point that is definitely outside of the domain $\mathbf{P}_{\text{out}}$ to the point of interest $\mathbf{P}$ is cast forth, and all intersections with the contour curves are counted. Possible elements for the contour lines are straight lines, arcs, or splines (B-Spline, NURBS). Problematic are the unlikely cases, in which the ray intersects the contour at a corner point (see fig. 13(a)), or in which the ray is collinear with a contour line (see fig. 13(b)) as the point membership is ambiguous. However, these cases can easily be detected and the algorithm simply changes the position of the reference point $\mathbf{P}_{\text{out}}$.

Journal publication: https://doi.org/10.1016/j.camwa.2017.01.027

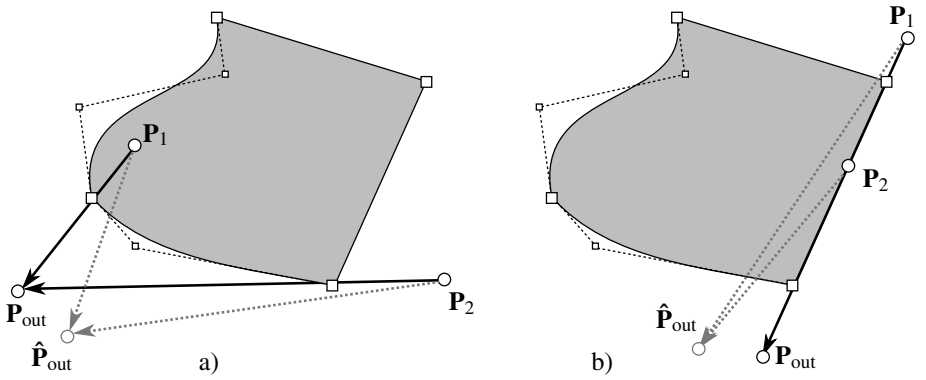


Figure 13: Problematic ray cast scenarios, where a) the ray intersects at a corner point and b) the ray is collinear with a contour line. However, a simple shift of the reference point ($\mathbf{P}_{\text{out}} \rightarrow \hat{\mathbf{P}}_{\text{out}}$) can resolve these problems.

While there are analytical solutions to find the intersections of the ray and a line or an arc, the evaluation of the number of intersections between a ray and a spline is not straightforward (see fig. 14). These intersections can be found using a brute force method, which performs an intersection search with a fixed set of initial parameter values along the ray. This is very inefficient, as it is necessary to perform several closest point searches, containing several inverse mappings onto the spline, for each intersection search. Moreover, most of the initial values will find the same intersection points and, furthermore, it is not guaranteed that all intersections are found. Typically, Newton's method is used for the inverse mapping. Newton's method is highly dependent on the initial value. Hence, two intersections lying close to each other might be detected as only one intersection. A robust and efficient way to obtain all zeros of Bézier curves with their multiplicity is presented in [34]. However, the algorithm presented therein is specific to Bézier curves and not tuned to deliver the parity (i.e. whether the number of intersections is odd or even) of the intersections. The algorithm presented in this paper can use any spline description such as Bézier curves, B-Splines or NURBS. Moreover, as only the parity of intersections is needed, in most cases, an evaluation of the intersection points is not necessary. In this context, an efficient and robust method to find the parity of intersections of a (ray) line and a spline is depicted in fig. 32 and algorithm 2.

As only the parity of intersections is needed, the convex hull property of the splines can be used. Let us consider cases a) and b) in fig. 14. In these cases, the ray segment (between the point of interest $\mathbf{P}$ and the point outside the model $\mathbf{P}_{\text{out}}$) has the same number of intersections as the infinite ray and, thus, determines the parity. This is due to the convex hull property which guarantees that the amount of intersection points between a line and a spline cannot be larger than the amount of intersections with a line and the spline's control polygon. There can be less intersections, but this does not change the parity of intersections.
The cases c) and d), as depicted in fig. 14, are more sophisticated. Here, the ray segment $\mathbf{r}_{\text{fin}}$ and an infinite ray $\mathbf{r}_{\text{inf}}$ do not have the same number of intersections and, hence, it is necessary to determine intersection points of the ray with the spline. To this end, suitable starting values are needed for the Newton iteration. For the determination of these starting values several possibilities are available. One possibility is presented in appendix A.2. For simplicity of formulation, we assume that the ray corresponds to the positive x-axis. In other cases, a suitable transformation to a local coordinate system is performed (see appendix A.2). All zeros which fulfill the following property are intersection points with the spline, and their number determines the parity for the ray cast.
The cases e) and f), depicted in fig. 14, where the rays intersect with the line segment between the first and last control point (in the following denoted as $\mathbf{l}_{\text{close}}$) are covered by an additional intersection test of infinite ray and line $\mathbf{l}_{\text{close}}$. This, of course, leads to a different parity from the (finite) ray segment, and hence results in an expensive inverse mapping. Thus, it desirable to choose the point $\mathbf{P}_{out}$ in such a way, that the cases e) and f) are seldom. The best choice is to place the point $\mathbf{P}_{out}$ on to the line $\mathbf{l}_{\text{close}}$ (see fig. 14(g) and (h)). This

Journal publication: https://doi.org/10.1016/j.camwa.2017.01.027

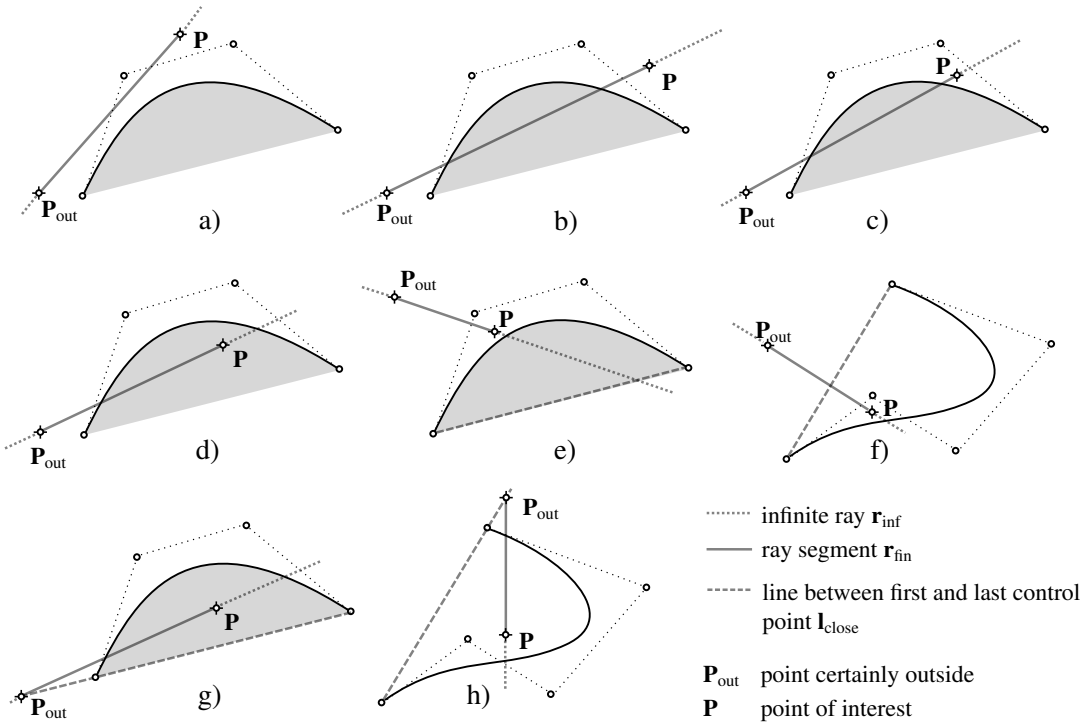


Figure 14: Different ray-cast scenarios: (a) Intersection only with control polygon. (b) Same number of intersections with spline as with control polygon. (c) Point between control polygon and spline (d) Point inside. In cases (e) and (f) a sole count of intersections with the control polygon will fail. To this end the infinite ray is also intersected with a line segment between first and last CP. At cases (g) and (h) the point outside $\mathbf{P}_{\text{out}}$ is set to be on the closing line $\mathbf{l}_{\text{close}}$, thus, avoiding in most cases an expensive full intersection test.

might however not always be possible. In this context consider a contour, which is constructed of more than one spline. Even for the simple case of two splines, where most likely an intersection point of both closing lines exist, i.e. $\mathbf{l}_{\text{close}_1} \cap \mathbf{l}_{\text{close}_2} \neq \emptyset$ it is not guaranteed that this point also lies outside of the domain. In these cases the point outside will be set on $\mathbf{l}_{\text{close}}$ of one spline, hence avoiding the expensive point-in-membership test at least for one spline.

4. Numerical examples

In this chapter, we present three different examples, which are created with extended primitives and extended operations. They are designed to provide an overview of the possibilities of the modeling approach presented in this paper.

4.1. Sweep example

The first example involves a coil spring, which is constructed using sweeps (see fig. 15). The spring is constructed by three primitives, which are combined with two union operations. For all three bodies, a circle sketch with a radius $r_{\text{sketch}} = 1$ is swept along the corresponding sweep path which, for the bottom and top torus, is again a circle of radius $r_{\text{torus}} = 10$. Bottom and top tori are aligned to the $x - y$ plane and located at $z_{\text{bottom}} = 0$ and $z_{\text{top}} = 24$, respectively. The computational domain ranges from $z = 0$ to $z = 1$, thus clipping

Journal publication: https://doi.org/10.1016/j.camwa.2017.01.027

bottom and top tori into half-tori. The sweep path of the coil is described by a helical NURBS of degree $p = 2$, the knot vector

$$U = [0,0,0,1,1,2,2,3,3,4,4,5,5,6,6,7,7,8,8,9,9,10,10,11,11,12,12,12] \quad ,$$

and the 3D control points with weights $w_1 = 1$ or $w_2 = \frac{1}{\sqrt{2}}$

$$\mathbf{P}_i = \begin{pmatrix} x_i \\ y_i \\ z_i \\ w_i \end{pmatrix} = \begin{bmatrix} 10 & 10 & 0 & -10 & -10 & -10 & 0 & 10 \\ 0 & 10 & 10 & 10 & 0 & -10 & -10 & -10 \\ 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 \\ w_1 & w_2 & w_1 & w_2 & w_1 & w_2 & w_1 & w_2 \end{bmatrix} \cdots$$

$$\cdots \begin{matrix} 10 & 10 & 0 & -10 & -10 & -10 & 0 & 10 \\ 0 & 10 & 10 & 10 & 0 & -10 & -10 & -10 \\ 8 & 9 & 10 & 11 & 12 & 13 & 14 & 15 \\ w_1 & w_2 & w_1 & w_2 & w_1 & w_2 & w_1 & w_2 \end{matrix} \cdots$$

$$\cdots \begin{bmatrix} 10 & 10 & 0 & -10 & -10 & -10 & 0 & 10 & 10 \\ 0 & 10 & 10 & 10 & 0 & -10 & -10 & -10 & 0 \\ 16 & 17 & 18 & 19 & 20 & 21 & 22 & 23 & 24 \\ w_1 & w_2 & w_1 & w_2 & w_1 & w_2 & w_1 & w_2 & w_1 \end{bmatrix} .$$

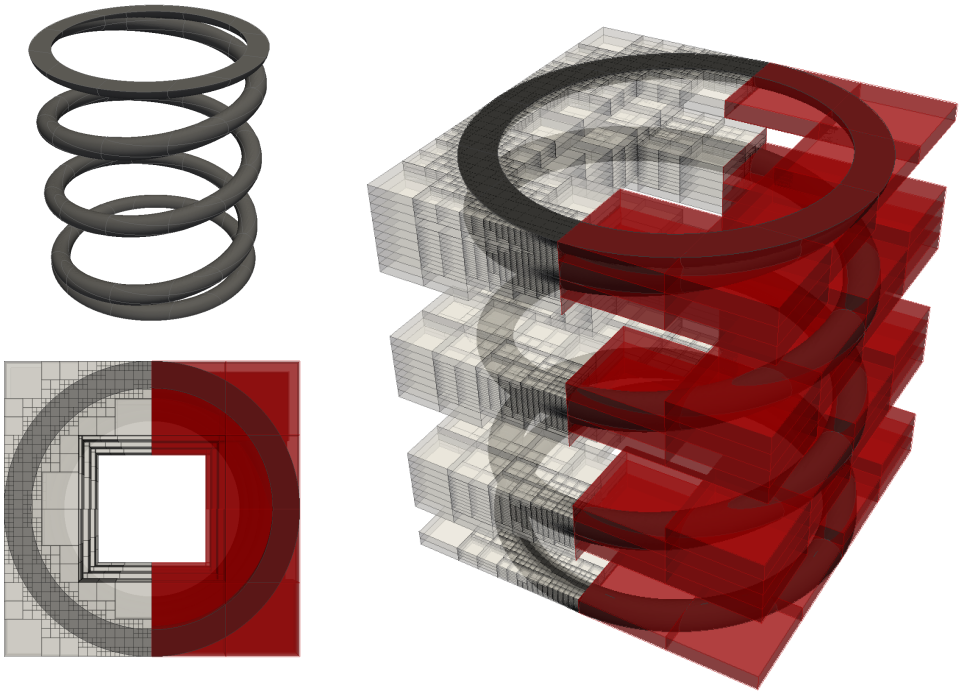


Figure 15: Coil spring model: Depicted are the finite cells (red) on the one side and the octree partitioning with a refinement depth of $k = 4$ on the other side (light gray), respectively, for top and isometric view.

For the simulation, we choose 4x4x24 finite cells with integrated Legendre shape functions and a polynomial degree of $p = 7$ in the trunc space [24]. Cells that are located completely outside the coil are deactivated,

Journal publication: https://doi.org/10.1016/j.camwa.2017.01.027

reducing the number of active finite cells from 384 to 134. The integration of the element matrices is carried out using an adaptive octree, whose maximum partitioning depth is preset to $k = 6$.

In this model, strong Dirichlet boundary conditions are applied at the top and bottom faces of the model, which correspond to the boundaries of the finite cells. The degrees of freedom here are fixed in all directions, except for the vertical displacement on the top face which is predefined by $\hat{\mathbf{u}}_z = -5$. Figure 16 shows the resulting displacement and the von Mises stresses, which provide a good overall insight into the structural load carrying behavior. They are smooth throughout most part of the domain and exhibit singular behavior at the intersection curves of the top and bottom tori with the coil. As in any finite element computation, an accurate solution of local stresses at singularities can not be accurately resolved without an appropriate refinement. A local refinement is not carried out for this example, but it is possible, e.g. by application of the multi-level *hp*-method recently proposed in [35].

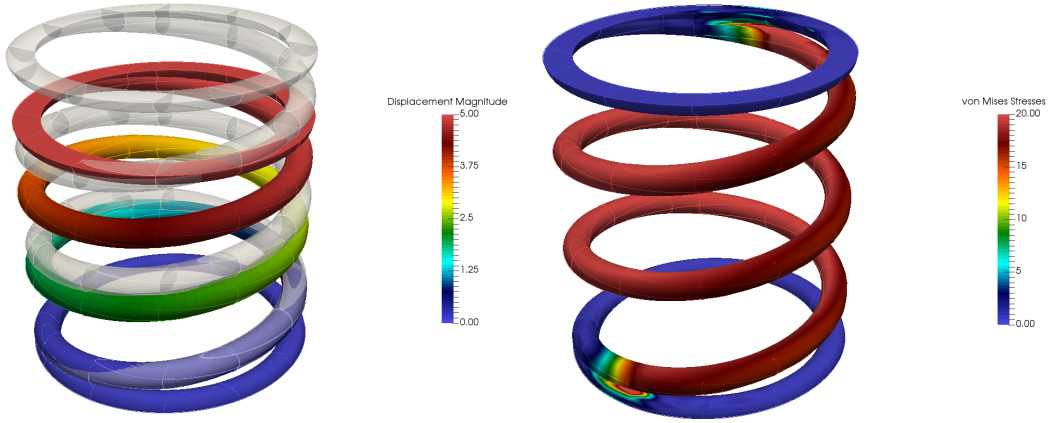


Figure 16: Displacements and von Mises stresses of the coil spring under predefined top displacement of $\hat{\mathbf{u}}_z = -5$.

It is noteworthy that only the CSG model is used in all involved steps, i.e. from the set-up of the model until the computation itself. A conversion from the CSG-model to an explicit B-rep is only carried out for the visualization of the results in the post-processing. Here, the marching cubes algorithm [17] is used to derive a triangulated surface on which the results are post-processed. However, even this conversion for visualization is not mandatory as volumetric post-processing is a possible option as well.

4.2. Loft example

The second example is a pipe elbow starting with a circular profile and ending with a rectangular cross section. It is constructed as a procedural model and then transformed to a CSG tree (see fig. 17). It combines several simple primitives (cylinders and cuboids) and two lofts along a quadratic B-Spline loft path. The dimensions of the example are depicted in fig. 18.

Journal publication: https://doi.org/10.1016/j.camwa.2017.01.027

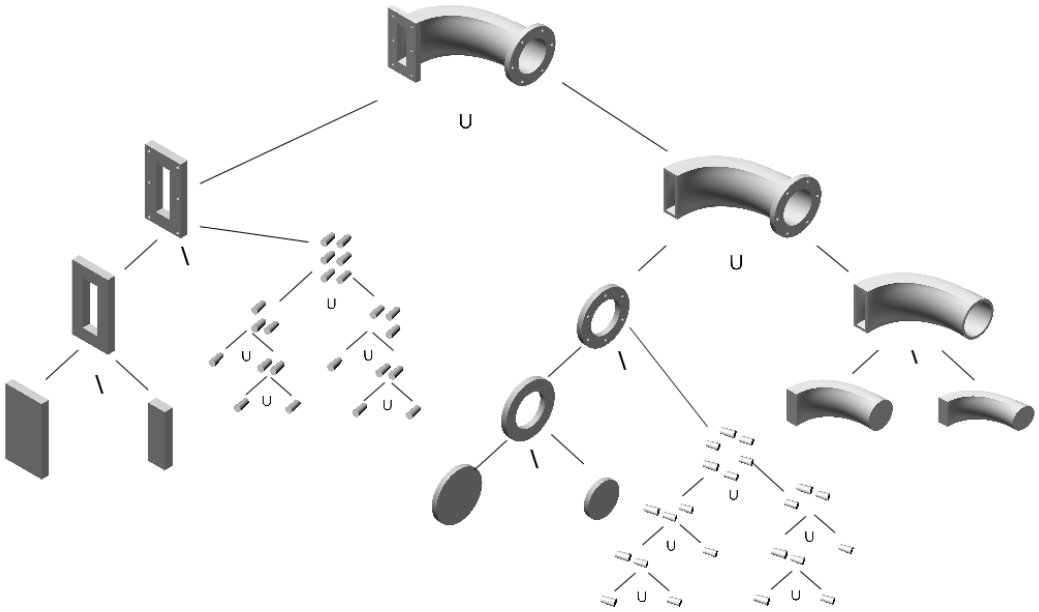


Figure 17: CSG tree of loft.

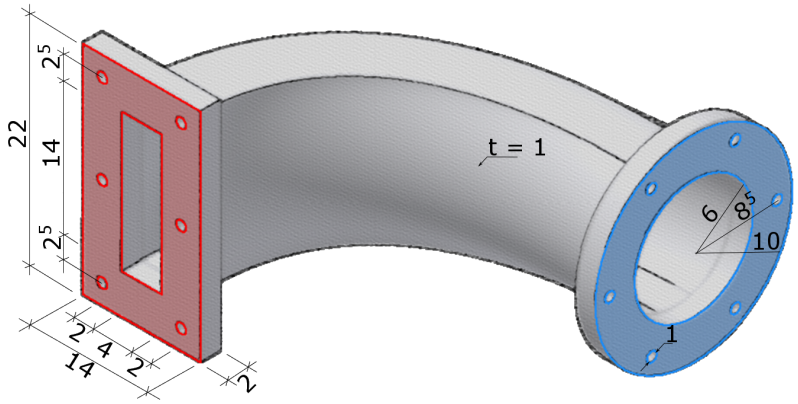


Figure 18: Dimensions of the loft.

Again, two models are created to show another advantage of the FCM and its capability to use the explicit volume description of CSG models. In parametric design, even a slight change of few parameters may result in a re-meshing of parts, or even the entire model. In this example, the position of one control point of the loft path is changed. This results in a slightly different model (see fig. 19). As this only leads to a change in the geometry, not in the topology, the mesh for a FEM-simulation does not necessarily require a re-meshing – but it could be morphed. However, in the proposed work-flow, the change in the geometry does not need any special treatment: As it is only the loft path that changes, the same CSG-tree can be used with the same computational mesh. The change of the geometry is resolved on the integration level, leading to a different composed integration.

Journal publication: https://doi.org/10.1016/j.camwa.2017.01.027

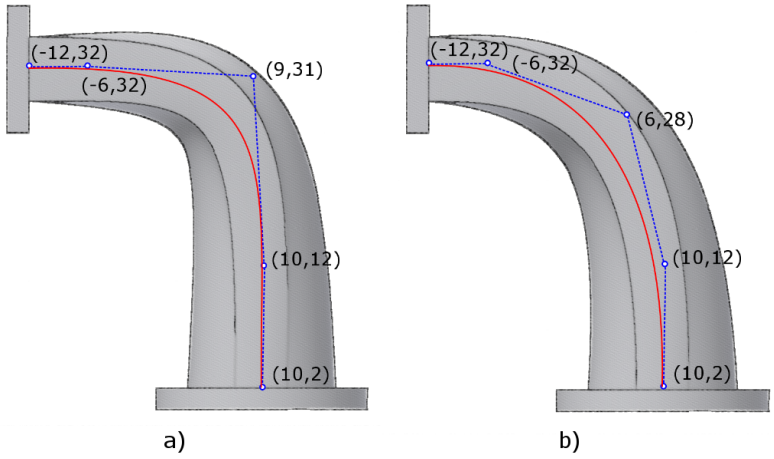


Figure 19: (a) Loft path of the first model. (b) Loft path of the second model.

The resolution of the model is chosen to be 20x15x20 cells with integrated Legendre shape functions and a polynomial degree of $p = 5$. As in the previous example, it is sufficient to consider cells that contain parts of the physical domain. The base plate (round, blue) is fixed, and a predefined deflection $\hat{\mathbf{u}} = 1$ is applied onto the left base plate (rectangular, red) using strong Dirichlet boundary conditions. For a precise integration of the stiffness matrix, the cells are partitioned with an octree to a maximum depth of five subdivisions.

Figure 20 shows the finite cells embedding the structural model and the computed displacements for both examples. The von Mises stresses are depicted in fig. 21.

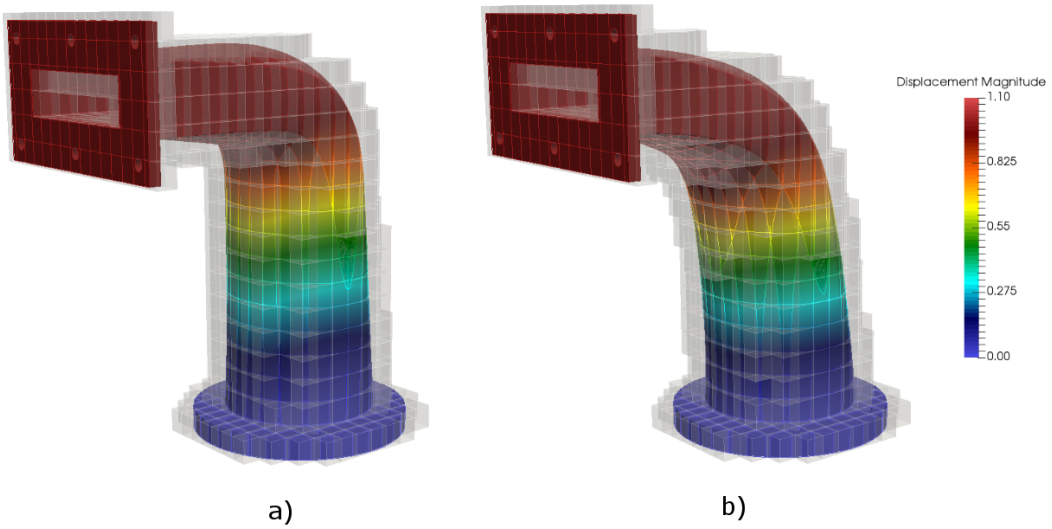


Figure 20: Displacements for the (a) original example and the (b) modified example.

Journal publication: https://doi.org/10.1016/j.camwa.2017.01.027

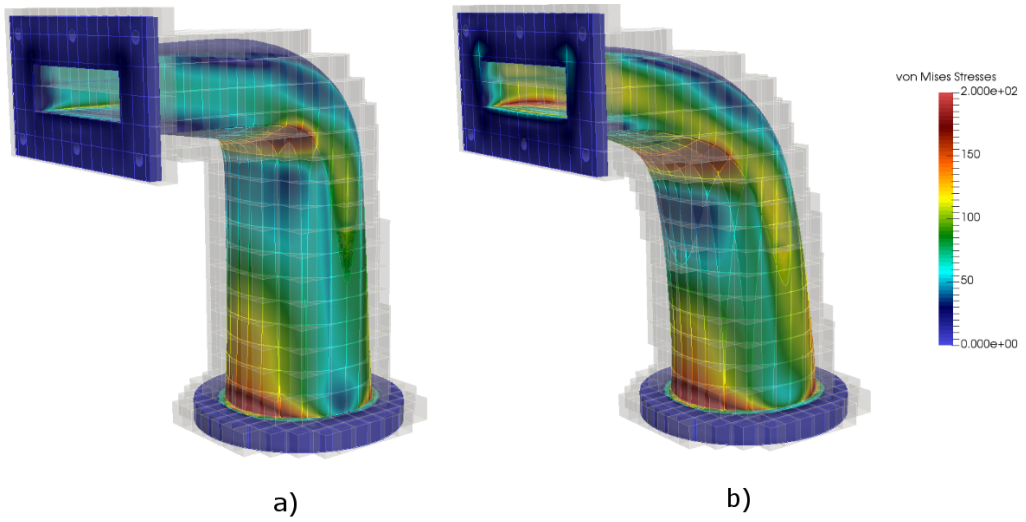


Figure 21: Von Mises stresses for the (a) original example and the (b) modified example.

As pointed out before, a geometrical change does not influence the proposed work-flow. In this case, even a topological change has hardly any impact. In a classical FEM-simulation, however, a re-meshing of the affected region would be required. A simple modification of the model is applied to illustrate a topological change. The number of holes in the rectangular plate is set to (i) six in the first and (ii) four in the second case. A Neumann boundary condition is applied to the holes - via added washers at each hole. The loaded surfaces of the washers are automatically recovered using the marching cubes algorithm. A pressure of $\hat{p} = 1$ is deployed to the upper row of washers, and a pressure of $\hat{p} = -1$ to the lower row, inducing a moment onto the rectangular plate (see fig. 22).

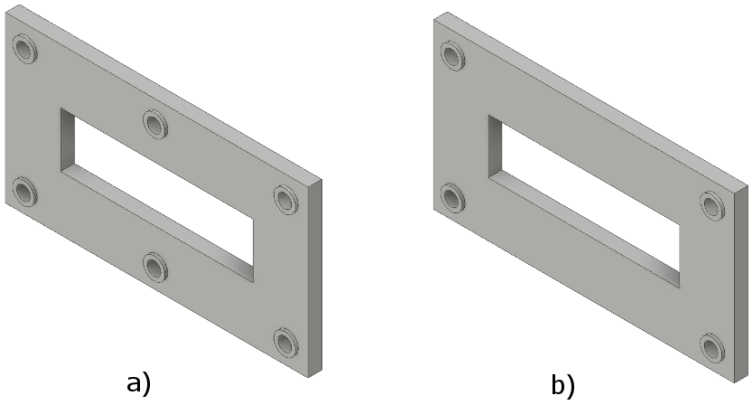


Figure 22: Different topological models: (a) Rectangular plate with 6 holes (b) and 4 holes.

Figure 23 shows the displacements and fig. 24 von Mises stresses at the rectangular plate for the two topologically different examples.

Journal publication: https://doi.org/10.1016/j.camwa.2017.01.027

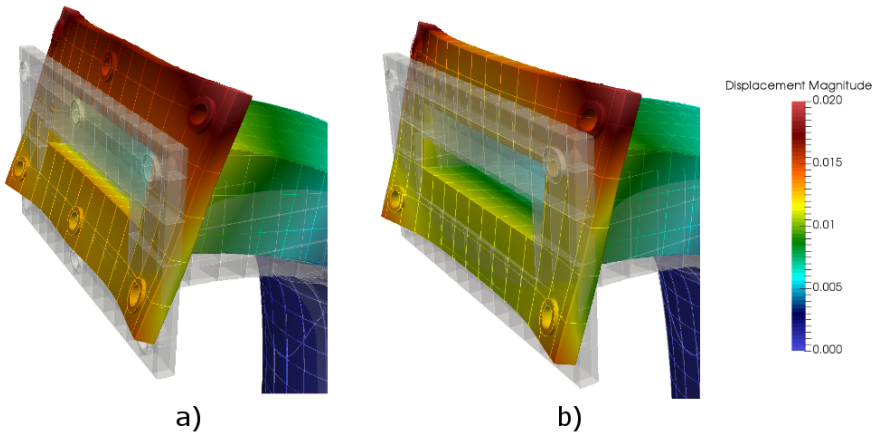


Figure 23: Displacements at the rectangular plate for the (a) first example and the (b) second example. (Displacements scaled by a factor 200)

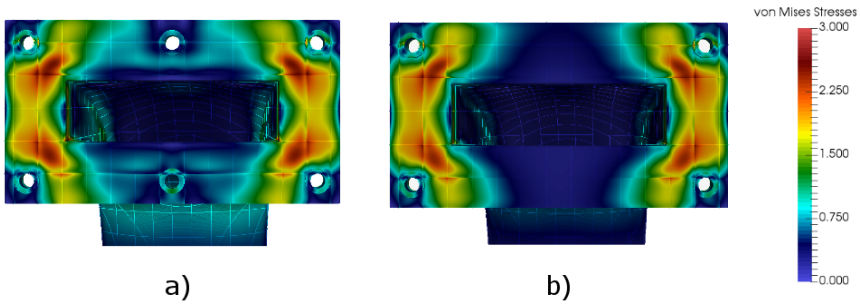


Figure 24: Von Mises stresses at the rectangular plate for the (a) first example and the (b) second example.

4.3. Extended operations example

The third example deals with commonly used extended operations. Starting from a cube, four edges are chamfered, two edges are filleted, three holes are drilled, and one shell operation is applied (see fig. 25).

As mentioned in section 2.1.2, the extended operators fillet, chamfer, and drilling a hole are just a combination of the original three boolean operations. Shells, however, are not a straightforward extension. The shell operation is applied to one surface, caving the volume while keeping a defined wall thickness to all other surfaces. In the present model, it is possible to manually model the shell operation with a set of Boolean operations and additional primitives by user interaction. This, however, is not straightforward for more sophisticated cases.

Two almost identical models serve to show the capability of the FCM. In the first model, the (solitary) borehole 1 is created by subtracting a cylinder. In the second model, the hole is created by two extruded half cylinders, which are shifted by 0.05% w.r.t. to the extension of the entire model (see fig. 26). This shift does not have any significant influence on the results of the simulation, but it does lead to a considerable increase in the effort needed for the classical FEM, where body fitted meshes must be used. Figure 27 shows a mesh created by Netgen [36] and the different refinements in the region around the holes. Although the shift is very small and only applied at one hole, the mesh of the second model has around 18 times more elements. Moreover, many

Journal publication: https://doi.org/10.1016/j.camwa.2017.01.027

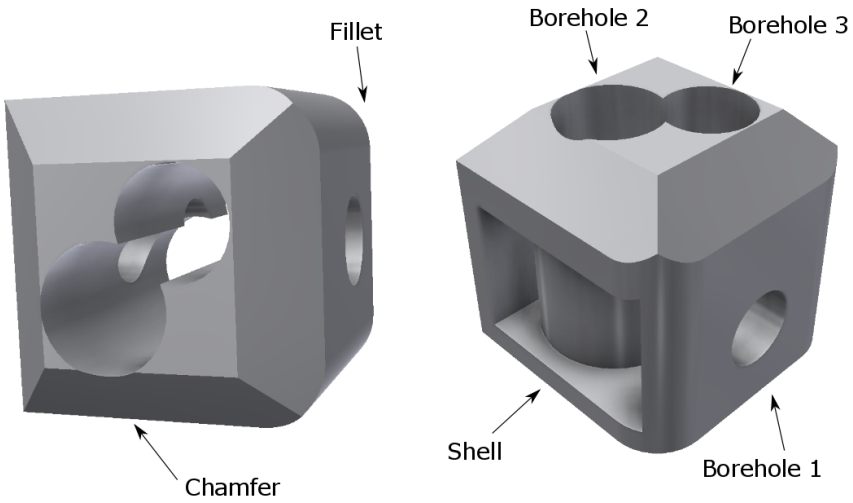


Figure 25: Extended operations applied on a cube: chamfer, fillet, drilling a hole and shell operation.

elements are very badly conditioned. Even if this mesh represents the 'exact' geometry of the CAD model, it is very probable that it is not the intended finite element discretization. Similarly, there are often seemingly unmotivated refinements in practical applications, which is why considerable engineering efforts have to be made to remodel a structure before an efficient numerical analysis can be carried out.

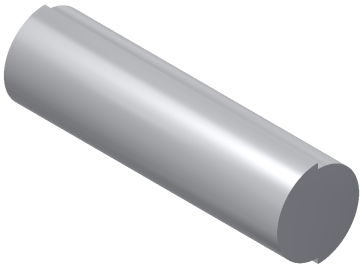


Figure 26: For drilling the borehole 1 in the second model: Two slightly shifted cylinders (Shift here not to scale!).

Journal publication: https://doi.org/10.1016/j.camwa.2017.01.027

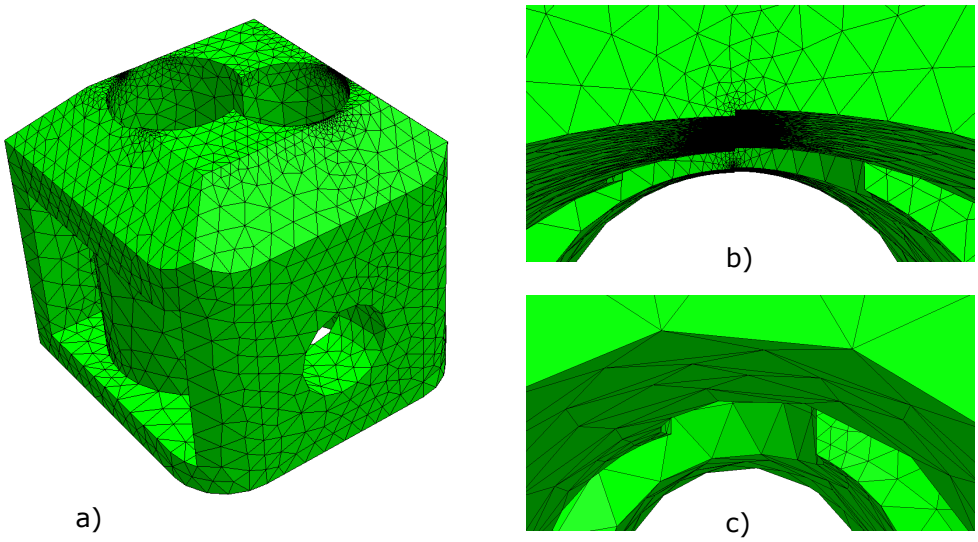


Figure 27: Model meshed using Netgen [36]: (a) Global view of the mesh, (b) refinement around the hole, which is constructed by two slightly shifted half-cylinders and (c) refinement around a hole without the (unnecessary) geometric detail.

For the FCM simulation, 10x10x10 finite cells using integrated Legendre polynomials of degree $p = 5$ and a octree partitioning depth of $k = 4$ are used. Again, the lower face is fixed, and the top face is displaced downwards by a value of $dz = -0.5$ using strong Dirichlet boundary conditions. Figure 28 shows the displacements and von Mises stresses of the model. A close-up of the displacements and von Mises stresses around the shifted hole is depicted in fig. 29. The results of the FCM computation for the two models (full cylinder versus two shifted half-cylinders to create borehole 1) are identical up to machine precision (10^{-9}), proving that the proposed method is robust against imprecise CAD modeling.

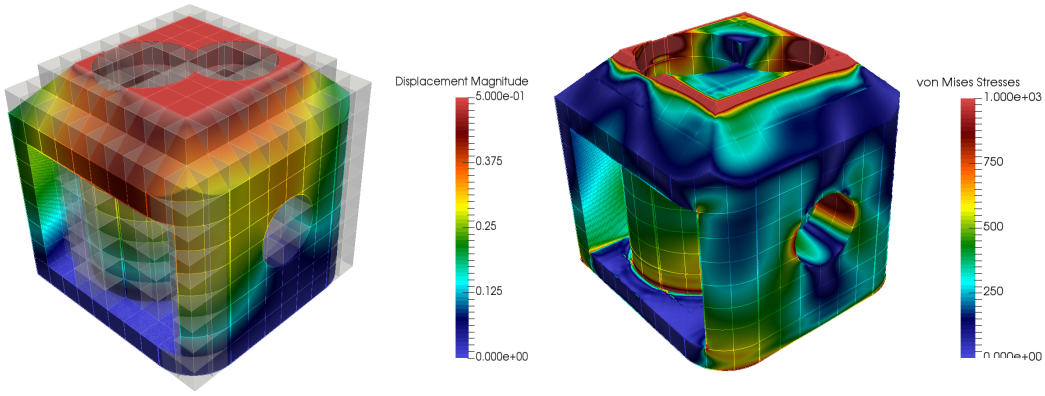


Figure 28: Displacements with active cells and von Mises stresses

Journal publication: https://doi.org/10.1016/j.camwa.2017.01.027

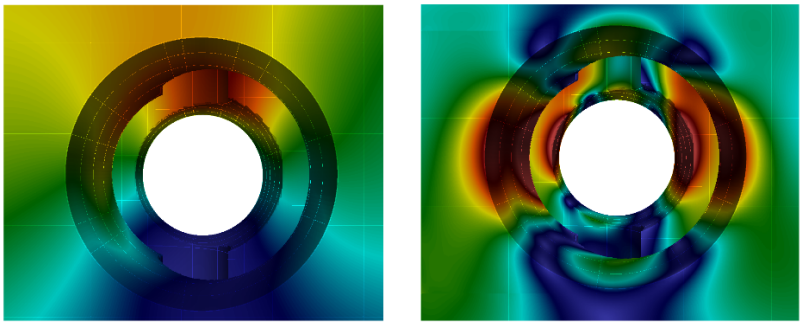


Figure 29: Close up at borehole 1: (a) Displacements and (b) von Mises stresses

5. Conclusions

For the industry, integrated design processes and numerical analyses without complex transitions such as meshing are of high relevance, and many research groups have focused on this topic during the last years. While Isogeometric Analysis provides an excellent method for the numerical simulation of boundary representation models and shell-like structures, this paper focused on models created with Constructive Solid Geometry. A method has been presented that leads straight from CAD modeling to a numerical simulation using CSG and the Finite Cell Method (FCM). FCM is able to use the implicit model description provided by the CSG model directly, which simplies the meshing process significantly. It was shown that point-in-membership tests can be carried out efficiently for extended primitives like sweeps, bodies of revolution, and lofts. Also, extended operations such as fillet, chamfer, and holes can be applied to the model easily. Using FCM as a simulation technique that can deal with the explicit geometry description of CSG models has several advantageous properties. Following the design-through-analysis idea, a meshing step can be skipped. Also, as the CSG is inherently watertight, the geometry does not have to be pre-processed. Furthermore, CSG provides information about the interior of the models.

Although CSG modeling does not support the modeling of free form surfaces and objects, it is still possible to include these as B-Rep primitives to the CSG tree, provided that they are watertight and thereby support a reliable point membership classification.

Nevertheless, the presented methodology has also some drawbacks. Shell operations require user action and are typically not straightforward. Another issue is the conversion from a sequential model to a CSG tree. In our implementation, the CSG tree is constructed parallel to the information input of the sequential model, resulting in a sub-optimal tree, which is typically of high depth. However, it is possible to rearrange the CSG-tree in a way that allows to skip many point-in-membership tests. Further challenges are the complex curve descriptions for sweeps and lofts by B-Splines and NURBS. As no explicit inverse mapping is available, the point-in-membership test for bodies defined by these curves can become computationally costly. Several approaches were made to improve the efficiency, among others the improved intersection test on B-Splines and NURBS (see Chapter 3.4), which can avoid most of the inverse mappings. Also, the adoption of intersections of primitives with their bounding boxes led to a considerable speedup.

Journal publication: https://doi.org/10.1016/j.camwa.2017.01.027

A. Appendix

A.1. Rotated local basis system

In contrast to the cases presented in section 3.3 the point-in-membership test is more sophisticated for the following cases (see fig. 30). Case (a) shows a sweep or loft whose sketch plane is not orthogonal to the sweep path at the starting point $\mathbf{C}(\xi_0)$. Nevertheless, the local basis of the sketch $\mathbf{B}$ and the basis spanned by the sweep path $\mathbf{A}$ are fixed in a certain relation, i.e. the dihedral angle ϕ between $\mathbf{A}$ and $\mathbf{B}$ remains constant along the path. In case (b), the normal of the intermediate sketches does not follow the tangent of the sweep path, but both starting and ending sketch remain parallel to each other. Finally, in case (c), the normal orientation of the sketch is only known at the starting and the ending sketch, and it changes along the sweep path. This latter case is most likely to occur during the construction of lofts where the normal of the starting and ending sketch often do not have the same angle to the tangent vector.

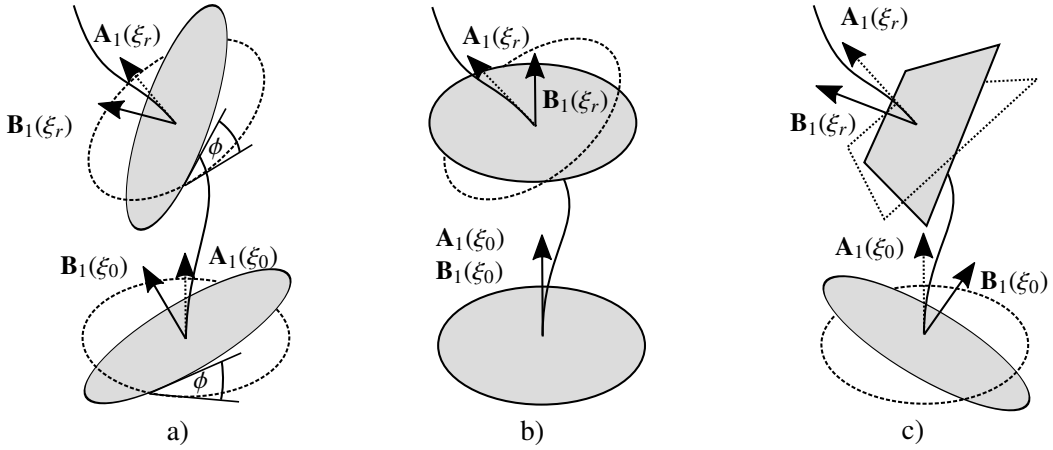


Figure 30: Considering a sweep or loft: (a) Sketch rotated local basis system of the sweep path under a certain constant dihedral angle ϕ. (b) Sketches always parallel to starting sketch. (c) Local coordinates system only known for starting and ending sketch.

For these three cases, an intermediate plane cannot be created at the closest point, but must be created in such a way that the z-coordinate of the mapped point $\tilde{\mathbf{P}}$ is zero. The local sketch basis spanned here is denoted by $\mathbf{B}(\xi_r)$.

$$\tilde{\mathbf{P}} = \begin{bmatrix} \tilde{P}_x \\ \tilde{P}_y \\ \tilde{P}_z \stackrel{!}{=} 0 \end{bmatrix}. \tag{27}$$

In the following, the steps to determine the intermediate sketch basis $\mathbf{B}(\xi_r)$ for the three cases are presented.

Journal publication: https://doi.org/10.1016/j.camwa.2017.01.027

Algorithm 1 Case (a): Rotated local basis systems under constant dihedral angle

Let ξ_0 be the parameter value of the starting point of the sweep path

0. Express at ξ_0 the base vectors of the sketch $\mathbf{B}_i(\xi_0)$ in terms of the local base of the path $\mathbf{A}_i(\xi_0)$.
Comment: This needs only to be done once at the set-up of the sketch
procedure FIND **A**-**B** RELATION
    Compute 3x3 transformation matrix $\mathbf{T} = \begin{bmatrix}\mathbf{T}_1 & \mathbf{T}_2 & \mathbf{T}_3\end{bmatrix}$
    for all $\mathbf{A}_i$ **do**
        for all $\mathbf{B}_j$ **do**
            $T_{ij} = \mathbf{A}_i(\xi_0) \cdot \mathbf{B}_j(\xi_0)$
        end for
    end for
end procedure

procedure FIND LOCAL BASIS SYSTEM $\mathbf{B}(\xi_r)$
    1. Get an initial guess on the sweep path
    Get closest point on curve $\mathbf{C}(\xi_r^{j_0})$
    Store $\xi_r^{j_0}$ as initial value for Newton iteration

    2. Apply Newton's method to find $\mathbf{B}(\xi_r)$ (see fig. 31).
    Let $\mathrm{d}\xi$ be a sufficiently small parameter increment for finite differences
    Initialize z-coordinate of (first) mapped point of interest $|\tilde{P}_z^1| \overset{!}{>} \varepsilon$
    while $|\tilde{P}_z^1| > \varepsilon$ **do**
        2.1 Create local basis of sketch $\mathbf{B}(\xi_r^j)$ at ξ_r^j
        Create local basis system of the sweep path $\mathbf{A}(\xi_r^j)$ at $\mathbf{C}(\xi_r^j)$ (e.g. Frenet base)
        for all $\mathbf{B}_i(\xi_r^j)$ **do**
            Get basis vectors of sketch base $\mathbf{B}_i(\xi_r^j) = \mathbf{A}(\xi_r^j)\,\mathbf{T}_i$
        end for

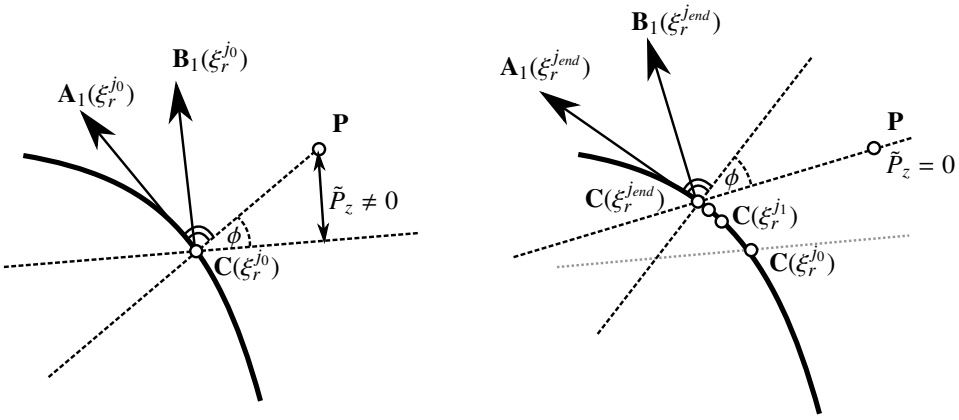
        2.2 Create local basis of sketch $\mathbf{B}(\xi_r^j + \mathrm{d}\xi)$ at $\xi_r^j + \mathrm{d}\xi$
        Create local basis system of the sweep path $\mathbf{A}(\xi_r^j + \mathrm{d}\xi)$ at $\mathbf{C}(\xi_r^j + \mathrm{d}\xi)$ (e.g. Frenet base)
        for all $\mathbf{B}_i(\xi_r^j + \mathrm{d}\xi)$ **do**
            Get basis vectors of sketch base $\mathbf{B}_i(\xi_r^j + \mathrm{d}\xi) = \mathbf{A}(\xi_r^j + \mathrm{d}\xi)\,\mathbf{T}_i$
        end for

        2.3 Map point of interest $\mathbf{P}$ to local sketch basis systems $\mathbf{B}(\xi_r^j)$ and $\mathbf{B}(\xi_r^j + \mathrm{d}\xi)$.
        First mapped point $\tilde{\mathbf{P}}^1 = \mathbf{P} \rightarrow \mathbf{B}(\xi_r^j)$
        Second mapped point $\tilde{\mathbf{P}}^2 = \mathbf{P} \rightarrow \mathbf{B}(\xi_r^j + \mathrm{d}\xi)$

        2.4 Next Newton step
        Newton step $\xi_r^{j+1} = \xi_r^j - \tilde{P}_z^1 \,/\, [(\tilde{P}_z^2 - \tilde{P}_z^1) \,/\, \mathrm{d}\xi]$
    end while

    return $\mathbf{B}(\xi_r^{j_{end}})$
end procedure

Journal publication: https://doi.org/10.1016/j.camwa.2017.01.027



Initial Newton step j_0 Final Newton step j_{end}

Figure 31: For the case that the dihedral angle ϕ between the local path basis $\mathbf{A}$ and the local sketch basis $\mathbf{B}$ remains constant: Find the point on the curve $\mathbf{C}(\xi_r)$ so that $\tilde{P}_z = 0$ with $\tilde{\mathbf{P}}$ being the point of interest mapped to local sketch basis $\mathbf{B}$. Starting with closest point on the curve $\mathbf{C}(\xi_r^{j_0})$, a Newton iteration is carried out until $\tilde{P}_z = 0$.

Case (b): According to case (a) (Algorithm 1), a local basis system $\mathbf{B}(\xi_r)$ must be found such that $\tilde{P}_z = 0$. In contrast to (a), the basis $\mathbf{B}(\xi_r)$ can be evaluated easier, as the base vectors $\mathbf{B}_i(\xi)$ are constant along the sweep path. Only the origin moves corresponding to the curve point $\mathbf{C}(\xi_r^j)$.

Case (c): As the relations between the local basis system of the path $\mathbf{A}$ and sketch $\mathbf{B}$ are only known at the starting and ending points $\mathbf{C}(\xi_0)$ and $\mathbf{C}(\xi_{end})$, a transformation matrix for both is set-up, $\mathbf{T}_0$ and $\mathbf{T}_{end}$, similar to case (a)(Algorithm 1). At each point on the path $\mathbf{C}(\xi_i)$ with both transformation matrices $\mathbf{T}_0$ and $\mathbf{T}_{end}$, a local sketch basis is formed $\mathbf{B}^{\mathbf{T}_0}(\xi_i)$ and $\mathbf{B}^{\mathbf{T}_{end}}(\xi_i)$. The basis vectors $\mathbf{B}_i(\xi_i)$ are (linearly) interpolated between $\mathbf{B}_i^{\mathbf{T}_0}(\xi_i)$ and $\mathbf{B}_i^{\mathbf{T}_{end}}(\xi_i)$ using the arc-length of the current point, similar to the point-in-membership test for lofts (see section 3.3.3).

Journal publication: https://doi.org/10.1016/j.camwa.2017.01.027

A.2. 2D Ray casting on spline curves

In the following, an algorithm for ray-casting with splines in 2D is presented. It also deals with the general case in which the ray does not coincide with the positive x-axis.

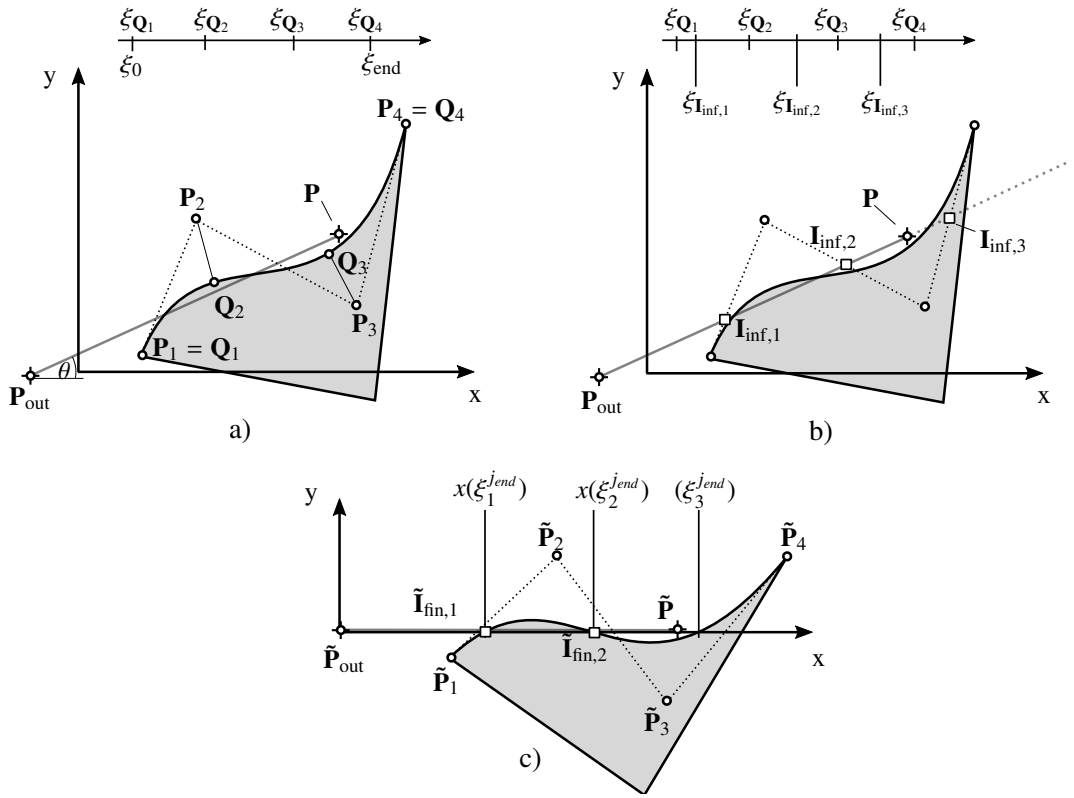


Figure 32: Procedure of ray-casting with splines: a) Mapping the control points to the parameter space $\xi_{\mathbf{Q}_i}$. b) Use linear interpolation to map intersection points with control polygon $\mathbf{I}_{\text{inf},k}$ to spline parameter space $\xi_{\mathbf{I}_{\text{inf},k}}$. c) Perform linear transformation such that the ray lies on the x-axis and perform a zero search to obtain the intersections $\tilde{\mathbf{I}}_{\text{fin},k}$.

Algorithm 2 Spline ray casting

Let $\mathbf{C}(\xi)$ be a spline defined by its control points $\mathbf{P}_i$
Let n be the number of control points $\mathbf{P}_i$
Let the ray $\mathbf{r}_{\text{inf}}$ and the segment $\mathbf{r}_{\text{fin}}$ be defined by a point certainly outside $\mathbf{P}_{\text{out}}$ and the point of interest $\mathbf{P}$

procedure INITIALIZE SPLINE CURVE
    0. Obtain for each control point a corresponding parameter value (fig. 32 a)).
    Comment: This needs only to be done once at the set-up of the spline
    for $i := 1$ **to** n **do**
        Find the closest point $\mathbf{Q}_i$ on the spline to control point $\mathbf{P}_i$ using eq. (24)
        Find corresponding parameter $\xi_{\mathbf{Q}_i}$
    end for
end procedure

Journal publication: https://doi.org/10.1016/j.camwa.2017.01.027

Algorithm 2 Spline ray casting (continued)

procedure RAY CAST ON SPLINE
1. Find intersection points between control polygon and finite and infinite ray (fig. 32 b)).
Let j and k denote the j-th and k-th intersection of the control polygon with the ray segment or the infinite ray respectively.
for $i := 1$ **to** $n - 1$ **do**
Find intersection point $\mathbf{I}_{\text{fin},j}$ between ray segment $\mathbf{r}_{\text{fin}}$ and control line $\overline{\mathbf{P_i P_{i+1}}}$
Find intersection point $\mathbf{I}_{\text{inf},k}$ between infinite ray $\mathbf{r}_{\text{inf}}$ and control line $\overline{\mathbf{P_i P_{i+1}}}$
end for
Find intersection point $\mathbf{I}_{\text{inf},k}$ between infinite ray $\mathbf{r}_{\text{inf}}$ and line segment $\overline{\mathbf{P_1 P_n}}$

2. Check for number of intersections for finite and infinite ray.
if $\#(\mathbf{I}_{\text{fin}})$ **equals** $\#(\mathbf{I}_{\text{inf}})$ **then**
2.1 Case: Same number of intersections for finite and infinite ray (fig. 14 a) and b)).

return $modulus\left(\frac{\#(\mathbf{I}_{\text{inf}})}{2}\right)$
else
2.2 Case: Different number of intersections for finite and infinite ray (fig. 14 c) and d)).

3. Interpolate starting values for intersection search (fig. 32 b)).
for all $\mathbf{I}_{\text{inf},k}$ **do**
Get corresponding control line $\overline{\mathbf{P_i P_{i+1}}}$
Make a linear interpolation between $\xi_{\mathbf{Q}_i}$ and $\xi_{\mathbf{Q}_{i+1}}$ to get $\xi_{\mathbf{I}_{\text{inf},k}}$
end for

4. Transform spline and ray to x-axis (fig. 32 c)).
Let θ be the angle between $\mathbf{r}_{\text{inf}}$ and the x-axis
Get transformed ray $\tilde{\mathbf{r}} = \overline{\tilde{\mathbf{P}}_{\text{out}}\tilde{\mathbf{P}}}$ with $\tilde{\mathbf{P}}_{\text{out}} = \mathbf{0}$ and $\tilde{\mathbf{P}} = (\mathbf{P} - \mathbf{P}_{\text{out}})\mathbf{T}_{\text{rot}}(\theta)$
for all $\mathbf{P}_i$ **do**
Get transformed control points $\tilde{\mathbf{P}}_i = (\mathbf{P}_i - \mathbf{P}_{\text{out}})\mathbf{T}_{\text{rot}}(\theta)$
end for
5. Newton's method to find zeros of transformed spline (fig. 32 c)) .
for $k := 1$ **to** $\#\left(\xi_{\mathbf{I}_{\text{inf},k}}\right)$ **do**
Starting value $\xi_k^{j_0} = \xi_{\mathbf{I}_{\text{inf},k}}$
Initialize $|y_k^{j_0}| \overset{!}{>} \varepsilon$
while $|y_k^j| > \varepsilon$ **do**
Get point on transformed spline $\tilde{\mathbf{C}}(\xi_k^j) = \begin{pmatrix} x_k^j \\ y_k^j \end{pmatrix}$.
Get tangent vector on transformed spline $\dot{\tilde{\mathbf{C}}}(\xi_k^j)$.
Newton step $\xi_k^{j+1} = \xi_k^j - \tilde{\mathbf{C}}(\xi_k^j)/\dot{\tilde{\mathbf{C}}}(\xi_k^j)$
end while

if $x_k^{j_{end}} < \tilde{P}_x$ **then**
Append intersection point of transformed spline and finite ray $\tilde{\mathbf{I}}_{\text{fin},k} = \begin{pmatrix} x_k^{j_{end}} \\ y_k^{j_{end}} \end{pmatrix}$.
end if
end for

return $modulus\left(\frac{\#(\mathbf{I}_{\text{fin}}))}{2}\right)$
end if
end procedure

Journal publication: https://doi.org/10.1016/j.camwa.2017.01.027

Acknowledgements

The first and the last author gratefully acknowledge the financial support of the German Research Foundation (DFG) under Grant RA 624/22-1.

References

[1] J. D. Foley, A. V. Dam, S. K. Feiner, J. F. Hughes, and R. L. Phillips, *Introduction to Computer Graphics*. Addison-Wesley, 1997.

[2] A. J. P. Gomes and J. G. Teixeira, "Form feature modelling in a hybrid CSG/BRep scheme," *Computers & Graphics*, vol. 15, no. 2, pp. 217–229, 1991.

[3] J. J. Shah and M. Mäntylä, *Parametric and Feature-Based CAD/CAM: Concepts, Techniques, and Applications*. John Wiley & Sons, 1995.

[4] J. Cottrell, T. J. Hughes, and Y. Bazilevs, *Isogeometric Analysis: Toward Integration of CAD and FEA*. New York: Wiley and Sons, 2009.

[5] 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*, vol. 194, pp. 4135–4195, Oct. 2005.

[6] L. Piegl and W. Tiller, *The NURBS Book*. Monographs in Visual Communication, Berlin, Heidelberg: Springer Berlin Heidelberg, 1997.

[7] Y. Bazilevs, V. M. Calo, J. A. Cottrell, J. A. Evans, T. J. R. Hughes, S. Lipton, M. A. Scott, and T. W. Sederberg, "Isogeometric analysis using T-splines," *Computer Methods in Applied Mechanics and Engineering*, vol. 199, pp. 229–263, Jan. 2010.

[8] F. Massarwi and G. Elber, "A B-spline based framework for volumetric object modeling," *Computer-Aided Design*, vol. 78, pp. 36–47, Sept. 2016.

[9] B.-Q. Zuo, Z.-D. Huang, Y.-W. Wang, and Z.-J. Wu, "Isogeometric analysis for CSG models," *Computer Methods in Applied Mechanics and Engineering*, vol. 285, pp. 102–124, Mar. 2015. 00000.

[10] A. Patera, "Nonconforming Mortar Elements Methods: Application to Spectral Dicretizations." `http://ntrs.nasa.gov/archive/nasa/casi.ntrs.nasa.gov/19890002965.pdf`, 1988.

[11] D. Natekar, X. Zhang, and G. Subbarayan, "Constructive solid analysis: A hierarchical, geometry-based meshless analysis procedure for integrated design and analysis," *Computer-Aided Design*, vol. 36, pp. 473–486, Apr. 2004.

[12] D. Schillinger, L. Dedè, M. A. Scott, J. A. Evans, M. J. Borden, E. Rank, and T. J. 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*, vol. 249-252, pp. 116–150, 2012.

[13] J. Parvizian, A. Düster, and E. Rank, "Topology optimization using the finite cell method," *Optimization and Engineering*, vol. 13, pp. 57–78, July 2011.

[14] E. Rank, M. Ruess, S. Kollmannsberger, D. Schillinger, and A. Düster, "Geometric modeling, isogeometric analysis and the finite cell method," *Computer Methods in Applied Mechanics and Engineering*, vol. 249-252, pp. 104–115, Dec. 2012.

Journal publication: https://doi.org/10.1016/j.camwa.2017.01.027

[15] H.-J. Bungartz, M. Griebel, and C. Zenger, *Introduction to Computer Graphics*. Charles River Media, 2004.

[16] A. A. G. Requicha and H. B. Voelker, *Constructive Solid Geometry, TM-25*. Production Automation Project, University of Rochester, 1977.

[17] W. E. Lorensen and H. E. Cline, "Marching cubes: A high resolution 3D surface construction algorithm," in *Proceedings of the 14th Annual Conference on Computer Graphics and Interactive Techniques*, (New York, NY), pp. 163–169, ACM Press, 1987.

[18] 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*, vol. 197, pp. 3768–3782, Aug. 2008.

[19] T. J. R. Hughes, *The Finite Element Method: Linear Static and Dynamic Finite Element Analysis*. Mineola, NY: Dover Publications, 2000.

[20] B. D. Reddy, *Introductory Functional Analysis: With Applications to Boundary Value Problems and Finite Elements*. No. 27 in Texts in applied mathematics, New York: Springer, 1998 ed., Nov. 1997.

[21] T. J. R. Hughes, "A simple scheme for developing 'upwind' finite elements," *International Journal for Numerical Methods in Engineering*, vol. 12, pp. 1359–1365, Jan. 1978. 00258.

[22] J. Parvizian, A. Düster, and E. Rank, "Finite cell method," *Computational Mechanics*, vol. 41, pp. 121–133, Apr. 2007.

[23] M. Dauge, A. Düster, and E. Rank, "Theoretical and Numerical Investigation of the Finite Cell Method," *Journal of Scientific Computing*, vol. 65, pp. 1039–1064, Mar. 2015. 00000.

[24] B. A. Szabó, A. Düster, and E. Rank, "The p-version of the finite element method," in *Encyclopedia of Computational Mechanics* (E. Stein, ed.), Chichester, West Sussex: John Wiley & Sons, 2004.

[25] 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*, vol. 200, pp. 3358–3380, Nov. 2011.

[26] 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*, vol. 50, pp. 445–478, Feb. 2012.

[27] S. Duczek, M. Joulaian, A. Düster, and U. Gabbert, "Simulation of Lamb waves using the spectral cell method," pp. 86951U–86951U–11, Apr. 2013.

[28] M. Joulaian and A. Düster, "Local enrichment of the finite cell method for problems with material interfaces," *Computational Mechanics*, vol. 52, pp. 741–762, Oct. 2013.

[29] A. Abedian, J. Parvizian, A. Düster, H. Khademyzadeh, and E. Rank, "Performance of Different Integration Schemes in Facing Discontinuities in the Finite Cell Method," *International Journal of Computational Methods*, vol. 10, p. 1350002, June 2013. 00027.

[30] L. Kudela, N. Zander, S. Kollmannsberger, and E. Rank, "Smart octrees: Accurately integrating discontinuous functions in 3D," *Computer Methods in Applied Mechanics and Engineering*, vol. 306, pp. 406–426, July 2016.

Journal publication: https://doi.org/10.1016/j.camwa.2017.01.027

[31] S. Kollmannsberger, A. Özcan, J. Baiges, M. Ruess, E. Rank, and A. Reali, "Parameter-free, weak imposition of Dirichlet boundary conditions and coupling of trimmed and non-conforming patches," *International Journal for Numerical Methods in Engineering*, vol. 101, pp. 670–699, Mar. 2015. 00001.

[32] 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*, vol. 95, pp. 811–846, Sept. 2013.

[33] H. Whitney, "On regular closed curves in the plane," *Compositio Mathematica*, vol. 4, pp. 276–284, 1937.

[34] J. Machchhar and G. Elber, "Revisiting the problem of zeros of univariate scalar Béziers," *Computer Aided Geometric Design*, vol. 43, pp. 16–26, Mar. 2016.

[35] 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*, vol. 55, pp. 499–517, Feb. 2015.

[36] J. Schöberl, "NETGEN." `http://www.hpfem.jku.at/netgen/`, 2003.

Journal publication: https://doi.org/10.1016/j.camwa.2017.01.027

2.3 B-rep and dirty geometries

As discussed in section 2.1, CSG models are only one possible geometric representation. Alternatively, and actually most commonly, B-Rep models are used which describe volumes implicitly by means of their boundary. However, B-Rep models are prone to flaws such as gaps, multiple entities, or other inconsistencies which lead to mathematically invalid volumes. These pose serious problems for mesh generators and demand healing the flawed geometries prior to meshing. A correct healing is often a non-trivial task. Yet, under certain restrictions, it is still possible to construct a valid point membership test for flawed geometries which can be used in the framework of the FCM. Thereby, healing may be avoided.

A straight forward definition of a flawed geometry is not directly available. The following publication, therefore, takes the reciprocal approach. First, it carefully lays out what a valid B-rep description is. Then, operators are introduced which cause typical, specific flaws. Finally a point membership test is constructed which robustly delivers valid answers for geometric models to which those flaws have been introduced. This point membership test is then used in the framework of the Finite Cell Method. The methodology enables a direct computation on flawed geometric B-Rep models without the need of mesh generation or mesh healing.

Journal Publication

title:	Integrating CAD and Numerical Analysis: Dirty Geometry handling using the Finite Cell Method
authors:	B. Wassermann, S. Kollmannsberger, S. Yin, L. Kudela, E. Rank
published at:	*Computer Methods in Applied Mechanics and Engineering*
publisher:	Elsevier
year:	2019
volume:	351
pages:	808–835
doi:	https://doi.org/10.1016/j.cma.2019.04.017

Integrating CAD and Numerical Analysis: 'Dirty Geometry' handling using the Finite Cell Method

Benjamin Wassermann[*1], Stefan Kollmannsberger[†1], Shuohui Yin[‡1,2], László Kudela[§1], and Ernst Rank[¶1,3]

[1]Chair for Computation in Engineering, Technical University of Munich, Arcisstr. 21, 80333 München, Germany
[2]School of Mechanical Engineering, Xiangtan University, Hunan 411105, PR China
[3]Institute for Advanced Study, Technical University of Munich, Lichtenbergstr. 2a, 805748 Garching, Germany

April 10, 2019

Abstract

This paper proposes a computational methodology for the integration of Computer Aided Design (CAD) and the Finite Cell Method (FCM) for models with "dirty geometries". FCM, being a fictitious domain approach based on higher order finite elements, embeds the physical model into a fictitious domain, which can be discretized without having to take into account the boundary of the physical domain. The true geometry is captured by a precise numerical integration of elements cut by the boundary. Thus, an effective Point Membership Classification algorithm that determines the inside-outside state of an integration point with respect to the physical domain is a core operation in FCM. To treat also "dirty geometries", i.e. imprecise or flawed geometric models, a combination of a segment-triangle intersection algorithm and a flood fill algorithm being insensitive to most CAD model flaws is proposed to identify the affiliation of the integration points. The present method thus allows direct computations on geometrically and topologically flawed models. The potential and merit for practical applications of the proposed method is demonstrated by several numerical examples.

Keywords: Computer-Aided Design, Dirty geometry, Finite Cell Method, Flood Fill, Point Membership Classification, Flawed geometry, Geometrical flaws, Topological Flaws

Submitted to CMAME: Computer Methods in Applied Mechanics and Engineering

[*]benjamin.wassermann@tum.de, Corresponding Author
[†]stefan.kollmannsberger@tum.de
[‡]yinsh2016@163.com
[§]laszlo.kudela@tum.de
[¶]ernst.rank@tum.de

Journal publication: https://doi.org/10.1016/j.cma.2019.04.017

Contents

Journal publication: https://doi.org/10.1016/j.cma.2019.04.017

1 Introduction

Product development in the scope of Computer Aided Engineering (CAE) typically involves Computer Aided Design (CAD) and numerical analyses. The life cycle of almost every complex mechanical product starts with the creation of a CAD model which is then converted into a suitable format for downstream CAE applications such as Finite Element Analysis, Rapid Prototyping, or automated manufacturing. However, a truly smooth transition from a geometric to a computational model is still challenging. This is especially the case for numerical simulations like the Finite Element Method. Very often, complex and time-consuming model preparation and pre-processing steps are necessary to obtain a decent numerical model that is suitable for analysis purposes. For complex CAD models, this transition process can take up to 80% of the overall analysis time [1].

For this reason, various alternative numerical approaches have been developed which seek to avoid or shorten this costly transition process (e.g., meshing). Isogeometric analysis (IGA) as the most prominent example aims at easing the transition from CAD to computational analysis by using the same spline basis functions for geometric modeling and numerical simulation [1, 2]. In a related earlier approach Cirak and Scott [3] presented an integrated design process based on Subdivision Surfaces. Kagan and Fischer [4] used B-spline finite elements in an effort to join design and analysis.

However, independent of the respective numerical approach, flaws may appear in the CAD model during the design-analysis cycle – such as double entities, gaps, overlaps, intersections, and slivers – as shown in Fig. 1. They are mainly due to data loss while the model is exchanged between different CAD and/or CAE systems, to inappropriate operations by the designer, or to approximation steps resulting in incompatible geometries. These model flaws, also called *dirty topologies* or *dirty geometries*, may be extremely small or even unapparent. While they are of no particular importance to a CAD engineer they may, however, cause serious problems for structural analyses. In the best case, they merely generate excessively fine meshes in some regions which are not relevant to structural analysis (e.g., at fillets, etc.) but drive up computational time unnecessarily. In the worst case, computations fail completely because no finite element mesh can be created. This is due to the fact that neither classical finite element approaches nor the newly developed methods mentioned above are designed to handle dirty topologies and geometries. Thus, extra effort is necessary to repair, heal, or reconstruct the model into an analysis-suitable geometry [5], even if the affected region is not of special interest to the structural analyst. This is also a major obstacle for IGA, which heavily relies on flawless geometries.

Solid CAD modeling systems mainly rely on two different representation techniques: Boundary Representation (B-Rep) and Constructive Solid Geometry (CSG)[6] which is often extended to a so-called procedural modeling. In CSG, a volume is described by volumetric primitives, whereas in B-Rep it is described via its surfaces. Consequently, B-Rep models provide direct and easy access to the explicit boundaries. However, B-Rep models are not necessarily valid, meaning that it might in some cases not be possible to determine whether a point lies inside or outside (Point Membership Classification). In contrast, CSG models are inherently watertight. Problems such as non-manifolds, dangling faces, or lines, or Boolean operations on disjoint objects need to be handled accordingly by the respective CAD system. A novel representation technique – V-Rep (volumetric representation) – was recently proposed by Elber et al.[7] and

Journal publication: https://doi.org/10.1016/j.cma.2019.04.017

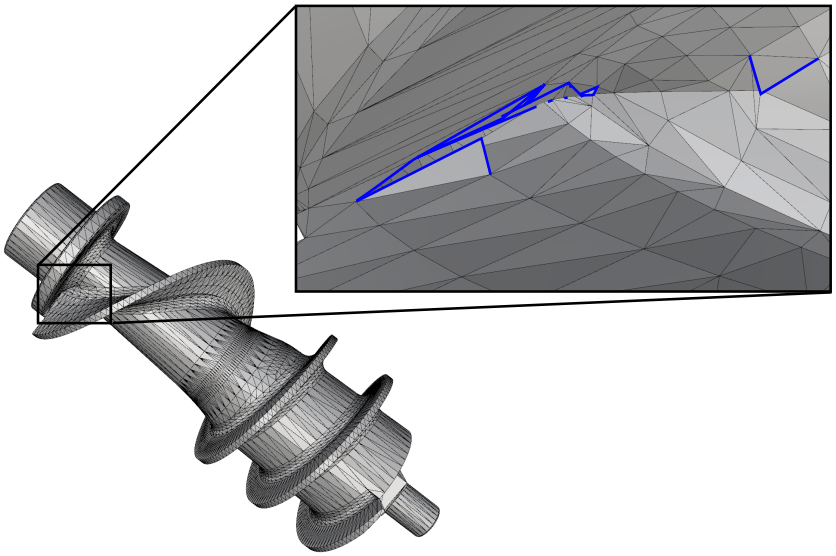


Figure 1: Cad model of a screw with flaws. Free edges are highlighted in blue.

implemented into the IRIT solid modeler[1]. V-Reps are constructed of volumetric, non-singular B-Spline primitives, thus, providing both an explicit volume and explicit surface description. As the V-Rep models follow the CSG idea of combining valid primitives, this approach can help to overcome several pitfalls in solid modeling. Within this paper, we focus on flawed or 'dirty' B-Rep models. The most direct way to address CAD model flaws is to heal or repair the model before meshing. The healing process involves identifying the type of model errors and fixing them individually. Butlin and Stops [8] listed topological and geometrical inconsistencies. The geometrical inconsistencies relate to their positions in space, while the topological inconsistencies relate to the connections or relationships among entities. Gu et al. [9] presented a visual catalog of potential flaws. Petersson and Chand [10] developed a suite of tools for the preparation of CAD geometries that are imported from IGES files and stored in the boundary representation for mesh generation; the algorithm can identify gross flaws and remove them automatically. Yang et al. [11] classified topological and geometrical flaws in CAD models and proposed a procedural method to verify 19 flaw types in STEP format and 12 types in the IGES format. Yang and Han [5] conducted a case study to investigate the typical nature of CAD model flaws. They reported the classification and frequency of each of the six most common error types that significantly increase the lead times, and they proposed a repair method based on the design history. Healing methods act either on the CAD model or on the mesh [5]. According to their approach, these methods can be classified into surface [12], volumetric [13] and hybrid [14] types. Surface-based geometry repair methods perform local modifications merging and fixing incorrect surface patches. Volumetric techniques are used to reconstruct a new global shape without flaws. However, this approach typically leads to information loss, especially at sharp features

[1]http://www.cs.technion.ac.il/ gershon/GuIrit/

Journal publication: https://doi.org/10.1016/j.cma.2019.04.017

such as kinks. Hybrid methods combine the advantages of local surface healing and global volumetric healing. To this end, flaws are detected and a volumetric reconstruction is performed only in their vicinity. These methods have been used for CAD models that are represented in typical B-Rep formats (e.g., STEP and IGES) as well as for polygonal meshes [15].

Although healing and repair methods have been applied successfully in recent years, healing can still be very labor intensive and time consuming in the scope of product development. As a remedy, mesh generation techniques have been developed which have the potential to generate meshes from flawed geometric models. In this line of research, Wang and Srinivasan [16] proposed an adaptive Cartesian mesh generation method. Herein, the computational grid is created inside the domain, which then connects to the boundary. Another technique – the Cartesian shrink-wrapping technique – was presented in [17] to generate triangular surface meshes automatically for 3D flawed geometries without healing. However, to generate a mesh, an initial watertight shell (called wrapper surface) needs to be constructed. Another line of research proposed by Gasparini et al. [18] is an approach to analyze geometrically imperfect models based on a geometrically adaptive integration technique that uses different model representations, i.e. space decomposition, B-Rep, and distance fields. This approach relies on a method that was first introduced by Kantorovich [19] and that has recently been commercialized [20]. Furthermore, this approach requires computation of a well-defined distance function to the boundaries – which is non-trivial for dirty geometries, as the orientation of boundary surfaces might be incorrect or the location of the boundaries is anticipated incorrectly, e.g., due to spurious entities, or intersections. However, two main issues arise applying geometry healing: (i) In the case that the geometry is healed locally, i.e. each flaw on its own, it is almost impossible to heal all flaws. Hence, a subsequent volumetric meshing is likely to fail. (ii) If the model is healed in a volumetric sense, i.e. the model is entirely reconstructed, a valid model can be obtained. However, typically sharp features, such as edges, corners or small details are lost. The automatic assumptions which are made during the volumetric healing lead to a changed model which is likely to be not in the designer's intent.

In this work, we present an alternative computational methodology which aims at dealing *robustly* with dirty topologies and geometries. At its core, it utilizes the Finite Cell Method (FCM) [21, 22], a fictitious domain method which uses classical linear, or higher-order finite elements. The FCM embeds the physical model into a fictitious domain which is then discretized by a simple, often axis-aligned grid. This grid does not have to conform to the boundary of the physical domain. Instead, the physical domain is recovered on the level of integration of element matrices and load vectors. A Point Membership Classification (PMC) test is carried out at each integration point to determine whether it lies inside or outside the physical domain. Hence, the only information needed from the CAD model is a reliable and robust PMC, which strongly reduces the geometrical and topological requirements on the validity of the geometric model. This observation allows for a new paradigm in the computational analysis: not to create an analysis-suitable model and/or to derive a mesh or distance field, but rather *to directly compute on geometrically and/or topologically flawed models by a flaw-insensitive computational method*. Thus, it is neither required to heal the flawed geometry nor to construct conforming meshes or distance fields. Instead, a PMC is constructed which is robust w.r.t. to a large number of model flaws. The Point Membership Classification test can then be evaluated with a certainty at least up to a geometric magnitude of the defect itself (as, e.g., in the case of gaps). This is

Journal publication: https://doi.org/10.1016/j.cma.2019.04.017

important because a subsequent computational analysis can then directly be carried out without healing. Moreover, the computational analysis may still deliver the necessary accuracy on those flawed models as their effect on the results of the computation remains local to the flaw itself. Only, if the local flaw lies directly in the region of interest it must be fixed. This is, however, only necessary to achieve higher accuracy – an analysis can be carried out either way.

The Finite Cell Method is a widely applicable method itself. While the original publications concerning the FCM treated linear elasticity in 2D and 3D [22], the scope of application was extended to various fields, such as elastoplasticity [23], constructive solid geometric models [24], topology optimization [25, 26], local enrichment for material interfaces [27], elastodynamics and wave propagation [28, 29, 30], and contact problems [31, 32]. Further developments include weakly enforced essential boundary conditions [33], local refinement schemes [34], and efficient integration techniques [35, 36, 37, 38]. Furthermore, the concept of the FCM is independent of the underlying approximation method. It does not have to be based on hierarchical Legendre shape functions but can also be built on a spline-based approximation like in Isogeometric Analysis, or spectral shape functions [39]. In this case, the fictitious domain approach is an adequate method for trimming Isogeometric Analysis, as presented and analyzed, e.g., in [40, 41, 42]. In [43], an efficient method to overcome the inherent problem of bad condition numbers based on precondition is presented. Approaches very similar to the FCM have been presented more recently, like the cutFEM method [44], which builds on earlier publications of Hansbo et al. [45]. Therein, small elements are explicitly stabilized by controlling the gradients across embedded boundaries connected neighboring cells in the fictitious domain. This is different to FCM where a stabilization is achieved to a certain extent by a small but non-zero stiffness in the fictitious domain.

In this contribution, the FCM is extended in order to directly simulate a CAD model with flaws. The paper is structured as follows: Section 2 provides a brief overview over geometrical and topological flaws. The basic formulation of the FCM and the requirements of a numerical simulation on flawed geometric CAD models are given in Section 3. A robust algorithm for Point Membership Classification on dirty geometries is presented in Section 4. Several numerical examples for the proposed methodology are presented and discussed in Section 5. Finally, conclusions are drawn in Section 6.

2 Dirty Topology/Geometry

In this section, we provide a very short general overview of Boundary Representation (B-Rep) models (sec. 2.1) and necessary conditions for their validity (sec. 2.2). By implication, 'dirty' geometries, or topologies are models which do not meet these requirements and are therefore mathematically invalid. To describe the wide variety of different flaws (sec: 2.3), we define mathematical operators (sec. 2.4) and apply them to a valid B-Rep model, thereby transforming a 'valid' into a 'dirty' B-Rep model (sec. 2.5).
Several of these flaw operators allow introducing a control parameter ε, indicating a geometric size of the respective flaws. Applying a sequence of flaw operators maps a flawless model to exactly one resulting flawed model. It is obvious that, given some flawed model, it is not possible to determine on which flawless model it could be based meaning that a class of equivalent

Journal publication: https://doi.org/10.1016/j.cma.2019.04.017

flawless models can be associated to one 'dirty' model. Our conceptual approach therefore only *assumes* the existence of a flawless model that is expected to be 'close' to the 'dirty' one. This is used as the geometric basis for analysis. Further, it is to be noted that no explicit knowledge of this flawless model is required.

2.1 Boundary Representation Models

B-Rep objects are described by their boundaries. A model Ω can consist of several sub-domains, which all describe a separate closed volumetric body B_i.

$$\Omega = \{ B_i \mid i \in \{1, ..., n\} \} \tag{1}$$

with n being the number of volumetric bodies. For simplicity of presentation, we assume that a B-Rep model consists only of one domain $\Omega = B$. A B-Rep body consists of topology T and geometry G [6]:

$$B(T, G) \tag{2}$$

The topology T describes the relations or logical location of all entities (2.1.1), whereas the geometry G provides the physical location of points, consequently defining the actual shape of the model (2.1.2).

2.1.1 Topology

The topology $T(t, r^{int}, r^{ext})$ provides the logical internal $r^{int} = \{r_i^{int}\}$ and external relations $r^{ext} = \{r_i^{ext}\}$ between the topological entities $t = \{t_i\}$, i.e. vertices v_i, edges e_j, and faces f_k. Thereby, each topological entity t_i has its own local, internal relation r_i^{int}, defining how and from which underlying topological entities it is constructed. Topological entities are typically represented by sets:

$$V = \{ v_i \mid i \in \{1, ..., n\} \} \tag{3}$$

$$E = \{ e_i \mid i \in \{1, ..., m\}, \; e_i = (v_\alpha, v_\beta), \; v_\alpha, v_\beta \in V \} \tag{4}$$

$$F = \{ f_i \mid i \in \{1, ..., o\}, \; f_i = \left((e_\kappa)_{\kappa \in \{\alpha, ..., \psi\}}, \boldsymbol{n}_i \right), \; e_\kappa \in E \} \tag{5}$$

with n, m, o being the number of vertices, edges, and faces, respectively. The ordered pair of vertices (v_α, v_β) contains the bounding vertices of an edge. A face f_i is described by an ordered pair containing: (i) the boundary edges, denoted by an ordered n-tuple (e_κ), with n being the number of boundary edges, and (ii) the respective normal vector $\boldsymbol{n}_i$. In some cases, the normal vector is provided implicitly by the order of the boundary edges (e_κ). The external relations r^{ext} describe the global adjacency relations between the particular entities (e.g., which faces are neighbors to each other). There are various possible methods to represent the internal and external adjacency relations, or a combination of both, such as the *winged edge model* or the *double connected edge list* [6]. Thereby, the adjacency relations can be represented by graphs. Figure 2 shows an exemplary detail of a topology consisting of three triangles. The pure external relations can, for example, be represented by the adjacency matrix r_{FF}^{ext} (see equation (6)). The

Journal publication: https://doi.org/10.1016/j.cma.2019.04.017

adjacency matrices for faces and edges r_{FE} (see equation (7)) and for edges and vertices r_{VE} (see equation (8)) represent a combination of internal and external relations [2].

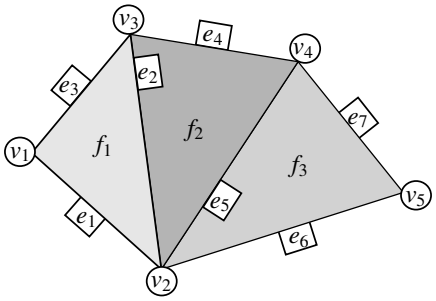


Figure 2: Example topology with $n = 5$ vertices, $m = 7$ edges, and $o = 3$ faces.

$$r_{FF}^{ext} \subset F \times F = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix} \tag{6}$$

$$r_{FE} \subset F \times E = \begin{bmatrix} 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 \end{bmatrix} \tag{7}$$

$$r_{VE} \subset V \times E = \begin{bmatrix} 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 \end{bmatrix} \tag{8}$$

2.1.2 Geometry

The geometry $G(\{g_i\})$ contains the geometric entities g_i, i.e. the points $\boldsymbol{P}_i$, curves $\boldsymbol{C}_j(\xi)$, and surfaces $\boldsymbol{S}_k(\xi, \eta)$, which describe the actual physical location of the boundary and, thus, the shape of the geometry. Curves and surfaces are often expressed in parametric representation:

$$\boldsymbol{P}_i = (\, x_i,\, y_i,\, z_i \,)^{\mathrm{T}} \tag{9}$$

$$\boldsymbol{C}_i(\xi) = \begin{pmatrix} x(\,\xi\,) \\ y(\,\xi\,) \\ z(\,\xi\,) \end{pmatrix} \quad e.g., \quad \boldsymbol{C}_i(\xi) = \sum_{j}^{n_{Q_j}} N_j(\,\xi\,) \cdot \boldsymbol{Q}_j \tag{10}$$

$$\boldsymbol{S}_i(\boldsymbol{\xi}) = \begin{pmatrix} x(\,\boldsymbol{\xi}\,) \\ y(\,\boldsymbol{\xi}\,) \\ z(\,\boldsymbol{\xi}\,) \end{pmatrix} \quad e.g., \quad \boldsymbol{S}_i(\boldsymbol{\xi}) = \sum_{j}^{n_{Q_j}} \sum_{k}^{n_{Q_k}} N_j(\,\xi\,) \cdot N_k(\,\eta\,) \cdot \boldsymbol{Q}_{j,k} \tag{11}$$

with $\xi \in \mathbb{R}$ and $\boldsymbol{\xi} = (\xi, \eta) \in \mathbb{R}^2$. $N_i(\xi)$ denote shape functions (such as Lagrange or Legendre polynomials, B-Splines, NURBS, etc.) and $\boldsymbol{Q}_i$ the associated (control-)points, which can, depending on the curve description, coincide with the geometrical points $\boldsymbol{P}_i$.
Analogous to the topology, the geometry G can be represented by sets:

$$P = \{\, \boldsymbol{P}_i \;\mid\; i = \{1, ..., n\} \,\} \tag{12}$$

$$C = \{\, \boldsymbol{C}_i \;\mid\; i = \{1, ..., 2 \cdot m\} \,\} \tag{13}$$

$$S = \{\, \boldsymbol{S}_i \;\mid\; i = \{1, ..., o\} \,\} \tag{14}$$

[2] *Please note:* An entry 1 in the adjacency matrix shows which entity (row) is connected to which other entity (column). An entry 0 indicates that no direct adjacency exists.

Journal publication: https://doi.org/10.1016/j.cma.2019.04.017

where the number of points and surfaces equals the number of vertices n and faces o, respectively. A special case are curves, where at each edge two adjoined faces meet, whose underlying surfaces have each their own boundary curves. Consequently, the number of curves is $2 \cdot m$ (see Fig. 3).

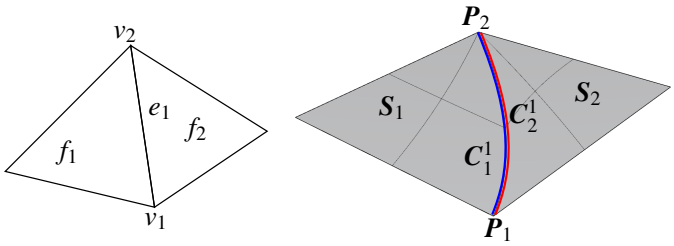


Figure 3: At each edge e_i two boundary curves $\boldsymbol{C}_{S_j}^{e_i}$ and $\boldsymbol{C}_{S_k}^{e_i}$ meet.

2.1.3 Minimal B-Rep and the STL format

The most commonly used B-Rep exchange format between CAD and analysis is STL (STereoLithography, or more expressive Standard Tessellation Language). STL can be interpreted as a minimal B-Rep format, as it provides only the least amount of necessary information. Additionally, no explicit separation between topology and geometry is made. STL consists of independent triangles, which are defined by their three corner points. As geometric information in form of point coordinates is provided explicitly only for vertices, curves and surfaces are linearly interpolated. No adjacency, or 'consistency' information is provided, which makes STL quite flexible – but also particularly prone to a variety of potential flaws. The relation between faces and vertices reads:

$$F^{STL} = \{ f_i \mid i \in \{1, ..., o\}, f_i = \left((v_\alpha, v_\beta, v_\gamma), \boldsymbol{t}_i\right), v_\kappa \in V, |V| = 3 \cdot o \} \tag{15}$$

Note that – due to the multiple definition of vertices – STL models are, strictly speaking, topologically not valid. Furthermore, the redundancy of point definitions and normal vectors, which could be derived from the orientation of the face has an eminent impact on the required memory for storage.

2.2 Conditions for valid B-Rep models

Although intuitively quite apparent, it is not straightforward to define a valid B-Rep model. Patrikalakis et al. [46] provided a definition: "A B-Rep model is valid if its faces form an orientable 2-manifold without boundary." From this, several requirements can be derived, some of which are also mentioned by Mäntylä [6] and Hoffmann [47].
Topology:

1. Different vertices do have different coordinates (see Fig. 6a).
2. One edge is shared by exactly two faces (see Fig. 4).

Journal publication: https://doi.org/10.1016/j.cma.2019.04.017

3. Faces at one vertex belong to one surface, i.e. at a vertex it is possible to cycle through all adjacent faces such that all of the vertex' edges are crossed exactly once (see Fig. 4).

4. The orientation of faces must follow *Moebius' Rule*, i.e. inside and outside must be distinguishable from each other (see Fig. 6d).

Geometry:

5. A curve must lie on the respective surface whose partial boundary it forms.

6. Both boundary curves at one edge must coincide (see Fig. 7a).

7. Surfaces must not self-intersect. From this – and from 5 – it follows that curves do not self-intersect either (see Fig. 7b).

8. Surfaces must not touch or intersect with other surfaces except at common edges (see Fig. 7c).

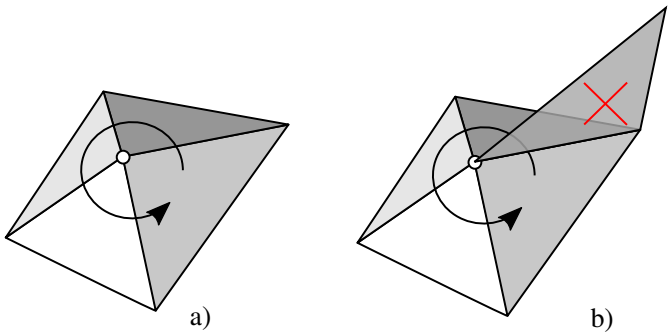


Figure 4: Vertex with adjoined edges and faces: (a) It is possible to cycle through the faces, passing each adjoined edge once. Hence, all faces belong to the same surface. (b) Not all faces belong to the same surface.

2.3 CAD model flaws

Model flaws can originate from different sources, such as mathematical inaccuracies, data conversion problems between different software systems, mistakes by designers, different design goals, etc. The probably most famous example of mathematical inaccuracies is the 'leaking teapot' model, as depicted in Fig. 5. The gap between spout and body of the teapot could only be avoided by more complex spline types (see, e.g., T-splines [48]), or unreasonably high polynomial degrees. The simplification results in a non-watertight geometry, a major obstacle for the interoperability between CAD and CAE. Figures 6, 7 and 8 provide an overview over the most common topological and geometrical modeling flaws.

Journal publication: https://doi.org/10.1016/j.cma.2019.04.017

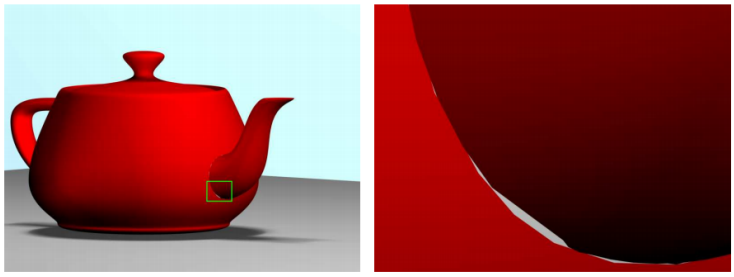
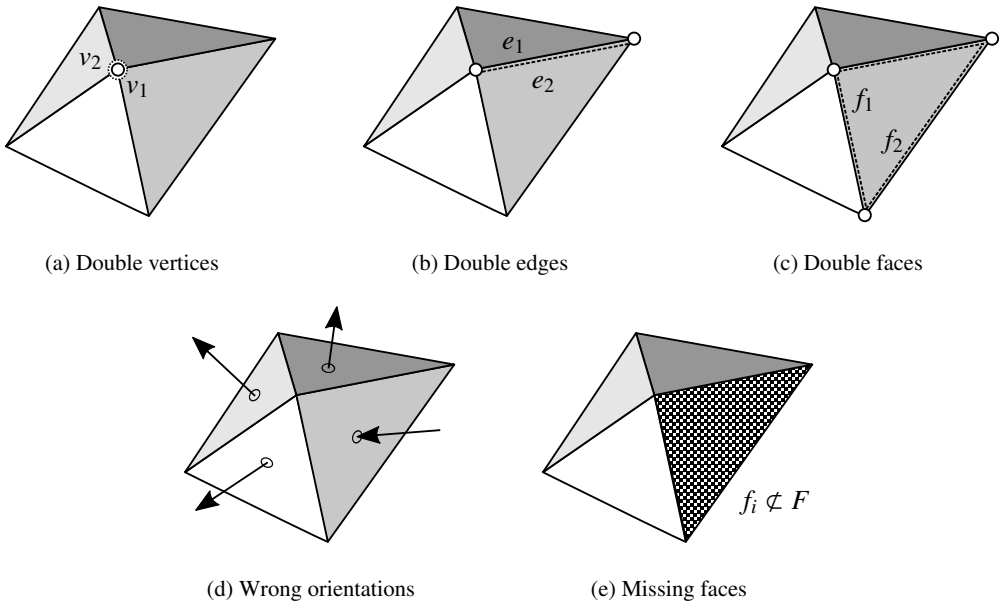


Figure 5: Gaps between trimmed NURBS patches of the Utah teapot (Picture taken from [49])

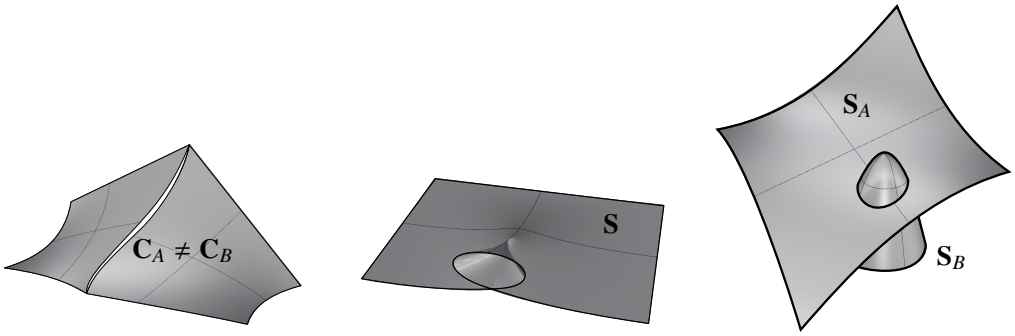


(a) Double vertices (b) Double edges (c) Double faces

(d) Wrong orientations (e) Missing faces

Figure 6: Topological flaws

Journal publication: https://doi.org/10.1016/j.cma.2019.04.017

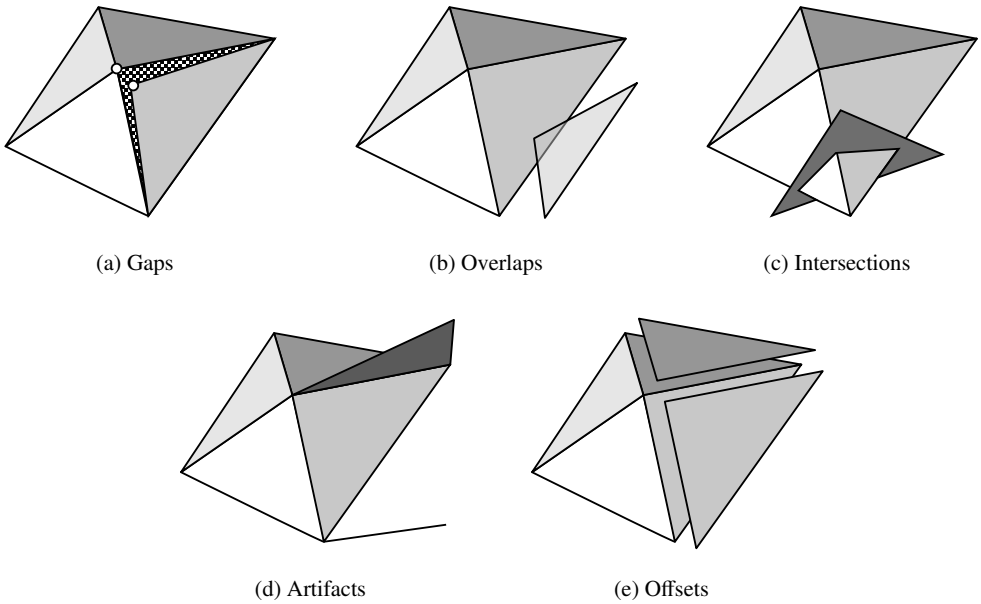


(a) Curves at common edge do not coincide

(b) Surface and boundary curve self-intersect

(c) Surface intersects with another surface

Figure 7: Geometrical flaws



(a) Gaps

(b) Overlaps

(c) Intersections

(d) Artifacts

(e) Offsets

Figure 8: Hybrid flaws which consist of topological and geometrical components

Journal publication: https://doi.org/10.1016/j.cma.2019.04.017

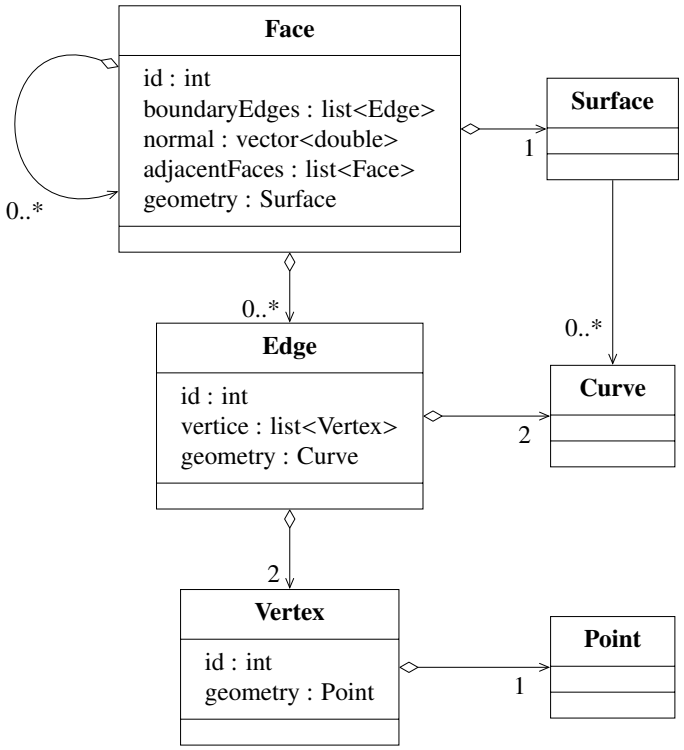


Figure 9: UML-diagram of a possible object-oriented B-Rep implementation

2.4 Flaw operators

In the following, we will introduce several operators that perform transformations on a valid B-Rep model, allowing for a controlled imposition of different flaws. To measure the size of the flaws, we introduce an error parameter ε, indicating the 'dirtiness' or inaccuracy of the model.

To provide an easily understandable formulation of the operators, we consider an object-oriented B-Rep data structure. Thereby, the implementation must allow a distinction between internal and external/adjacency relations. Figure 9 provides a UML diagram of a possible hierarchical implementation. For an introduction to the notation of the UML (Unified Modeling Language) see, e.g., [50]. Here, the external adjacency relations r^{ext} are realized at the faces, where the adjacent faces are stored in the field: *adjacentFaces*. All other external adjacency relations (e.g., which edges are adjacent) can be derived from this and from the respective internal relations r^{int}.

Let ω^{t_i} be the B-Rep sub-part, or segment, which corresponds to a topological entity t_i, e.g., a face, an edge, or a vertex. The segment ω^{t_i} consists of all information that is needed to visualize t_i. Hence, it must contain t_i and, recursively, all underlying sub-topologies and geometries that are related by respective internal adjacency relations r^{int} (see Fig. 10).

$$\omega^{t_i}(T^{t_i}, G^{t_i}) \subset B \tag{16}$$

Journal publication: https://doi.org/10.1016/j.cma.2019.04.017

with $T^{t_i}\left(\tau, \rho^{int}\right)$ and $G^{t_i}(\gamma)$ being the respective topology and geometry, where $\tau = \{\tau_i\}$ and $\gamma = \{\gamma_i\}$ denote the sets of those topological and geometrical entities which are recursively related by the internal relations $\rho^{int} = \{\rho_i^{int}\}$. Consequently, three different segments are possible: 1) vertex segments, b) edge segments, and c) face segments. As topological entities $\{\tau_i\}$, a face segment, for example, contains the face itself and all associated edges and vertices. As geometric entities $\{\gamma_j\}$, it holds the corresponding surface with its boundary curves and corner points. Additionally, all internal relations $\{\rho_i^{int}\}$ are contained, i.e. the relations among face, boundary edges, and vertices, as well as the relations to the geometric entities. Not contained are *external* adjacency relations, i.e. those to neighboring faces or edges.

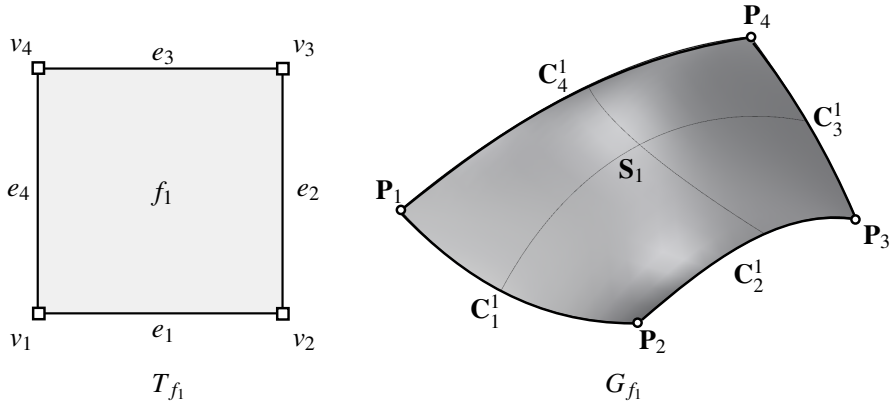


Figure 10: B-Rep sub-part ω^{f_1}, which corresponds to face f_1 and consists of the topology T^{f_1} and the corresponding geometry G^{f_1}.

In the following, $\tilde{a}$ denotes the object a after the transformation and let $\text{dist}\,(a, b) = \inf\{\,\|a, b\|_2\,\}$ be the minimum Euclidean distance between two objects, e.g., the distance between the closest points on two different surfaces.

1. Let O^{select} be a extraction operator, which selects for a topological entity t_i the corresponding segment ω^{t_i} from the body B.

$$O^{select}\ (B\,,\ t_i) \mapsto \omega^{t_i} \tag{17}$$

   Note that the external relations are not extracted. Hence, the segment forgets about its *logical* location in the body.

2. Let O^{join} be a join operator that adds a segment ω^{t_i} to the body B.

$$O^{join}\left(B\,,\ \omega^{t_i}\right) \mapsto \tilde{B}\,,\quad \text{where}\quad \tilde{B} = B \cup \omega^{t_i} \tag{18}$$

3. Let $O^{shallowCopy}$ be a shallow copy operator that copies an arbitrary entity a_i. For a more detailed description of the object-oriented concept of 'shallow' and 'deep' copying see,

Journal publication: https://doi.org/10.1016/j.cma.2019.04.017

e.g., [51]. a_i can be a topological entity t_i, a geometrical entity g_i, or an internal r_i^{int} or external relation r_i^{ext}.

$$O^{shallowCopy}\,(a_i) \mapsto \tilde{a}_i\,, \quad \text{where} \quad \tilde{a}_i := a_i \tag{19}$$

The ':=' in (19) is to be understood as the shallow copy assignment, according to [51]. Note that the new object is distinguishable from the old object, e.g., by an updated id, or, in the context of object-oriented programming, by a different memory address. Yet, it still uses the same references to other objects as the original segment.

4. Let $O^{deepCopy}$ be an internal deep copy operator [51] that performs a deep copy operation on a segment ω^{t_i}. To this end, a shallow copy operation is carried out on all corresponding topological and geometrical entities, as well as the internal adjacency relations.

$$\begin{aligned} O^{deepCopy}\left(\omega^{t_i}\right) \mapsto \tilde{\omega}^{t_i}(\tilde{T}^{t_i}, \tilde{G}^{t_i})\,, \quad \text{with} \quad & \tilde{T}^{t_i}\left(\tilde{\tau}, \tilde{\rho}^{int}\right)\,,\ \tilde{\tau}_i := O^{shallowCopy}(\tau_i)\,, \\ & \tilde{\rho}_i^{int} := O^{shallowCopy}(\rho_i^{int})\ \ \forall \tau_i, \rho_i^{int} \in T^{t_i}\,, \\ \text{and} \quad & \\ & \tilde{G}^{t_i}\,(\tilde{\gamma})\ ,\ \tilde{\gamma}_i := O^{shallowCopy}(\gamma_i)\ \ \forall \gamma_i \in G^{t_i} \end{aligned} \tag{20}$$

Note that the deep copied segment ω^{t_i} has no information about its *logical* location in B, i.e. it has no external adjacency relations, and that all internal relations are updated to reference the new topological and geometrical entities.

5. Let O^{delete} be a deletion operator that deletes a face f_i and its related geometry $g^{f_i} \subset G$ consisting of the underlying surface $\boldsymbol{S}_i$ and the corresponding boundary curves $\{\boldsymbol{C}_j^{\boldsymbol{S}_i}\}$. Thereby, the characteristic size of the resulting opening must not exceed a given, e.g., user-defined minimal accuracy ε. Let δ be the diameter of the largest possible inscribed sphere of the surface $\boldsymbol{S}_i$ to be deleted.

$$O^{delete}\left(B, f_i, g^{f_i}\right) \mapsto \tilde{B}\,(\tilde{T}, \tilde{G})\,, \quad \text{where} \quad \tilde{F} = F \setminus f_i\ ;\ \tilde{G} = G \setminus g^{f_i}\ ,\ \delta < \varepsilon \tag{21}$$

Note that, as a deletion of an edge, or vertex would lead to an uncontrollable cascade of deletions of superior entities, only a face deletion is allowed in this context. A B-Rep model with a deleted edge or node without deletion of referencing faces would not even be readable and is not considered in our investigation.

6. Let $O^{explode}$ be an operator that removes all external relations r^{ext} from a body B. This can be achieved by extracting (17), copying (20), and joining (18) all face segments $\omega_{f_i}\ \forall f_i \in F$. The resulting body $\tilde{B}$ is then described by independent topological sup-parts/segments ω_{f_i}.

$$O^{explode}\,(B) = O^{join}\left(B^{\circ},\, O^{deepCopy}\left(O^{extract}\,(B,\, F)\right)\right) \mapsto \tilde{B}\,, \quad \text{where} \quad \tilde{r}^{ext} = \emptyset \tag{22}$$

where B° is an empty body.

Journal publication: https://doi.org/10.1016/j.cma.2019.04.017

7. Let O^{flip} be a topological flip operator that flips the normal $\boldsymbol{n}_i$ of a face f_i.

$$O^{flip}\,(f_i) \mapsto \tilde{f}_i\,, \quad \text{where} \quad \tilde{f}_i = ((e_\kappa), \tilde{\boldsymbol{n}}_i = -1 \cdot \boldsymbol{n}_i) \tag{23}$$

8. Let O^{move} be a geometric move operation that moves the point $\boldsymbol{P}_i$ within the range ε. Additionally, all adjoined surfaces and curves are adapted consistently such that they form a 2-manifold without boundaries after the operation. This involves the following adaptions to the adjoined surfaces $S^{\boldsymbol{P}_i} = \{\boldsymbol{S}_i^{\boldsymbol{P}_i}\}$ and curves $C^{\boldsymbol{P}_i} = \{\boldsymbol{C}_i^{\boldsymbol{P}_i}\}$:

   - The resulting point $\tilde{\boldsymbol{P}}_i$ must again lie on the altered surfaces $\tilde{S}^{\tilde{\boldsymbol{P}}_i}$ and curves $\tilde{C}^{\tilde{\boldsymbol{P}}_i}$.
   - All pairs of the resulting adjoined surfaces $(\tilde{\boldsymbol{S}}_A^{e_k}, \tilde{\boldsymbol{S}}_B^{e_k})$ must again meet at their common edge/curve e_k. Consequently, the two respective boundary curves $(\tilde{\boldsymbol{C}}_A^{e_k}, \tilde{\boldsymbol{C}}_B^{e_k})$ must coincide.

   The latter condition is omitted in the case of an already broken topology, where an edge no longer has two adjoined faces/surfaces.

$$\begin{aligned} &O^{move}\,(\boldsymbol{P}_i, G) \mapsto \tilde{G}\,, \quad \text{where} \quad 0 < \operatorname{dist}\left(\boldsymbol{P}_i, \tilde{\boldsymbol{P}}_i\right) < \varepsilon\,, \\ &\quad \text{and} \\ &\quad \exists\, \boldsymbol{\xi} : \tilde{\boldsymbol{S}}_i^{\tilde{\boldsymbol{P}}_i}(\boldsymbol{\xi}) = \tilde{\boldsymbol{P}}_i \;\; \forall\, \tilde{\boldsymbol{S}}_i^{\tilde{\boldsymbol{P}}_i}\,, \; \exists\, \zeta : \tilde{\boldsymbol{C}}_i(\zeta) = \tilde{\boldsymbol{P}}_i \;\; \forall\, \tilde{\boldsymbol{C}}_i^{\tilde{\boldsymbol{P}}_i}\,, \\ &\quad \text{and} \\ &\quad \operatorname{dist}\left(\tilde{\boldsymbol{C}}_A^{e_k}, \tilde{\boldsymbol{C}}_B^{e_k}\right) = 0 \;\; \forall\, (\tilde{\boldsymbol{S}}_A^{e_k}, \tilde{\boldsymbol{S}}_B^{e_k}) \text{ at } v_i \end{aligned} \tag{24}$$

9. Let O^{detach} be a geometrical operator that detaches two adjacent surfaces, $\boldsymbol{S}_i$ and $\boldsymbol{S}_j$, which meet at the edge e_k (see fig. 7a). To this end, one surface $\boldsymbol{S}_i$ and its respective boundary curve $\boldsymbol{C}_{\boldsymbol{S}_i}^{e_k}$ at e_k are changed. Again, the characteristic size of the potentially resulting opening must not exceed ε.

$$\begin{aligned} O^{detach}\,(G, e_k) \mapsto\; \tilde{G}\,, \quad \text{where} \quad &\operatorname{dist}\left(\tilde{\boldsymbol{S}}_i, \tilde{\boldsymbol{C}}_{\boldsymbol{S}_i}^{e_k}(\xi)\right) = 0\,, \\ &0 \le \operatorname{dist}\left(\boldsymbol{C}_{\boldsymbol{S}_i}^{e_k}, \tilde{\boldsymbol{C}}_{\boldsymbol{S}_i}^{e_k}(\xi)\right) < \varepsilon \;\; \forall \xi \in [\xi_a, \xi_b] \end{aligned} \tag{25}$$

   with $[\xi_a, \xi_b]$ being the respective interval on which the boundary curve is defined.

10. Let $O^{intersect}$ be a geometric operator that alters a surface $\boldsymbol{S}_i(\boldsymbol{\xi})$ such that it touches or intersects with another surface $\boldsymbol{S}_j(\boldsymbol{\eta})$ apart from common edges. Note that we assume that there is no intersection in the original model, according to the definition of a valid B-Rep model.

$$\begin{aligned} &O^{intersect}\,(\boldsymbol{S}_i(\boldsymbol{\xi})) \mapsto\; \tilde{\boldsymbol{S}}_i(\boldsymbol{\xi})\,, \quad \text{where} \\ &\exists\, (\boldsymbol{\xi}, \boldsymbol{\eta}) \;:\; \operatorname{dist}\left(\tilde{\boldsymbol{S}}_i(\boldsymbol{\xi}), \boldsymbol{S}_j(\boldsymbol{\eta})\right) = 0 \;\wedge\; \operatorname{dist}\left(\tilde{\boldsymbol{S}}_i(\boldsymbol{\xi}), \tilde{\boldsymbol{C}}_k^{\tilde{\boldsymbol{S}}_i}\right) > 0 \\ &\forall\, \tilde{\boldsymbol{C}}_k^{\tilde{\boldsymbol{S}}_i} \in \Gamma^{\tilde{\boldsymbol{S}}_i}\,,\; i \neq j \end{aligned} \tag{26}$$

   with $\Gamma^{\tilde{\boldsymbol{S}}_i}$ being the set of boundary curves of $\tilde{\boldsymbol{S}}_i$.

Journal publication: https://doi.org/10.1016/j.cma.2019.04.017

A special case of intersections are self- intersections:

$$O^{selfIntersect}\left(\boldsymbol{S}_i(\boldsymbol{\xi})\right) \mapsto \tilde{\boldsymbol{S}}_i(\boldsymbol{\xi})\,, \quad \text{where} \quad \exists\,(\boldsymbol{\xi},\boldsymbol{\eta}) \;:\; \operatorname{dist}\left(\tilde{\boldsymbol{S}}_i(\boldsymbol{\xi}), \boldsymbol{S}_i(\boldsymbol{\eta})\right) = 0\,,\; \boldsymbol{\xi} \neq \boldsymbol{\eta} \tag{27}$$

2.5 Application of flaw operators

We now continue with the definition of a flawed model. To this end, we apply the flaw operators defined in section 2.4 onto a valid B-Rep model. The 'dirtiness' of the model is then defined by ε. It should be mentioned that, for models that are drafted by a real-life CAD system, flaws do not necessarily originate from these operators, yet most flawed models can equivalently be created by a sequence of these operators.

Let $B(T,G)$ be a valid flawless B-Rep body. Note that operators acting on the body is to be understood as acting on a segment ω^{t_i}, or single topological, or geometrical entity, or relation.

1. Single topological entities t_i and their corresponding segments ω_{t_i} can be copied and added to B with a combination of the extraction (17), the deep copying (20), and the joining (18) operator:

$$\tilde{B}\left(\tilde{T},\tilde{G}\right) := O^{join}\left(B\left(T,G\right),\, O^{deepCopy}\left(O^{extract}\left(B\left(T,G\right),\, t_i\right)\right)\right) \tag{28}$$

   The resulting B-Rep model is invalid as it has multiple entities (refer to Figs. 6a, 6b, and 6c), which violates condition 1. As an example, consider the STL format where each triangle (re-)defines its corner points. Also, multiply defined faces/surfaces appear frequently in free form CAD models, which leads to a touching/intersection of the surfaces (refer to Fig. 8c).

2. Application of the deletion operator (21) on a face f_i:

$$\tilde{B}\left(\tilde{T},\tilde{G}\right) := O^{delete}\left(B, f_i, g_{f_i}\right) \tag{29}$$

   The deletion of a face violates condition 2 (see Fig. 6e). Thereby, the size of the resulting opening restricted to be smaller than ε.

3. Application of the explosion operator (22):

$$\tilde{B}\left(\tilde{T},\tilde{G}\right) := O^{explode}(B(T,G)) \tag{30}$$

   Most B-Rep models are constructed from independent surfaces, which are later joined into a (hopefully) valid B-Rep model. This join operation corresponds to the inverse of the explosion operation. It is yet well known that a strict 'join'-operation is not necessarily possible (or maybe not feasible) e.g., in case of an intersection of two NURBS surfaces [49]. Also, STL models are constructed by independent triangles. Such models violate the topological conditions 1, 2, and 3. Geometrically, they can still form a closed 2-manifold without boundaries. However, these models are very prone to a variety of different flaws, as no external adjacency relations are provided explicitly.

Journal publication: https://doi.org/10.1016/j.cma.2019.04.017

4. Application of the flip operator (23):

$$\tilde{B}\,(\tilde{T}, G) := O^{flip}(B(T, G)) \tag{31}$$

The resulting B-Rep model does not fulfill *Moebius' Rule* anymore (see condition 4). This flaw usually appears if the normal is defined implicitly by the order of the boundary edges (see Fig. 6d). However, this error also appears quite frequently if the normal is given explicitly, e.g., in the case of STL.

5. Application of the move operator (24):

$$\tilde{B}\,(T, \tilde{G}) := O^{move}(B(T, G)) \tag{32}$$

Applied on a valid B-Rep body, the move operator preserves a geometric *2-manifold, without boundary*. However, the orientability can be lost (see condition:4). As an example, consider a point $\boldsymbol{P}_i$ on surface $\boldsymbol{S}_j$, which is close to surface $\boldsymbol{S}_k$ with distance $\mathrm{dist}\,(\boldsymbol{P}_i, \boldsymbol{S}_k) < \varepsilon$. A movement then can lead to an intersection of the two surfaces. This violates condition 8 (see Fig. 7c).

6. Application of the detach operator (25):

$$\tilde{B}\,(T, \tilde{G}) := O^{detach}(B(T, G)) \tag{33}$$

The resulting model violates condition 6. This is likely to happen at the intersection of free-form surfaces. The boundary curves would require unreasonably high polynomial degrees to perfectly coincide. Possible flaws can e.g., be openings or intersections (see Figs. 7a and 7c). As an example, consider the leaking Utah teapot.

7. Application of the intersection operator (26):

$$\tilde{B}\,(T, \tilde{G}) := O^{intersect}(B(T, G)) \tag{34}$$

The resulting model may violate conditions 7 or 8. Apart from gaps, intersections frequently appear at patch boundaries as well (see Fig. 7a). Intersections can also occur if two surfaces are too close to each other. In this case, they additionally violate *Moebius' Rule* 4 (see Figs. 7c and 7b). A special case are overlaps, where two surfaces touch each other (see Fig. 8b).

8. Application of the copy (20) and the move operators (24) to a single face f_i :

$$\begin{aligned} \tilde{\omega}^{f_i} &:= O^{deepCopy}\left(O^{extract}\,(B,\ f_i)\right) \\ \breve{\omega}^{f_i} &:= O^{move}(\boldsymbol{P}_j \in \tilde{\omega}^{f_i},\ \tilde{G}^{t_i}) \\ \tilde{B}\,(\tilde{T}, \tilde{G}) &:= O^{join}\left(B,\ \breve{\omega}^{f_i}\right) \end{aligned} \tag{35}$$

This chain of operations allows to create offsets and artifacts, i.e. entities which do not belong to the outer hull and lead to a violation of the conditions 3 and 2 (see Figs. 8d and

Journal publication: https://doi.org/10.1016/j.cma.2019.04.017

8e).

9. Application of the explosion (22) and move operators (24):

$$\begin{aligned} \tilde{B}(\tilde{T}, \tilde{G}) &:= O^{explode}(B) \\ \check{B}(\tilde{T}, \check{G}) &:= O^{move}(\tilde{\boldsymbol{P}}_i \in \tilde{G},\, \tilde{G}) \end{aligned} \tag{36}$$

Starting from an exploded model, moving one or more points can lead to various common flaws – such as gaps, intersections, or overlaps (see Figs. 8a, 8c, 8b). As many B-Rep modeling tools work with exploded models, i.e. with independent surfaces, these flaws appear very commonly, particularly at patch boundaries. Also, the STL format stores a body with independent triangles.

Note that, independent of the performed operations, it is imperative for the presented method that the size of all openings and gaps is restricted to be smaller than a pre-defined ε. This is required not only for each individual flaw operation but also for the resulting model after a sequence of flaw operations, e.g., a sequence of individual moves of a segment.

The resulting flawed models are invalid in a mathematical sense, which renders a subsequent conversion into a simulation model either impossible or invalid. The necessity to heal the flaws can neither be circumvented by meshing, as in the classical FEM, nor by a direct simulation as in IGA. It is, however, possible to compute 'dirty' models directly with an *embedded domain method* such as the Finite Cell Method (Section 3). To this end, we construct a specially adapted Point Membership Classification test (Section 4) which is blind to flaws up to a characteristic size ε.

3 Finite Cell Method

The Finite Cell Method is a higher order fictitious domain method. However, the approach presented within this paper does not rely on higher-order elements. Hence, it can be also applicable for linear fictitious domain methods. FCM offers simple meshing of potentially complex domains into a structured grid of, e.g., cuboid cells without compromising the accuracy of the underlying numerical method. For completeness of this paper, the basic concepts are briefly introduced in this section. We restrict ourselves to linear elasticity – emphasizing however, that the FCM has been extended to more general partial differential equations [52, 53, 54, 55].

3.1 Basic formulation

In the Finite Cell Method, an n-dimensional open and bounded physical domain Ω_{phy} is embedded in a fictitious domain Ω_{fict} to form an extended domain $\Omega_{\cup}$, as illustrated in Fig. 11 in two dimensions. The resulting domain $\Omega_{\cup}$ has a simple shape which can be meshed easily, without conforming to the boundary of Ω_{phy}.

The weak form of the equilibrium equation for the extended domain $\Omega_{\cup}$ is defined as

$$\int_{\Omega_{\cup}} [\mathbf{Lv}]^T \alpha \mathbf{C} [\mathbf{Lu}] \, \mathrm{d}\Omega = \int_{\Omega_{\cup}} \mathbf{v}^T \alpha \mathbf{f} \, \mathrm{d}\Omega + \int_{\Gamma_N} \mathbf{v}^T \bar{\mathbf{t}} \, \mathrm{d}\Gamma \quad , \tag{37}$$

Journal publication: https://doi.org/10.1016/j.cma.2019.04.017

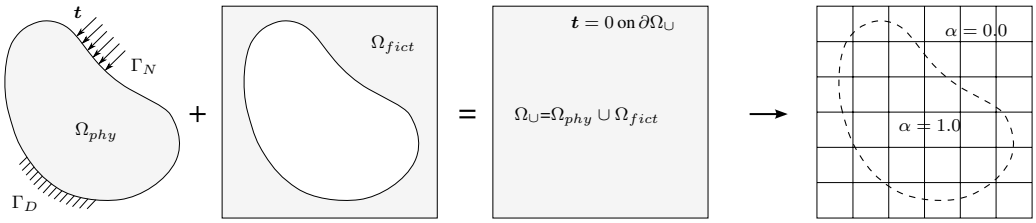


Figure 11: The concept of the Finite Cell Method [56]

where $\mathbf{u}$ is a displacement function, $\mathbf{v}$ a test function, $\mathbf{L}$ is the linear strain operator, and $\mathbf{C}$ denotes the elasticity matrix of the physical domain Ω_{phy}, yet extended to $\Omega_\cup$. $\mathbf{f}$ and $\bar{\mathbf{t}}$ denote the body load and the prescribed tractions on the Neumann boundary, respectively. The indicator function α is defined as

$$\alpha(\mathbf{x}) = \begin{cases} 1 & \forall \mathbf{x} \in \Omega_{phy} \\ 10^{-q} & \forall \mathbf{x} \in \Omega_{fict} \end{cases} , \tag{38}$$

In the limiting case of $q \to \infty$, the standard weak form for an elasticity problem on Ω_{phys} is obtained. In practical applications, a sufficiently large $q = 6..10$ (see [21, 22]) is chosen, introducing a modeling error to the formulation [57], which yet stabilizes the numerical scheme and controls the conditioning number of the discrete equation system – see [43] for a detailed analysis. Ω_{fict} is then discretized in 'finite cells' of simple shape (rectangles or cuboids). In the context of this paper, we assume for simplicity a uniform grid of finite cells, yet note that generalizations to locally refined grids [58, 59] and unstructured meshes [60, 61, 62] have been studied extensively.

3.2 Geometry treatment

In FCM, the physical domain Ω_{phys} (i.e. the geometry) is recovered by the discontinuous scalar field α. Consequently, the complexity of the geometry is shifted from the finite elements to the integration of the element matrices and load vectors, which imposes less geometrical requirements on the model. It is in fact sufficient to provide a robust Point Membership Classification (PMC), i.e. for every point $x \in \mathbb{R}^n$, it must be possible to decide whether it is inside or outside of Ω_{phys}. This implies that Ω_{phys} must have a mathematically valid description. Due to the discontinuity of α, the integrands in cut cells need to be computed by specially constructed quadrature rules, see, e.g., [35, 38, 63] for a recent overview of possible schemes. To perform a suitable integration, the domain is approximated by a space-tree TR_{int}. The leaves of TR_{int} are called integration leaves c_{int}. Additional information, such as explicit surface descriptions are only needed for the application of boundary conditions as well as for post-processing (see Sec. 3.3).

Journal publication: https://doi.org/10.1016/j.cma.2019.04.017

3.3 Boundary conditions

Neumann boundary conditions are applied according to equation (37) in an integral sense on the boundary Γ_N. Homogeneous Neumann conditions (i.e. zero traction) require no treatment, as they are automatically satisfied by setting $\alpha = 0$ or, in an approximate sense, to a small value in Ω_{fict}. As the boundary of the physical model typically does not coincide with the edges/faces of the finite cell mesh, Dirichlet boundary conditions need to be enforced also in a weak sense. To this end, several methods have been adopted, such as the penalty method, Nitsche's method, or Lagrange Multipliers [33, 64, 65, 66].
For the integration of Dirichlet and inhomogeneous Neumann boundary conditions, an explicit surface description is needed. This can be of poor quality. For the enforcement of Neumann boundary conditions, however, a surface without multiple faces/surfaces or large overlaps is required, as these flaws would introduce physically modified boundary conditions (i.e. additional loads, heat sources, etc.). To this end, we propose the following automatable method to convert a 'dirty' surface into a surface without multiple entities or overlaps:

1. Triangulate the respective surface (if not already provided, e.g., with STL).
2. Get the intersection points between the surface mesh and the element boundaries.
3. Create an element-wise point cloud from the intersection points and respective triangle corner points.
4. Perform an element-wise Delaunay triangulation on the respective point cloud.

The resulting element-wise triangular meshes are used only for integration and can consequently be independent of each other. Note that the requirements to these local surface meshes are by far less restrictive than they would be for a surface mesh as a starting point for volume mesh generation. Note that a potential triangulation of the surface will cause an approximation error.

4 Robust Point Membership Classification for flawed CAD models

As explained in Section 3, the only geometric information required to setup the system matrices for the Finite Cell Method is an unambiguous statement about the location of a point, i.e whether it lies inside or outside of the domain of computation. Considering flawed CAD models (e.g., with undesired openings), the concept of 'inside' or 'outside' is fuzzy – at least up to the characteristic size of the flaw ε. In this section, we present a robust Point Membership Classification method for 'dirty' STL B-Rep models. The presented approach is, however, not restricted to STL models, and it can easily be extended to other boundary representations.

4.1 Point Membership Classification for valid CAD models

PMC algorithms are fundamental and extensively used operations, e.g., in computer graphics, computer games, and in geoinformatics [67]. For different geometric representations, various

Journal publication: https://doi.org/10.1016/j.cma.2019.04.017

PMC algorithms exist. For CSG models, a point is classified against all the underlying primitives and the resulting boolean expressions (see [24]). Ray casting [67] is often used for boundary representation models. Further variants are approximation-tree-based algorithms [68], point cloud methods [69], sign of offset [70], and the swath method [71]. As the space-tree based approximation and the ray-casting are needed in the following, these aspects will be explained in more detail:

- *Ray casting:* The ray casting method is an efficient and suitable algorithm for general polytopes, and it is extensively used in computational graphics, e.g., for depth maps. To classify a given point with respect to a geometric model, a ray is shot in an arbitrary direction and the intersections with the boundary are counted. The parity (even, or odd) of intersections then provides information on whether the point lies inside or outside. For flawless models, ray-casting is accurate. For flawed CAD models, however, ray casting delivers no reliable statement about the point's domain membership, as almost all flaws influence the parity of intersections

- *Space-tree based PMC:* For the tree-based PMC, the domain is discretized by a space-tree TR_{int}, with leaves c_{geo}. Leaves intersected by the surface are marked as cut. Subsequently, a flood-fill algorithm is applied to the leaves c_{geo}. Starting from a seed point, whose domain membership is known, all connected leaves are marked as inside or outside, respectively. A challenge in this methodology is posed only by undesired openings or unintentional gaps. In these cases, a too fine approximation with leaves smaller than the size of the flaws would cause the flood-fill algorithm to mark the entire domain as inside or outside. Furthermore, despite its robustness against most flaws, the octree TR_{int} gives only a coarse step-wise approximation of the geometry.

4.2 General approach for flawed models

The presented PMC method combines the robustness of space-tree approximation with the accuracy of ray-casting. The general approach works as follows:

1. The CAD model is approximated by a watertight space-tree TR_{geo}. Watertightness is imperative to ensure that the subsequent flood-fill can distinguish between inside and outside.

2. A flood fill algorithm is applied on TR_{geo} to mark all connected points as inside and outside, respectively. This yields a filled space tree $\widehat{TR}_{geo}$. Remark: For all points that are not on cut leaves, the approximation tree $\widehat{TR}_{geo}$ can be used as fast, efficient, and accurate PMC.

3. An additional ray-casting is only carried out for points lying inside the cut boundary leaves – in order to approximate the structure more precisely.

Step 1 and Step 3 will now be described in more detail. For a description of the well-known flood fill algorithm in step 2 we refer to, e.g., [72].

Journal publication: https://doi.org/10.1016/j.cma.2019.04.017

4.3 Watertight space tree approximation

To ensure that the approximation space-tree $\widehat{TR}_{geo}$ is watertight, the size of the smallest leafs $d_{c_{geo}}$ must not undercut the characteristic size of the largest gap/opening ε_{gap}.

$$d_{c_{geo}} > \varepsilon_{gap} \tag{39}$$

ε_{gap} is typically not known apriori and is determined by an iterative decrease of the cell size, until the subsequent fill algorithm fills the entire domain. From this, it follows that the maximal partitioning depth n_{max} of $\widehat{TR}_{geo}$ is bounded by the ratio of domain size d_{domain} of the tree $\widehat{TR}_{geo}$ to the dimension of the gaps/openings ε_{gap}:

$$n_{max} < \log_2\left(\frac{d_{domain}}{\varepsilon_{gap}}\right) \tag{40}$$

This limitation might allow, depending on the size of the gaps/openings only a very coarse approximation of the true geometry (see Figure 25). Concerning all other types of considered flaws, a test using the space tree $\widehat{TR}_{geo}$ is robust. Note that the reconstruction tree can be set up for an arbitrary flaw size ε_{gap}, as long as at least one inner cell can be detected. The quality of the result will then only be dependent on the secondary PMC test (see Section 4.4).

Note that, generally, the space-trees TR_{int} and $\widehat{TR}_{geo}$ are distinct. While TR_{int} is constructed in order to numerically integrate the discontinuous element matrices for finite cells (see Section 3), the purpose of $\widehat{TR}_{geo}$ is merely to support the Point Membership Classification of the integration points.

After the surface is approximated by the space tree, the flood fill algorithm [72] can be applied to mark connected regions. Figure 13 shows the octree approximation of a simple example (Fig. 12), which has several typical flaws. The size of the opening ε_{gap} allows a maximum subdivision level of $n_{max} = 7$. Hence, the ratio of the largest gap to overall size is in the range of:

$$\frac{1}{256} < \frac{\varepsilon_{gap}}{d_{domain}} < \frac{1}{128} \quad . \tag{41}$$

Journal publication: https://doi.org/10.1016/j.cma.2019.04.017

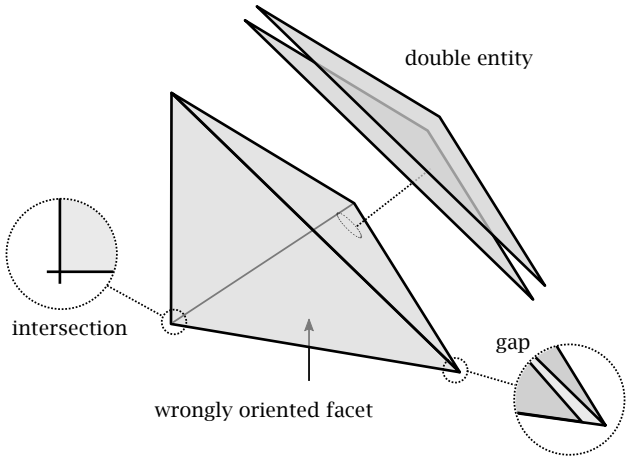
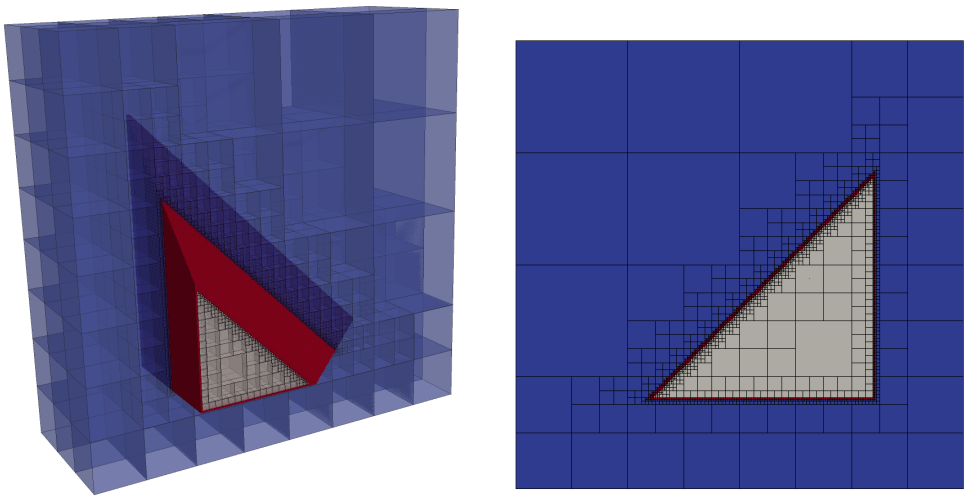


Figure 12: Example of an STL model with typical flaws.



(a) Part of a 3D octree approximation (b) 2D slice

Figure 13: Octree approximation of the embedded tetrahedral domain. The outer domain (blue) is separated by the cut leaves (red) from the inner domain (grey). The subdivision level is $n_{max} = 7$.

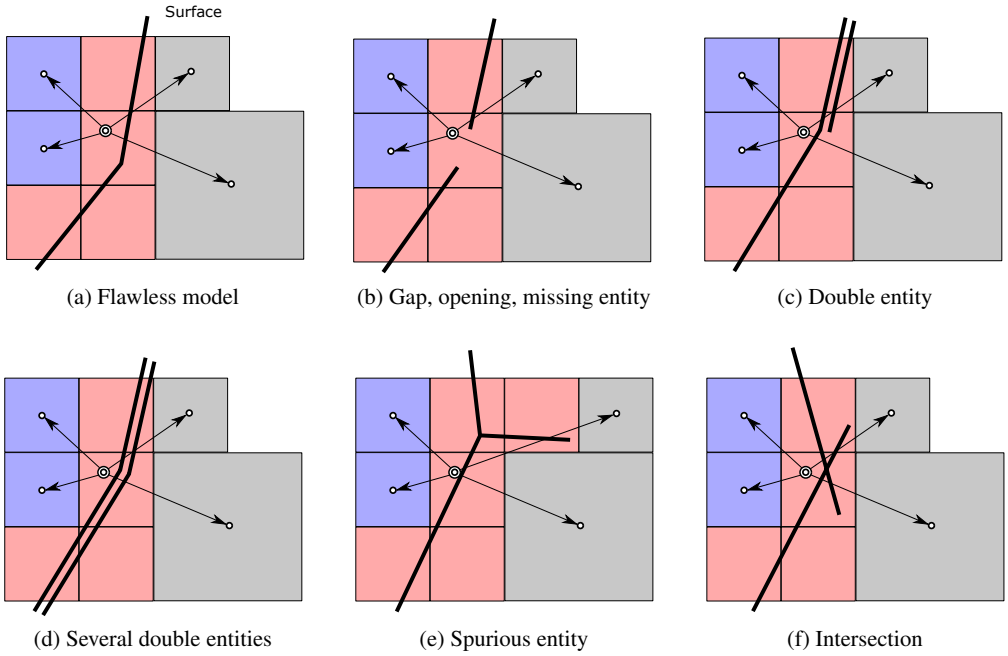
4.4 Point Membership Classification on cut leaves

The space-tree $\widehat{TR}_{geo}$ represents the surface only very roughly and, thus, cannot be used for a precise numerical analysis. Hence, in order to improve the representation of the boundary, an additional PMC using ray casting is carried out on cut leaves. Let us first assume that the model is flawless (see Fig. 14a). Then, the ray test for any integration point in an integration leaf c_{int}

Journal publication: https://doi.org/10.1016/j.cma.2019.04.017

yields a unique result without ambiguity, independent of the direction of the ray. In case of a flawed surface, the result may be ambiguous, depending on the selected direction of the ray (Figs. 14b-f). To handle this problem, we test rays in different directions, more precisely to the midpoints of all neighboring non-cut cells, which restricts the intersection tests to be carried out in the vicinity of the integration point and guarantees that various directions are queried. Hence, the probability for a correct result is increased. The PMC is then decided 'following the vote of the majority'. Clearly, this 'vote' can be wrong w.r.t. the (in general unknown) flawless model. This wrong decision results in an integration error for the computation of element matrices. In a mathematical sense, we are performing a 'variational crime' (see, e.g., [73]). For geometrically small flaws, the smallness of this integration error can be readily assumed – as, by construction of the two-stage PMC, it can only occur in the smallest leaf c_{geo} cut by the surface.

We can even bound this error by bracketing, i.e. by solving the elasticity problem (23) – once under the assumption that all ambiguous integration points are inside, and once assuming them outside of the domain of computation (see Section 4.5 and Example 5.1), thus ensuring that the approximation quality of the method is not corrupted.



(a) Flawless model (b) Gap, opening, missing entity (c) Double entity

(d) Several double entities (e) Spurious entity (f) Intersection

Figure 14: Multiple ray casting for different flaws (red: cut leafs, grey: inside, blue: outside). In these examples d) and f) would lead to indifferent results.

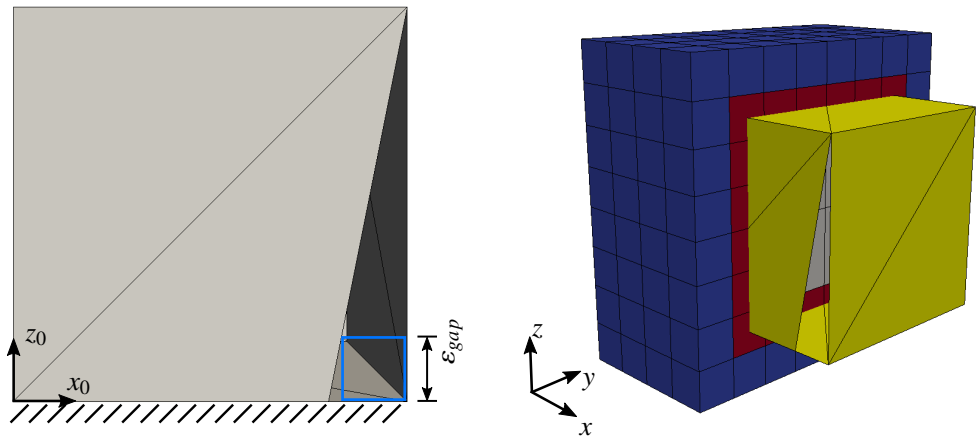
Note that also other possibilities for the secondary PMC test can be applied, such as ray-casting in only a few, or just one direction, which will lead to a significant speedup but increases the probability of wrong results. Another possibility lies in the combination with a PMC test based on point clouds, as this test is sensitive to other types of flaws, such as wrongly oriented normals,

Journal publication: https://doi.org/10.1016/j.cma.2019.04.017

or intersections.

4.5 Parameter study on the influence of the gap size

As stated in Section 2.5 a flawed model has, in general, no mathematically valid solution. Therefore it is not possible to define an 'error' of the computed approximation w.r.t. an exact solution. Yet, in order to judge the quality we compare for a simple example energies of approximate and reference solution in dependence of the size of a flaw in the B-Rep model. In particular, we investigate the influence of the largest gap size ε_{gap} on the internal strain energy for a cube with the dimensions $1 \times 1 \times 1$ loaded under self-weight. The cube is clamped at the bottom. It is embedded in $9 \times 9 \times 9$ elements employing integrated Legendre polynomials of degree $p = 3$. The B-Rep model of the cube consists of twelve triangles. One triangle is not properly connected to two of its neighbors resulting in a flawed model with a gap of characteristic size ε^i_{gap} (see Figure 15). The size of the gap limits the maximum subdivision depth of the reconstruction tree, meaning that more refined trees would lead to a non-watertight boundary of the tree (see Section 4.3).



(a) (b)

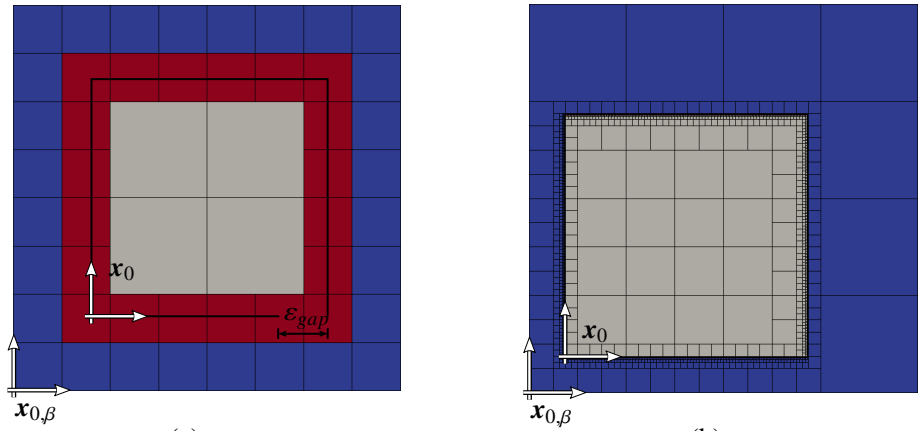
Figure 15: Parameter study on a unit cube: a) Characteristic gap size ε_{gap}. b) Reconstruction tree on the flawed geometry.

The embedding domain used for the reconstruction tree has the dimension $1.6 \times 1.6 \times 1.6$. The quality of reconstruction not only depends on the depth of the tree but also on the relative position of the domain of computation (the cube) and the tree. This influence is studied by gradually 'shifting' the origin $x_{0,beta}$ of the cube along a diagonal in space:

$$\boldsymbol{x}_{0,\beta} = \begin{bmatrix} -0.3 \\ -0.3 \\ -0.3 \end{bmatrix} + \beta \cdot \begin{bmatrix} 0.05 \\ 0.05 \\ 0.05 \end{bmatrix} \quad , \tag{42}$$

Journal publication: https://doi.org/10.1016/j.cma.2019.04.017

with $\beta = 0...3$. Figure 16 shows two different reconstruction trees for different origin positions.



(a) (b)

Figure 16: Cut through two reconstruction trees for different gap sizes, consequently maximum subdivision depths and for different origin positions: a) $\varepsilon_{gap} = 0.2$, $n_{max} = 3$, shift by $\beta = 0$. b) $\varepsilon_{gap} = 0.0031$, $n_{max} = 9$, shift by $\beta = 3$.

Figure 17 shows the influence of the characteristic size of the gap ε_{gap} on the error in the internal energy. The abscissa depicts the characteristic size of the gap compared to the unit length of the cube in percent. The values correspond to the respective maximum subdivision depths $n^i_{max} = 9...3$ resulting from gap sizes $\varepsilon^i_{gap} = \frac{1.6}{2^{n^i_{max}}}$ from left to right. The ordinate shows the deviation of the internal strain energy U to the reference energy U_{ref} in percent. The reference energy U_{ref} is computed on a flawless model. Accurate results in energy are obtained even for large gap sizes of up to 20% of the domain length. The quality of the solution is confirmed by Figure 18, showing a plot of principle stresses of the reference solutions and approximate solutions for two gap sizes.

Journal publication: https://doi.org/10.1016/j.cma.2019.04.017

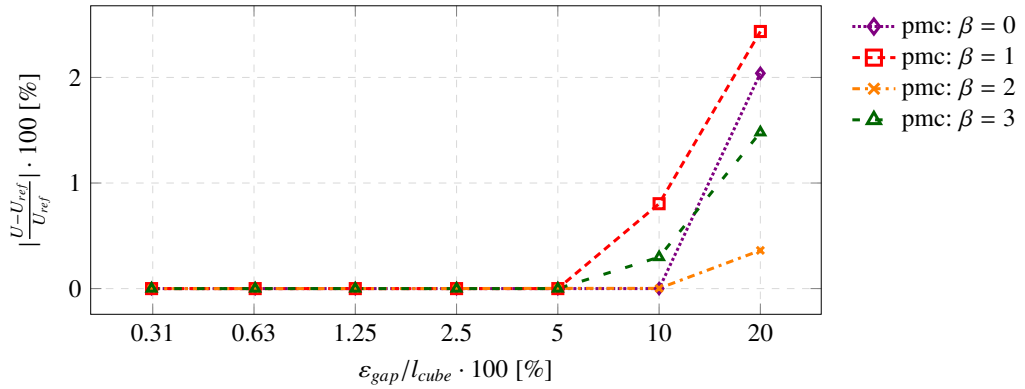
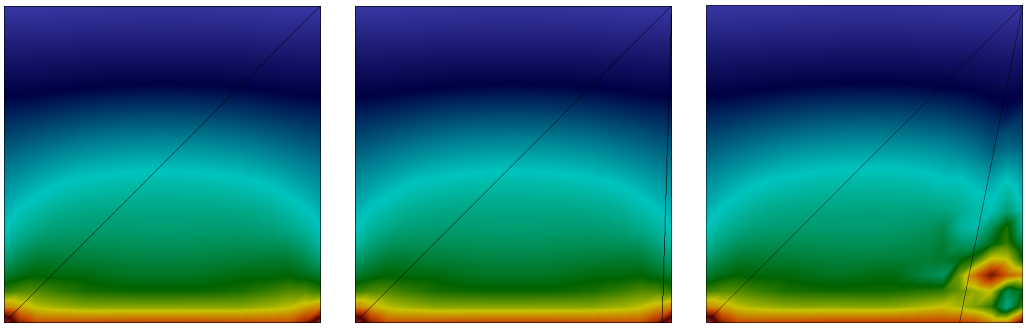


Figure 17: Relative deviation of energies depending on the gap size for different positions of the cube.



(a) $\varepsilon_{gap} = 0$, $n_{max} = \infty$ (b) $\varepsilon_{gap} = 0.025$, $n_{max} = 6$ (c) $\varepsilon_{gap} = 0.2$, $n_{max} = 3$

Figure 18: Principal stresses for the flawless model (a) and gap sizes of $\varepsilon_{gap} = 2.5\%$ (b) and $\varepsilon_{gap} = 20\%$ (c).

Although this study supports the quality of the presented approach, it cannot guarantee limitation of an error in energy of even of local solution quantities in general situations. They strongly depend on the complexity of the model and the amount and type of flaws. A crucial factor is also the location of the flaw. If it is located in highly stressed regions, the influence will be bigger than if it were located in regions of low stress. It remains to an engineer to judge the feasibility of the solution.

5 Numerical examples

To demonstrate the accuracy and robustness of the proposed approach, three examples are presented. The first simple example serves to verify the proposed method. To this end, a plate with a hole is simulated and compared to a flawless reference solution. The complex screw in the second example proves the applicability for sophisticated, defective CAD models. Again,

Journal publication: https://doi.org/10.1016/j.cma.2019.04.017

a flawless reference model was available. The last example is an engine bracket taken directly from engineering practice. This model is a perfect example of a flawed geometry, as many NURBS-patches do not fit together. An attempt to mesh the model showed that 337.544 triangles had a free edge, i.e. are flawed.

5.1 Example 1: Thick-walled plate with circular hole

As a classical benchmark for 3D problems, we choose the thick-walled plate with four circular holes [21]. The Young's modulus is set to $E = 10000.0\,N/mm^2$ and the Poisson's ratio to $\nu = 0.30$. The plate is loaded with a surface traction $\bar{t}_n = 100.0\,N/mm^2$. Symmetry boundary conditions are used, allowing to simulate only a quarter of the domain (see Fig. 19). The dimensions of the model are $b = h = 4.0\,mm$ and $t = r = 1.0\,mm$. To show the robustness of the proposed method, several flaws – namely intersections, gaps, double entities, and offsets – are introduced on the surfaces (see Figs. 20 and 21).

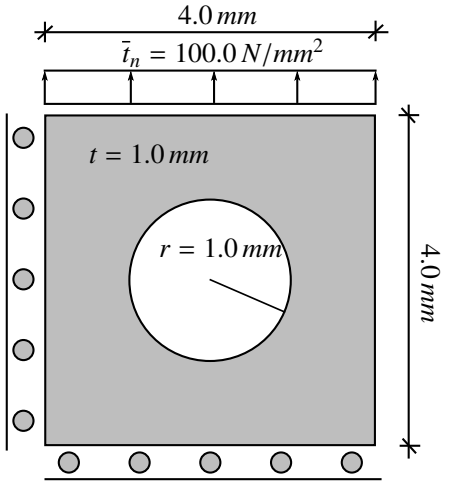


Figure 19: Thick-walled plate with circular hole under surface load

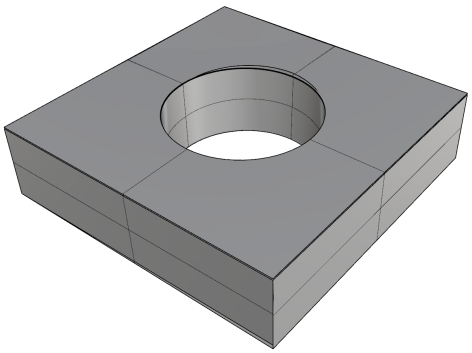
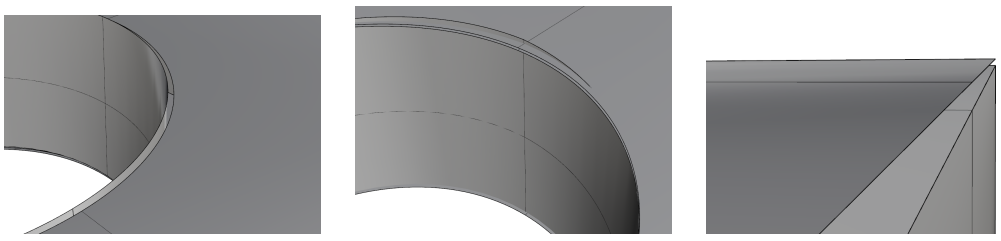


Figure 20: Flawed B-Rep model containing several gaps, intersections, offsets, and multiple entities



(a) Gap (b) Intersection (c) Offset

Figure 21: Flaw details: a) gap b) intersection c) offset of the top surface

Journal publication: https://doi.org/10.1016/j.cma.2019.04.017

The domain is discretized into $10 \times 10 \times 1$ finite cells employing integrated Legendre polynomials as basis functions. The background grid and the qualitative displacement are depicted in Figure 22. A convergence study was carried out for p-refinement using $p = 1...6$. To measure the accuracy of the approach, the strain energy is computed and passed on to the reference solution u_{ex}, which was computed with an extensive boundary-conforming finite element analysis. The minimal size of the cells $\hat{c}_{geo}$ of the geometric tree $\widehat{TR}_{geo}$ was limited by the size of the largest gap, allowing a maximum subdivision depth of $n_{max} = 5$. Note that the tree is also refined at the 'flat' surfaces of the plate. This results in ~1.33 million cells, of which ~1.15 million cells are located on the deepest level (see Fig. 23).

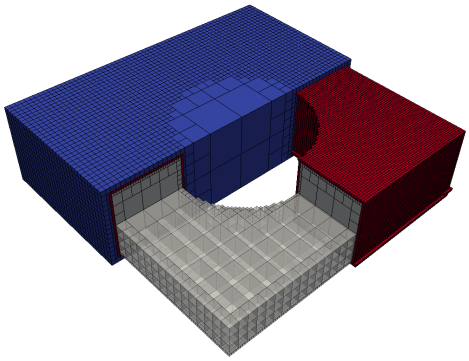


Figure 22: Approximation tree $\widehat{TR}_{geo}$ with subdivision depth $n_{max} = 5$

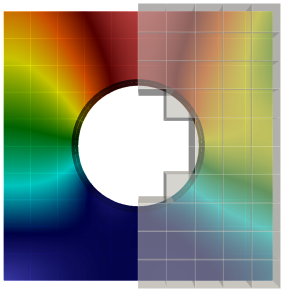


Figure 23: Displacement and finite cell discretization

Figure 24 plots the strain energy for the different polynomial degrees. Note that the relative error in the strain energy can only be computed for the valid model, as this is the only possible basis to compute a reference solution.

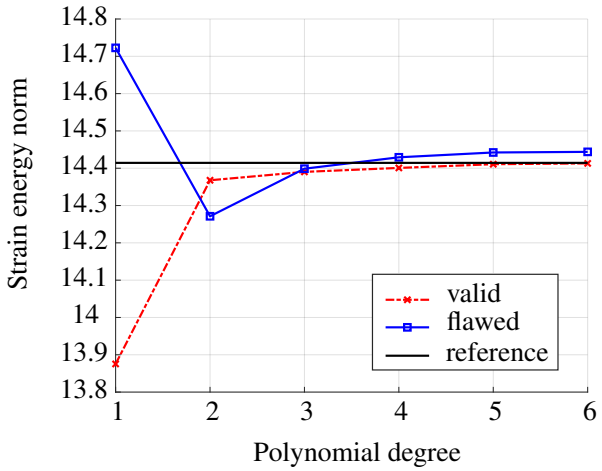


Figure 24: Strain energy norm for polynomial degrees $p = 1...6$

Journal publication: https://doi.org/10.1016/j.cma.2019.04.017

In Figure 24, it can be seen, that both models converge to a slightly different value. This is, of course, to be expected – as both models have a slightly different shape and volume. The good convergence of the flawed model is attributed to the fact that errors due to flaws are very localized. This is an inherent property of the proposed methodology.

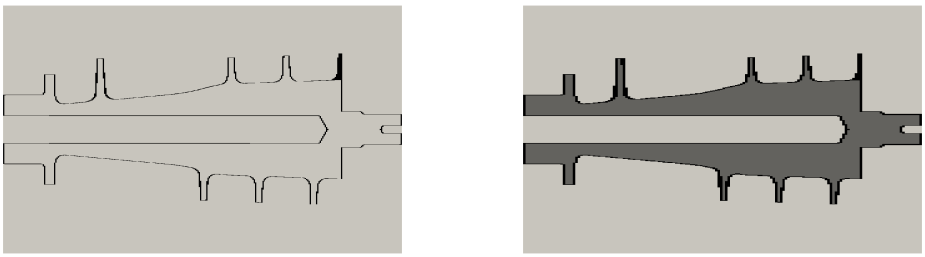
A detailed investigation of the flawed geometry can be carried out based on the ray-casting tests, which are applied on the integration cells (see Section 4.4). As an example, we consider a polynomial degree of $p = 3$. For the integration of the system matrices, 6 406 920 points need to be evaluated. From these, a total number of 2 225 641 points ($\sim$ 35%) are lying on cut cells. Typically, 12 to 18 ray-castings are carried out on each of these points. 1 503 636 points ($\sim$23%) are ambiguous, i.e. at least one ray delivers a different result compared to the majority. In 656 009 cases ($\sim$10%), a 'vote for the majority' is not possible, as the number of rays voting for inside and outside is equal. This large amount is mainly due to the many double entities. To compute the upper and lower boundaries of the energy norm, two additional simulations were carried out – once with all ambiguous integration points counting as inside and once counting as outside. The energy norm for lower boundary was $\sim$0.4% lower, and for the upper boundary $\sim$2.9% larger compared to the simulation with 'vote for the majority'. Due to the fact that the error is restricted to the smallest geometrical leaves and the ray-casting errors occur only in the vicinity of the flaws, the deviation in the strain energy norm is rather small.

5.2 Example 2: Screw

This example demonstrates how the algorithm performs for a more complex geometry. To this end, we consider the potentially flawed CAD model of a screw, depicted in Fig. 1. The simulation was carried out on $10 \times 30 \times 10$ finite cells using trivariate B-Splines of polynomial degree $p = 3$ and the open knot vector $U = [0, 0, 0, 0, 1, 2, 3, 3, 3, 3]$. The partitioning depth for the integration of cut FCM cells was set to $k = 3$. The tip surface was loaded with a constant pressure, and the bottom surface was clamped. At the top, several flaws were introduced, resulting in gaps, overlaps, and intersections. In the detailed view in Fig. 29, the free edges (i.e. edges which have only one adjoined face) are highlighted in blue.

Fig. 25 shows the effect of a too fine resolution of $\widehat{TR}_{geo}$. For a subdivision depth $n_{max} = 5$, the flood fill algorithm marks the entire domain as outside.

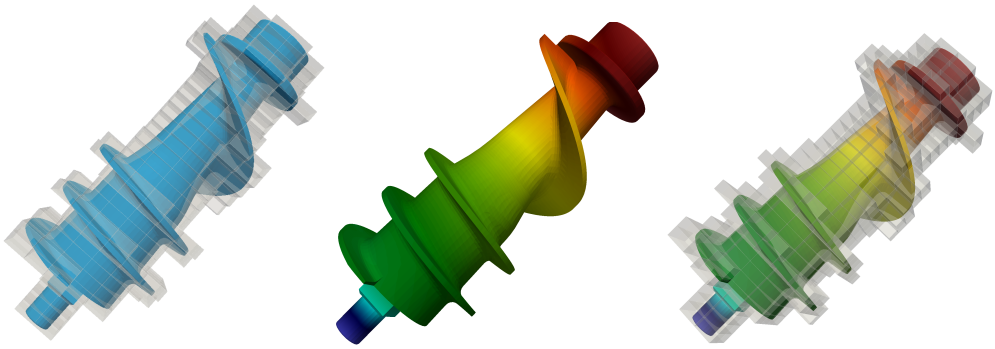
Journal publication: https://doi.org/10.1016/j.cma.2019.04.017



(a) $n_{max} = 5$ (b) $n_{max} = 4$

Figure 25: Cut through an octree approximation $\widehat{TR}_{geo}$, with a) too small cut leaves c_{geo} (black), so that all (non cut) leaves are marked as outside (light grey). b) with one subdivision level less leaves inside (dark grey) can be detected.

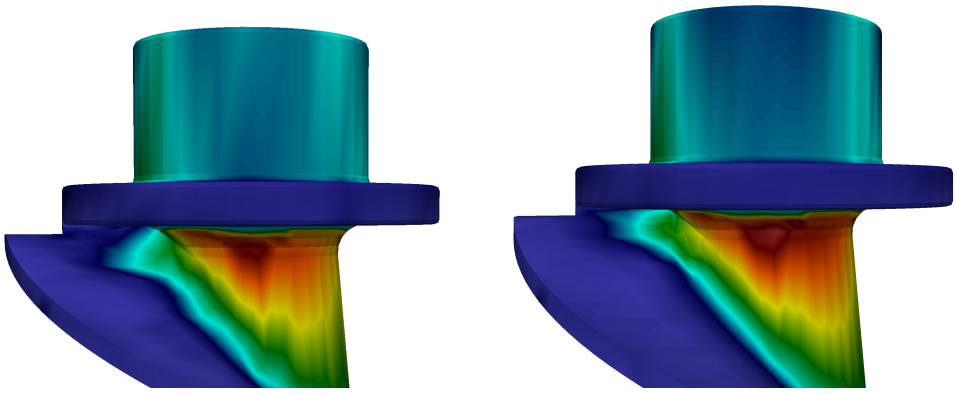
A visual inspection of the displacements of the flawed and the valid model shows no difference (see Fig. 26), whereas differences around the flaws can be detected for the von Mises stresses (see Fig. 27). The stresses at the flawed model are more noisy compared to the valid model.



(a) FCM mesh (b) Flawed (c) Valid, including FCM mesh

Figure 26: Finite Cell mesh and displacement of the flawed and the valid model

Journal publication: https://doi.org/10.1016/j.cma.2019.04.017

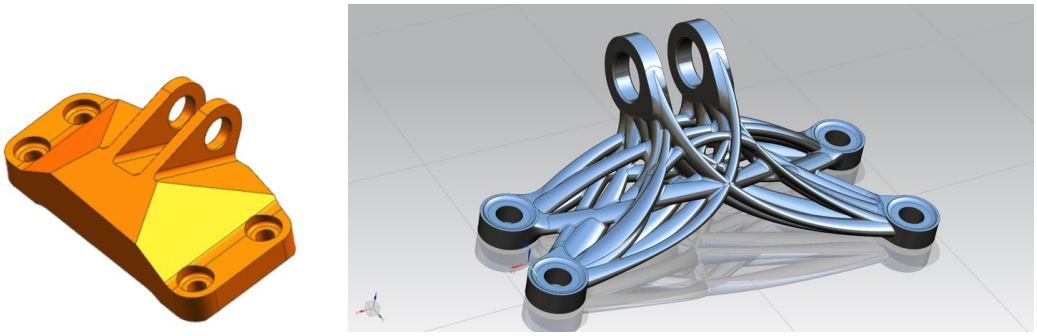


(a) Flawed (b) Valid

Figure 27: Von Mises stresses around the flawed region

5.3 Example 3: Engine Brake

In 2013, a collaboration of Grab Cad and General Electric arranged a competition to find the optimal design of an engine bracket for a General Electric turbofan [74]. The submitted designs were then evaluated, and the top ten were produced using additive manufacturing. The model depicted in Fig. 28 was designed by Sean Morrissey [3].



(a) Initial design (b) Optimized design

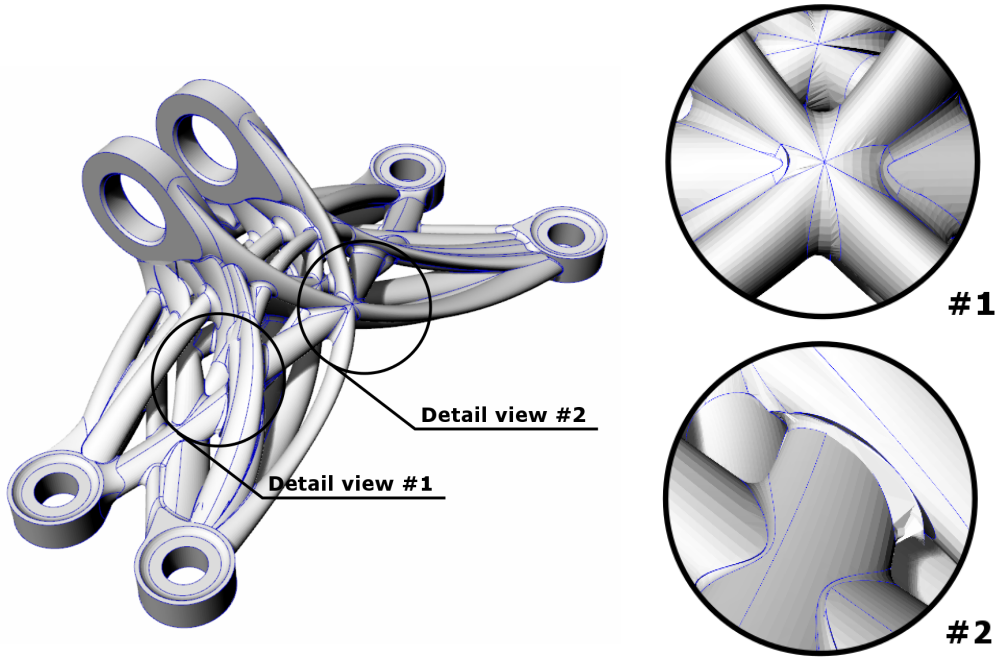
Figure 28: General Electric design challenge for the optimal shape of a jet engine brake

In an attempt to perform a heat diffusion simulation motivated by a local heat source induced by a laser beam during additive manufacturing it turns out that 337.544 triangles have a free edge, indicating a gap/opening between the patches. 2324 triangles were oriented in the wrong direction and innumerable intersections occurred. Due to the immense amount of flaws, geometry healing – and, thus, also the meshing – is not applicable on this raw model. However, using

[3] https://grabcad.com/sean.morrissey-1

Journal publication: https://doi.org/10.1016/j.cma.2019.04.017

the approach presented in this paper, we were able to immediately run a simulation without any further treatment of flaws.



(a) STL model (b) Details

Figure 29: Blue lines denote open edges: In many cases the free edges are fairly close. As can be seen in the detailed views, however, some of the gaps and openings are quite large.

We chose $18 \times 11 \times 6$ elements for the simulation. A partitioning depth for the geometry approximation tree $\widehat{TR}_{geo}$ of $n_{max} = 3$ on each finite cell was applicable. A laser beam is modeled by a small heat source where a local refinement of the finite cell grid was applied. Figure 30 shows the resulting temperatures in the specimen.

Journal publication: https://doi.org/10.1016/j.cma.2019.04.017

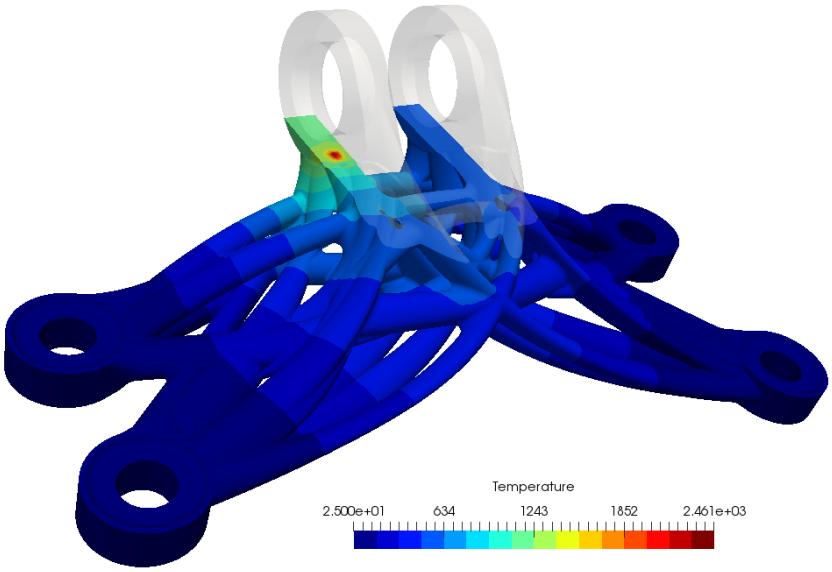


Figure 30: Temperature distribution in the bracket due to a point heat source.

6 Conclusions

This work presents a methodology to address challenges flawed CAD models pose to computational mechanics. Unlike other methods that rely on model reconstruction or geometry healing, the proposed approach herein allows for a numerical analysis directly on the corrupted 'dirty' geometry. Certainly, a simulation on broken geometries will inevitably lead to errors, which are yet of a similar nature to modeling errors due to a representation of a NURBS-based geometry by faceted surfaces. The size of this modeling error depends on the geometric size of the flaws, e.g., the width of a gap between patches. The influence of these errors remains local to the flaw itself. Moreover, the error can be bounded by performing bracketing simulations. Therein, the upper bound is delivered by a computation considering all ambiguous integration points to lie inside the physical domain and the lower bound is generated by considering the inverse situation. Several examples demonstrate the capability of the proposed method, showing that results of high accuracy can be obtained.

Journal publication: https://doi.org/10.1016/j.cma.2019.04.017

Acknowledgements

We gratefully acknowledge the support of the German Research Foundation under the Grant No. Ra 624/22-2.

Shuohui Yin from Hohai University, China as a visiting student at Technische Universität München, Germany, thanks Prof. Ernst Rank and Dr. Stefan Kollmannsberger at Technische Universität München for one-year academic guidance and discussion. He also thankfully acknowledges the support of the scholarship from China Scholarship Council (CSC).

References

[1] J. A. Cottrell, T. J. R. Hughes, and Y. Bazilevs, *Isogeometric Analysis: Toward Integration of CAD and FEA*. John Wiley & Sons, Aug. 2009. ISBN 978-0-470-74909-8

[2] 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*, vol. 194, no. 39–41, pp. 4135–4195, Oct. 2005. doi: 10.1016/j.cma.2004.10.008

[3] F. Cirak, M. J. Scott, E. K. Antonsson, M. Ortiz, and P. Schröder, "Integrated modeling, finite-element analysis, and engineering design for thin-shell structures using subdivision," *Computer-Aided Design*, vol. 34, no. 2, pp. 137–148, Feb. 2002. doi: 10.1016/S0010-4485(01)00061-6

[4] P. Kagan and A. Fischer, "Integrated mechanically based CAE system using B-Spline finite elements," *Computer-Aided Design*, vol. 32, no. 8, pp. 539–552, Aug. 2000. doi: 10.1016/S0010-4485(00)00041-5

[5] J. Yang and S. Han, "Repairing CAD model errors based on the design history," *Computer-Aided Design*, vol. 38, no. 6, pp. 627–640, Jun. 2006. doi: 10.1016/j.cad.2006.02.007

[6] M. Mäntylä, *An Introduction to Solid Modeling*, ser. Principles of computer science series. Rockville: Computer Science Press, 1988, no. 13. ISBN 978-0-88175-108-6

[7] F. Massarwi and G. Elber, "A B-spline based framework for volumetric object modeling," *Computer-Aided Design*, vol. 78, pp. 36–47, Sep. 2016. doi: 10.1016/j.cad.2016.05.003

[8] G. Butlin and C. Stops, "CAD Data Repair," in *Proceedings of the 5th International Meshing Roundtable*, 1996, pp. 7–12.

[9] H. Gu, T. R. Chase, D. C. Cheney, T. T. Bailey, and D. Johnson, "Identifying, Correcting, and Avoiding Errors in Computer-Aided Design Models Which Affect Interoperability," *Journal of Computing and Information Science in Engineering*, vol. 1, no. 2, pp. 156–166, May 2001. doi: 10.1115/1.1384887

Journal publication: https://doi.org/10.1016/j.cma.2019.04.017

[10] N. A. Petersson and K. K. Chand, "Detecting Translation Errors in CAD Surfaces and Preparing Geometries for Mesh Generation," Lawrence Livermore National Lab., CA (US), Newport Beach, CA, Tech. Rep. UCRL-JC-144019, Aug. 2001.

[11] J. Yang, S. Han, and S. Park, "A method for verification of computer-aided design model errors," *Journal of Engineering Design*, vol. 16, no. 3, pp. 337–352, Jun. 2005. doi: 10.1080/09544820500126565

[12] C. S. Chong, A. Senthil Kumar, and H. P. Lee, "Automatic Mesh-healing Technique for Model Repair and Finite Element Model Generation," *Finite Elements in Analysis and Design*, vol. 43, no. 15, pp. 1109–1119, Nov. 2007. doi: 10.1016/j.finel.2007.06.009

[13] F. S. Nooruddin and G. Turk, "Simplification and repair of polygonal models using volumetric techniques," *IEEE Transactions on Visualization and Computer Graphics*, vol. 9, no. 2, pp. 191–205, Apr. 2003. doi: 10.1109/TVCG.2003.1196006

[14] S. Bischoff and L. Kobbelt, "Structure Preserving CAD Model Repair," *Computer Graphics Forum*, vol. 24, no. 3, pp. 527–536, Sep. 2005. doi: 10.1111/j.1467-8659.2005.00878.x

[15] O. Busaryev, T. K. Dey, and J. A. Levine, "Repairing and Meshing Imperfect Shapes with Delaunay Refinement," in *2009 SIAM/ACM Joint Conference on Geometric and Physical Modeling*. New York, NY, USA: ACM, 2009. doi: 10.1145/1629255.1629259

[16] Z. J. Wang and K. Srinivasan, "An adaptive Cartesian grid generation method for 'Dirty' geometry," *International Journal for Numerical Methods in Fluids*, vol. 39, no. 8, pp. 703–717, Jul. 2002. doi: 10.1002/fld.344

[17] Y. K. Lee, C. K. Lim, H. Ghazialam, H. Vardhan, and E. Eklund, "Surface Mesh Generation for Dirty Geometries by the Cartesian Shrink-wrapping Technique," *Engineering with Computers*, vol. 26, no. 4, pp. 377–390, Aug. 2010. doi: 10.1007/s00366-009-0171-0

[18] R. Gasparini, T. Kosta, and I. Tsukanov, "Engineering analysis in imprecise geometric models," *Finite Elements in Analysis and Design*, vol. 66, pp. 96–109, Apr. 2013. doi: 10.1016/j.finel.2012.10.011

[19] L. V. Kantorovich and V. I. Krylov, *Approximate Methods of Higher Analysis*. Interscience Publishers, 1958. ISBN 978-0-486-82160-3

[20] IntactSolutions, "ScanAndSolve," http://www.scan-and-solve.com/, 2013.

[21] J. Parvizian, A. Düster, and E. Rank, "Finite cell method," *Computational Mechanics*, vol. 41, no. 1, pp. 121–133, Apr. 2007. doi: 10.1007/s00466-007-0173-y

[22] 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*, vol. 197, no. 45–48, pp. 3768–3782, Aug. 2008. doi: 10.1016/j.cma.2008.02.036

Journal publication: https://doi.org/10.1016/j.cma.2019.04.017

[23] A. Abedian, J. Parvizian, A. Düster, and E. Rank, "Finite cell method compared to h-version finite element method for elasto-plastic problems," *Applied Mathematics and Mechanics*, vol. 35, no. 10, pp. 1239–1248, Oct. 2014. doi: 10.1007/s10483-014-1861-9

[24] B. Wassermann, S. Kollmannsberger, T. Bog, and E. Rank, "From geometric design to numerical analysis: A direct approach using the Finite Cell Method on Constructive Solid Geometry," *Computers & Mathematics with Applications*, Mar. 2017. doi: 10.1016/j.camwa.2017.01.027

[25] S. Cai, W. Zhang, J. Zhu, and T. Gao, "Stress constrained shape and topology optimization with fixed mesh: A B-spline finite cell method combined with level set function," *Computer Methods in Applied Mechanics and Engineering*, vol. 278, pp. 361–387, Aug. 2014. doi: 10.1016/j.cma.2014.06.007

[26] J. P. Groen, M. Langelaar, O. Sigmund, and M. Ruess, "Higher-order multi-resolution topology optimization using the finite cell method," *International Journal for Numerical Methods in Engineering*, Jan. 2016. doi: 10.1002/nme.5432

[27] M. Joulaian and A. Düster, "Local enrichment of the finite cell method for problems with material interfaces," *Computational Mechanics*, vol. 52, no. 4, pp. 741–762, Oct. 2013. doi: 10.1007/s00466-013-0853-8

[28] M. Joulaian, S. Duczek, U. Gabbert, and A. Düster, "Finite and spectral cell method for wave propagation in heterogeneous materials," *Computational Mechanics*, vol. 54, no. 3, pp. 661–675, Apr. 2014. doi: 10.1007/s00466-014-1019-z

[29] 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*, vol. 99, no. 1, pp. 26–53, Jul. 2014. doi: 10.1002/nme.4663

[30] M. Elhaddad, N. Zander, S. Kollmannsberger, A. Shadavakhsh, V. Nübel, and E. Rank, "Finite Cell Method: High-Order Structural Dynamics for Complex Geometries," *International Journal of Structural Stability and Dynamics*, vol. 15, no. 7, p. 1540018, Apr. 2015. doi: 10.1142/S0219455415400180

[31] T. Bog, N. Zander, S. Kollmannsberger, and E. Rank, "Weak imposition of frictionless contact constraints on automatically recovered high-order, embedded interfaces using the finite cell method," *Computational Mechanics*, Aug. 2017. doi: 10.1007/s00466-017-1464-6

[32] A. Mongeau, "Large deformation two- and three- dimensional contact on embedded interfaces using the Finite Cell Method," Master's Thesis, Technische Universität München, Dec. 2015.

[33] S. Kollmannsberger, A. Özcan, J. Baiges, M. Ruess, E. Rank, and A. Reali, "Parameter-free, weak imposition of Dirichlet boundary conditions and coupling of trimmed and non-conforming patches," *International Journal for Numerical Methods in Engineering*, vol. 101, no. 9, pp. 670–699, Mar. 2015. doi: 10.1002/nme.4817

Journal publication: https://doi.org/10.1016/j.cma.2019.04.017

[34] 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*, vol. 55, no. 3, pp. 499–517, Feb. 2015. doi: 10.1007/s00466-014-1118-x

[35] L. Kudela, N. Zander, S. Kollmannsberger, and E. Rank, "Smart octrees: Accurately integrating discontinuous functions in 3D," *Computer Methods in Applied Mechanics and Engineering*, vol. 306, pp. 406–426, Jul. 2016. doi: 10.1016/j.cma.2016.04.006

[36] T.-P. Fries and S. Omerović, "Higher-order accurate integration of implicit geometries," *International Journal for Numerical Methods in Engineering*, vol. 106, no. 5, pp. 323–371, Jan. 2015. doi: 10.1002/nme.5121

[37] M. Joulaian, S. Hubrich, and A. Düster, "Numerical integration of discontinuities on arbitrary domains based on moment fitting," *Computational Mechanics*, vol. 57, no. 6, pp. 979–999, Jun. 2016. doi: 10.1007/s00466-016-1273-3

[38] S. Hubrich, P. D. Stolfo, L. Kudela, S. Kollmannsberger, E. Rank, A. Schröder, and A. Düster, "Numerical integration of discontinuous functions: Moment fitting and smart octree," *Computational Mechanics*, pp. 1–19, Jul. 2017. doi: 10.1007/s00466-017-1441-0

[39] D. Giraldo and D. Restrepo, "The spectral cell method in nonlinear earthquake modeling," *Computational Mechanics*, pp. 1–21, Aug. 2017. doi: 10.1007/s00466-017-1454-8

[40] 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*, vol. 50, no. 4, pp. 445–478, Feb. 2012. doi: 10.1007/s00466-012-0684-z

[41] E. Rank, M. Ruess, S. Kollmannsberger, D. Schillinger, and A. Düster, "Geometric modeling, isogeometric analysis and the finite cell method," *Computer Methods in Applied Mechanics and Engineering*, vol. 249-252, pp. 104–115, Dec. 2012. doi: 10.1016/j.cma.2012.05.022

[42] 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*, vol. 269, pp. 46–71, Feb. 2014. doi: 10.1016/j.cma.2013.10.009

[43] F. de Prenter, C. V. Verhoosel, G. J. van Zwieten, and E. H. van Brummelen, "Condition number analysis and preconditioning of the finite cell method," *Computer Methods in Applied Mechanics and Engineering*, vol. 316, no. Supplement C, pp. 297–327, Apr. 2017. doi: 10.1016/j.cma.2016.07.006

[44] E. Burman, P. Hansbo, and M. G. Larson, "A stabilized cut finite element method for partial differential equations on surfaces: The Laplace–Beltrami operator," *Computer Methods in Applied Mechanics and Engineering*, vol. 285, pp. 188–207, Mar. 2015. doi: 10.1016/j.cma.2014.10.044

Journal publication: https://doi.org/10.1016/j.cma.2019.04.017

[45] E. Burman and P. Hansbo, "Fictitious domain finite element methods using cut elements: I. A stabilized Lagrange multiplier method," *Computer Methods in Applied Mechanics and Engineering*, vol. 199, no. 41-44, pp. 2680–2686, Oct. 2010. doi: 10.1016/j.cma.2010.05.011

[46] N. M. Patrikalakis, T. Sakkalis, and G. Shen, "Boundary Representation Models: Validity and Rectification," in *The Mathematics of Surfaces IX*. Springer, London, 2000, pp. 389–409. ISBN 978-1-4471-1153-5 978-1-4471-0495-7

[47] C. M. Hoffmann, *Geometric and Solid Modeling: An Introduction*, ser. The Morgan Kaufmann series in computer graphics and geometric modeling. San Mateo, Calif: Morgan Kaufmann, 1989. ISBN 978-1-55860-067-6

[48] Y. Bazilevs, V. M. Calo, J. A. Cottrell, J. A. Evans, T. J. R. Hughes, S. Lipton, M. A. Scott, and T. W. Sederberg, "Isogeometric analysis using T-splines," *Computer Methods in Applied Mechanics and Engineering*, vol. 199, no. 5–8, pp. 229–263, Jan. 2010. doi: 10.1016/j.cma.2009.02.036

[49] T. W. Sederberg, G. T. Finnigan, X. Li, H. Lin, and H. Ipson, "Watertight Trimmed NURBS," in *ACM SIGGRAPH 2008 Papers*. New York, NY, USA: ACM, 2008. doi: 10.1145/1399504.1360678

[50] B. Rumpe, *Modeling with UML: Language, Concepts, Methods*. Springer International Publishing, 2016. ISBN 978-3-319-33932-0

[51] A. Goldberg and D. Robson, *Smalltalk-80: The Language and Its Implementation*. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 1983. ISBN 978-0-201-11371-6

[52] 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*, vol. 89, no. 9, pp. 1171–1202, 2012. doi: 10.1002/nme.3289

[53] N. Zander, S. Kollmannsberger, M. Ruess, Z. Yosibash, and E. Rank, "The Finite Cell Method for linear thermoelasticity," *Computers & Mathematics with Applications*, vol. 64, no. 11, pp. 3527–3541, Dec. 2012. doi: 10.1016/j.camwa.2012.09.002

[54] M. Elhaddad, N. Zander, T. Bog, L. Kudela, S. Kollmannsberger, J. S. Kirschke, T. Baum, M. Ruess, and E. Rank, "Multi-level hp-finite cell method for embedded interface problems with application in biomechanics," *International Journal for Numerical Methods in Biomedical Engineering*, vol. 34, no. 4, p. e2951, 2018. doi: 10.1002/cnm.2951

[55] S. Kollmannsberger, A. Özcan, M. Carraturo, N. Zander, and E. Rank, "A hierarchical computational model for moving thermal loads and phase changes with applications to selective laser melting," *Computers & Mathematics with Applications*, vol. 75, no. 5, pp. 1483–1497, Mar. 2018. doi: 10.1016/j.camwa.2017.11.014

Journal publication: https://doi.org/10.1016/j.cma.2019.04.017

[56] A. Düster, E. Rank, and B. A. Szabó, "The p-version of the finite element method and finite cell methods," in *Encyclopedia of Computational Mechanics*, E. Stein, R. Borst, and T. J. R. Hughes, Eds. Chichester, West Sussex: John Wiley & Sons, 2017, vol. 2, pp. 1–35. ISBN 978-1-119-00379-3

[57] M. Dauge, A. Düster, and E. Rank, "Theoretical and Numerical Investigation of the Finite Cell Method," *Journal of Scientific Computing*, vol. 65, no. 3, pp. 1039–1064, Mar. 2015. doi: 10.1007/s10915-015-9997-3

[58] 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*, vol. 200, no. 47-48, pp. 3358–3380, Nov. 2011. doi: 10.1016/j.cma.2011.08.002

[59] N. Zander, T. Bog, M. Elhaddad, F. Frischmann, S. Kollmannsberger, and E. Rank, "The multi-level hp-method for three-dimensional problems: Dynamically changing high-order mesh refinement with arbitrary hanging nodes," *Computer Methods in Applied Mechanics and Engineering*, vol. 310, pp. 252–277, Oct. 2016. doi: 10.1016/j.cma.2016.07.007

[60] V. Varduhn, M.-C. Hsu, M. Ruess, and D. Schillinger, "The tetrahedral finite cell method: Higher-order immersogeometric analysis on adaptive non-boundary-fitted meshes," *International Journal for Numerical Methods in Engineering*, vol. 107, no. 12, pp. 1054–1079, Jan. 2016. doi: 10.1002/nme.5207

[61] 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*, vol. 284, pp. 1005–1053, Feb. 2015. doi: 10.1016/j.cma.2014.10.040 00015.

[62] S. Duczek and U. Gabbert, "The finite cell method for polygonal meshes: Poly-FCM," *Computational Mechanics*, pp. 1–32, Jun. 2016. doi: 10.1007/s00466-016-1307-x

[63] A. Abedian, J. Parvizian, A. Düster, H. Khademyzadeh, and E. Rank, "Performance of Different Integration Schemes in Facing Discontinuities in the Finite Cell Method," *International Journal of Computational Methods*, vol. 10, no. 03, p. 1350002, Jun. 2013. doi: 10.1142/S0219876213500023

[64] M. Ruess, Y. Bazilevs, D. Schillinger, N. Zander, and E. Rank, "Weakly enforced boundary conditions for the NURBS-based Finite Cell Method," in *European Congress on Computational Methods in Applied Sciences and Engineering (ECCOMAS)*, Vienna, Austria, 2012. ISBN 978-3-9502481-9-7

[65] 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*, vol. 95, no. 10, pp. 811–846, Sep. 2013. doi: 10.1002/nme.4522

Journal publication: https://doi.org/10.1016/j.cma.2019.04.017

[66] 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*, vol. 284, pp. 881–905, Feb. 2015. doi: 10.1016/j.cma.2014.11.014

[67] F. P. Preparata and M. I. Shamos, *Computational Geometry: An Introduction*. New York, NY, USA: Springer-Verlag New York, Inc., 1985. ISBN 978-0-387-96131-6

[68] B. Žalik and I. Kolingerova, "A cell-based point-in-polygon algorithm suitable for large sets of points," *Computers & Geosciences*, vol. 27, no. 10, pp. 1135–1145, Dec. 2001. doi: 10.1016/S0098-3004(01)00037-1

[69] A. Sitek, R. H. Huesman, and G. T. Gullberg, "Tomographic reconstruction using an adaptive tetrahedral mesh defined by a point cloud," *IEEE transactions on medical imaging*, vol. 25, no. 9, pp. 1172–1179, Sep. 2006. doi: 10.1109/TMI.2006.879319

[70] G. Taylor, "Point in Polygon Test," *Survey Review*, vol. 32, no. 254, pp. 479–484, Oct. 1994. doi: 10.1179/sre.1994.32.254.479

[71] K. B. Salomon, "An efficient point-in-polygon algorithm," *Computers & Geosciences*, vol. 4, no. 2, pp. 173–178, Jan. 1978. doi: 10.1016/0098-3004(78)90085-7

[72] J. D. Foley, A. V. Dam, S. K. Feiner, J. F. Hughes, and R. L. Phillips, *Introduction to Computer Graphics*. Addison-Wesley, 1997. ISBN 978-0-201-60921-9

[73] G. Strang, *An Analysis of the Finite Element Method*. Englewood Cliffs, N.J: Prentice-Hall, 1973. ISBN 0-13-032946-0

[74] GrabCad, "General Electric jet engine bracket challenge - GrabCAD," https://grabcad.com/challenges/ge-jet-engine-bracket-challenge, 2013.

Journal publication: https://doi.org/10.1016/j.cma.2019.04.017

2.4 From pictures to simulations

Geometric models aren't always readily available for computational analysis. This is, for example, often the case in the field of cultural heritage preservation, as there are usually no digital CAD models available for historical structures. Moreover, even if schematic drawings exist, the shape of the object may differ from them – especially if the structure is exposed to damaging effects such as erosion, floods, earthquakes, or wars. In these cases, other shape measurement techniques need to be employed. The two most popular methods for this purpose are terrestrial laser scanning and close range photogrammetry-based reconstructions. Especially photogrammetry has gained a lot of attention recently, due to the inexpensiveness of the required equipment and because of the rapid development of the computational resources as well as the associated algorithms that allow for efficient, almost real-time reconstructions [29].

As already pointed out at the end of section 2.1, the methods of laser scanning and photogrammetry both reproduce the shape of the geometry of interest in the form of point clouds: a set of points representing the surface of the object. Such point clouds are normally not directly suited for numerical analysis. Instead, it is necessary to employ the reconstruction process depicted in fig. 9, and the Finite Cell Method is able to provide a suitable shortcut.

This shortcut relies on the construction of a robust point in membership test for point clouds, which can be constructed quite easily. Consider the domain Ω_{phy}, represented by a set of sample points $\mathbf{p}_i$ and their associated normal vectors $\mathbf{n}_i$, as depicted in fig. 11. If no outliers are present, the set of pairs $S = \{\mathbf{p}_i, \mathbf{n}_i\}$ represent a discrete sampling of the boundary $\partial\Omega_{\text{phy}}$ of the domain.

Each element in S defines a hyperplane that separates the space in two half spaces: the open half-space Ω_i^- on the side of the hyperplane where the normal vector $\mathbf{n}_i$ points, and the closed half-space Ω_i^+ on the other side. For every $\mathbf{x} \in \Omega_i^+$, the following holds:

$$(\mathbf{p}_i - \mathbf{x}) \cdot \mathbf{n}_i \geq 0. \tag{1}$$

Therefore, a simple way to determine whether a quadrature point $\mathbf{q}$ lies inside or outside the domain is to find the $\mathbf{p}_i$ and the associated $\mathbf{n}_i$ in S that lies closest to $\mathbf{q}$, and to evaluate the

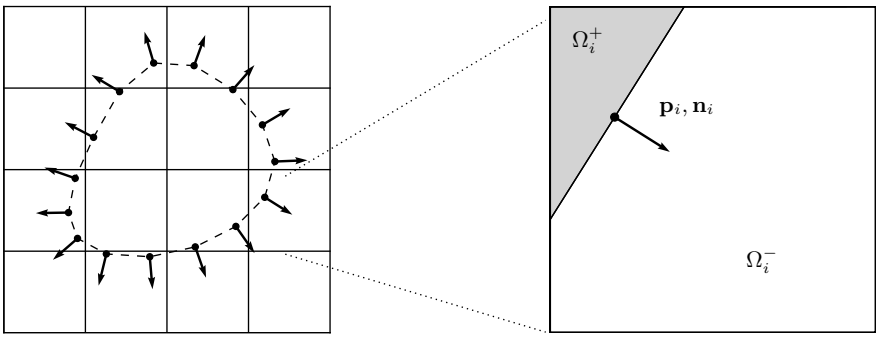


Figure 11: Point membership classification on oriented point clouds. The domain is represented by a set of points $\mathbf{p}_i$ and associated normals $\mathbf{n}_i$. Every such pair locally separates the space along a hyperplane into two half-spaces: Ω_i^- and Ω_i^+.

scalar product of eq. (1). This rather simple algorithm, equipped with an efficient nearest neighbor query, works robustly on complex point clouds as well.

Figures 12 and 13 depict the entire pipeline of how a computation may be carried out starting from a set of pictures of an object. Figure 12 depicts the step of shape acquisition. A statue is photographed from several angles. These pictures then form the basis of generating an oriented point cloud. To this end, it is possible to employ standard computer vision software such as the open source toolbox *OpenMVG* [30], for example. The point cloud is then filtered

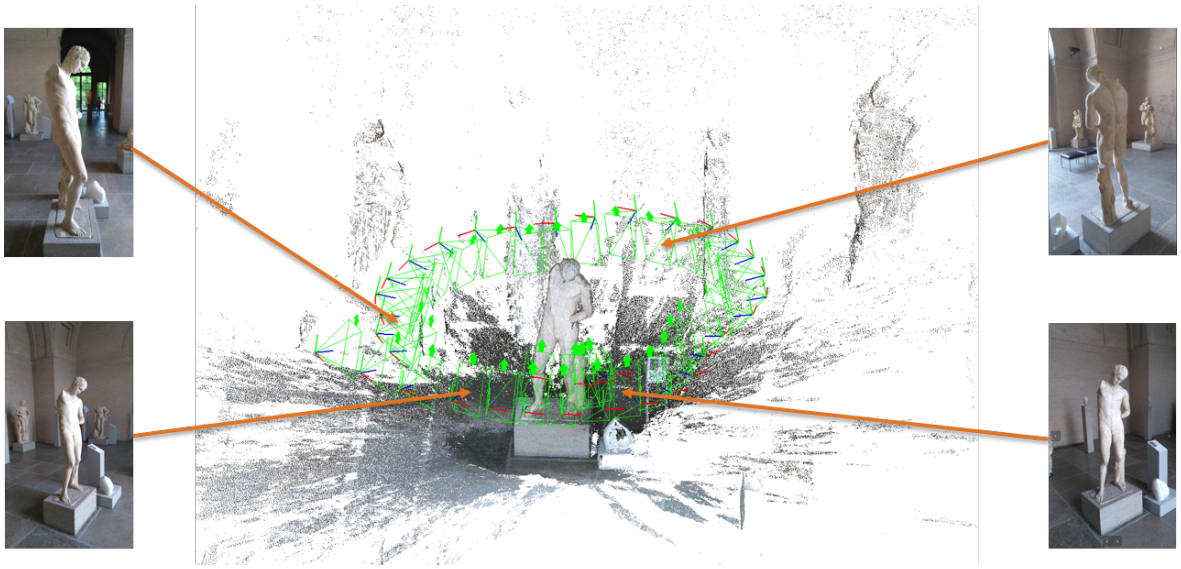
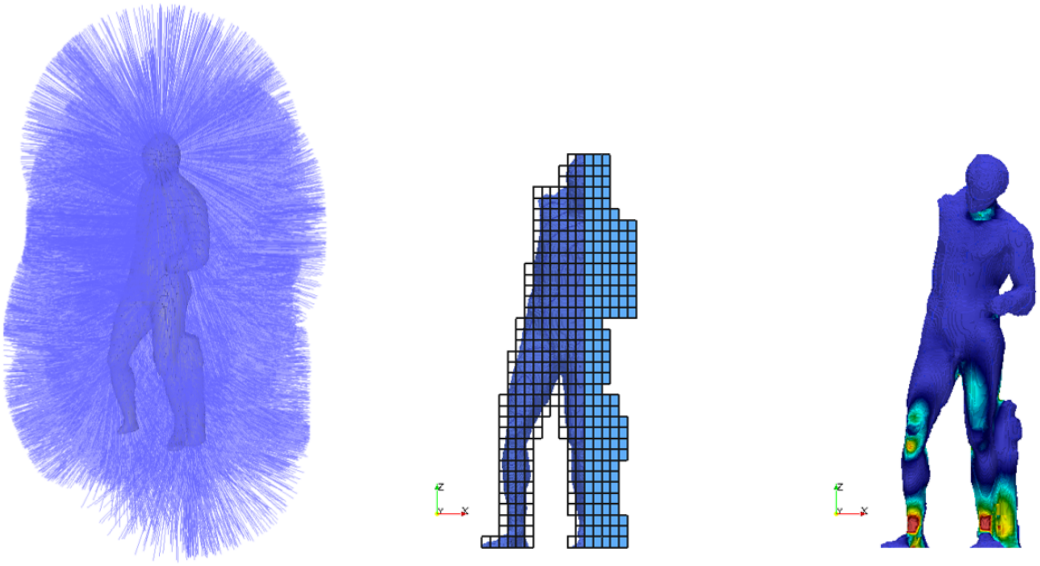


Figure 12: xyz

in such a way that outliers – i.e. points that are not considered to be part of the hull enclosing the volume to be computed – are eliminated. These pre-processing steps are common to all computations based on point clouds, and directly associated to the step of image acquisition, see also fig. 9. The shortcut of an analysis for computational mechanics using the Finite Cell Method is then composed of the three stages depicted in fig. 13.

The previous steps of shape acquisition result in the oriented point cloud depicted in fig. 13a along with its normal vectors. The point cloud is inserted into a grid of finite cells of the size of the bounding box enclosing the point cloud. The finite cells that don't contain points are deleted, see fig. 13b for a visualization of the resulting discretization. Next, Dirichlet boundary conditions are assigned directly to the cells at the bottom of the statue, and self weight is assigned as a volume load. The weak form is then integrated using the point membership test given by eq. (1). For the sake of simplicity, the integration is carried out using an octree. The finite cell analysis then delivers the Von Mises stresses depicted in fig. 13c. The stresses are normalized since a homogeneous linear material behavior is assumed.

The extension of the Finite Cell Method to point clouds as geometric models depicted in figs. 12 and 13 is very promising but has not yet fully matured. The basic ideas are published in [31]. Extensions including a more thorough investigation of the convergence of the method and its sensitivity to outliers – as well as the application of Neumann boundary conditions directly on the point clouds which, in most cases, do not coincide with the boundary of the discretization are published in [27]. Open tasks for research are: the application of Dirichlet boundary



(a) Oriented point cloud (cleaned from fig. 12) with normal vectors

(b) Point cloud embedded into FCM grid

(c) Von Mises stresses (normalized)

Figure 13: Direct simulation of point clouds with the Finite Cell Method

conditions such as, for example, those presented in section 3.3 and the construction of smarter integration schemes similar to those presented in section 3.2.

3 Discretization

3.1 Overview

The wide range of applications and their specific requirements also spurred developments in the underlying discretizational technology of the FCM itself. First and foremost, contributions to an improved integration will be pointed out in section 3.2 before the appropriate handling of Dirichlet boundary conditions is addressed in section 3.3. These first two issues are of crucial importance for any kind of embedded domain method such as the FCM. This is followed by a brief discussion of contributions which are not restricted to the FCM but naturally extend to boundary conforming methods of high order in general. The focus here lies on challenges for elliptic equations which stem from irregularities in the solution. Two approaches are possible for their appropriate resolution: a) a direct enrichment using the known type of irregularity directly at the correct position, and b) an enrichment in the vicinity of the solution with functions to better approximate the possibly unknown type of irregularity. Numerous other classifications exist (see, e.g. [32]), but this simple one suffices in the scope of this treatise. The contributions to these types of challenges are briefly discussed in section 3.4 by means of examples. Therein, one type of enrichment, the multi-level *hp*-method, is especially highlighted by including the corresponding journal publication in the subsection . The technical possibility to be able to enrich the approximation then leads to questions of where this enrichment should be carried out, which type shall be used, and how much of it. The corresponding contribution is briefly summarized in Refinement indicators and error estimation.

3.2 Integration

Probably the most decisive question concerning efficiency and accuracy of immersed-boundary methods in general and the FCM in particular is the numerical integration of the bilinear form for cells cut by a boundary. The bi-linear form defined on these cells exhibits a discontinuity introduced by α, which is not captured by element boundaries and for which standard Gaussian integration schemes lead to poor accuracy. As a remedy, adapted quadrature schemes for the FCM have been developed, see for example in [33, 34] [35]. These schemes are based on adaptive quadtree and octree refinements – or they take the voxelized discretization of the physical domain into account directly. They can be fully automated because they are robust with respect to complicated geometries and non-smooth integrands. This is highly desirable, especially considering that the initial goal of the FCM was to peruse a fully automatic computational analysis.

However, due to the large number of integration points, the related numerical effort may still be very high. The obvious reason is that quadtrees and octrees solely rely on a point membership test and only iteratively approximate the boundary of the domain. Still, it is possible to construct much more efficient schemes using less Gaussian points, for example by means of moment-fitting methods [36]. These, in essence, construct an individual integration rule by solving the moment equations for given positions of integration points to compute their corresponding weights for each individual cut cell. This reduced set of integration points and corresponding weights can then be used in subsequent computations. The solution of the mo-

ment equations can be computationally expensive. Therefore, moment fitting especially pays off in transient computations in which the integration points may be used repeatedly among with their corresponding weights. Very recently, it was demonstrated that the computation of the equivalent integration weights can be significantly reduced[3] such that the computational effort to obtain the weights is drastically reduced [37]. Thus, the computational time invested into moment fitting amortizes much earlier as compared to classical tree based schemes – to the point that moment fitting becomes a favorable choice even for non-transient computations.

An alternative approach to construct smarter integration schemes with less Gaussian points is to incorporate more knowledge about the boundary of the physical model. The next step in this direction is to use the locations where the boundary of the physical problem intersects the computational grid. The most obvious approach in this context is to sub-triangulate or sub-tetrahedralize cut cells. This was carried out e.g. in [38], in the context of fixed-grid fluid-structure interaction, or in [39] in the context of the weak imposition of Dirichlet boundary conditions. However, depending on the geometric complexity of the boundary, this type of sub-integration may still lead to numerous Gaussian points and, at the same time, compromises the robustness of this approach.

Interestingly, it is possible to combine the robustness of the integration by a quad- or octree with a higher accuracy of the integration. The most straightforward starting point is to use as little integration points as possible and a quadrilateral subdivision. However, this leads to the infamous 16 marching cubes cases which are inconvenient to handle. They can be reduced to only two cases under the provision that both triangular *and* quadrilateral sub-integration elements are allowed, see fig. 14.

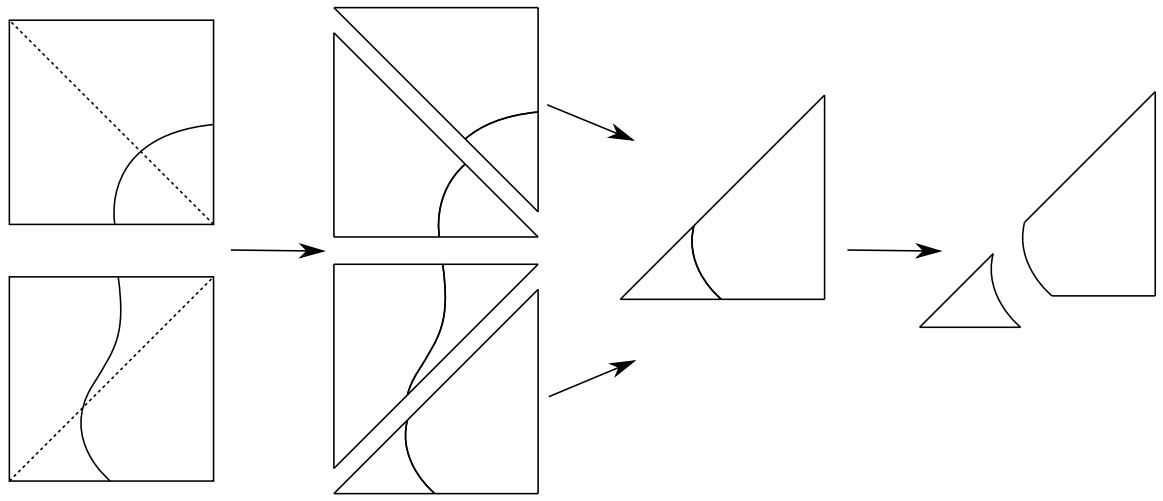


Figure 14: Idea of exact composed integration: Every cut cell is subdivided into two triangles along a diagonal line. The triangles are further cut by the boundary. Only the depicted two cases need to be handled because topologically more complex cases are handled by recursive subdivision. [40]

The original robustness of a quadtree is simply added by the following observation: intersection cases that are topologically more complex all fall back to the two cases depicted in fig. 14. If a curve exhibits a kink, then the subdivision of the quadrilateral into triangles is carried

[3] This is achieved by choosing Lagrange functions defined by Gauss-Lobatto points. This leads to a diagonal coefficient matrix for the moment fitting equations such that solving for the unknown integration weights becomes trivial.

out in such a way that the subdivision runs through that kink. A recursive application of these two steps quickly converges to an accurate integration. These findings were published in [40] along with a more detailed description and examples.

Unfortunately, this scheme does not directly extend to three dimensions because a division of one octant of an octee into tetra- and hexahedrals only works well only for smooth surfaces. Cases in which sharp features in cutting surfaces needed to be incorporated are problematic. A fallback option into a further refinement does not resolve these cases and, therefore, robustness is lost. However, a closer look reveals that it is actually not necessary to divide cut cases into tetrahedrals or other topological entities such as wedges. Instead, it is possible to use the octree directly in the following simple manner: a) subdivide cut cells regularly into 8 octants b) move the inner node towards the cutting surface c) move the nodes of the cut surface of the octant into the intersection point with the cutting surface. d) for a high-order version: blend the surfaces of the octant towards the cutting surface. As an example, consider the case of an octant subdivided into the two domains Ω_{phys} and Ω_{fict} as depicted in fig. 15, then:

- If any of the 12 edges is intersected by the domain $\partial\Omega_{\text{phys}}$, it is identified as an *active edge*. On active edges, the corresponding edge node is moved to the point of intersection. If an edge is inactive, its edge node stays in the center.
- If any of the 6 faces is bounded by an active edge, the face is identified as an *active face*. On active faces, the face node is moved onto the intersection curve between $\partial\Omega_{\text{phys}}$ and the face. If the face is inactive, its face node remains in its center.
- The mid-node of the cut cell is moved onto $\partial\Omega_{\text{phys}}$.

The resulting cells have corners that are either located in Ω_{phys} or in Ω_{fict}. This gives a linear approximation of the boundary. Higher order cases are treated by blending the surfaces of the corresponding octants towards the surface dividing $\Omega_{\text{phys}}$and Ω_{fict}.

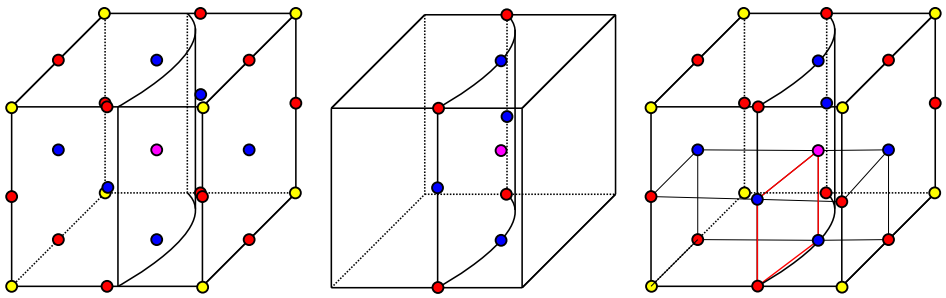


Figure 15: Smart octree generation. The nodes on active edges, faces, and the mid-node are moved onto the interface. For simplicity, only two resulting subcells are shown. The corners of the red bilinear quadrilateral lie on the interface.[41]

This naturally provides the possibility to treat six kinks in surfaces plus one C^0 node irregularity per octree-refinement. If one refinement is not sufficient to treat all present features at once, the algorithm is applied recursively. This scheme converges extremely fast. Actually, no more than two refinements are needed in practical engineering applications. Further algorithmic details were published in [41]. For a comparison between the performance of the smart octree and moment fitting see [42]

It is yet an open question which of the integration schemes discussed in the beginning of this section is the most suitable in which case. The most favorable integration scheme will likely

depend on the geometric model used (CSG, B-rep, voxels, point clouds) regarding the physics involved (geometrical or material linear or non-linearities, transient or stationary problems). Another aspect is whether topological changes are expected during the computation.

3.3 Dirichlet boundary conditions

The treatment of Dirichlet boundary conditions is a common but challenging task for immersed-boundary methods, since the non-boundary-conforming discretization prohibits a strong enforcement in the classical sense. Alternatively, the penalty method [43] or a weak enforcement strategy, which was first introduced by Nitsche for Laplace problems [44], were extended to the FCM [45, 39]. They exhibit the problem that a stabilization parameter is needed to maintain positive definiteness of the system. This parameter can be estimated by a local eigenvalue computation. At the time, the goal was to avoid this estimation. A promising alternative was developed for low orders methods in [46] and extended to high-order methods and curved geometries in [47]. The basic idea is to start with a mixed formulation in which the material law is discretized with functions in L_2. Their discontinuity at the boundaries of the cells allows to condense the unknown gradients of the primal variables out of the system at the discrete level such that they do not have to be computed explicitly. The key point is that this leads to terms which are equivalent to the classical Nitsche approach – with the only difference that the stabilization parameter is replaced by a term similar to an inverse mass matrix defined on Ω_{fict}. The discretization of that term depends on the choice of polynomials used to discretize the material law. It could be demonstrated that this term indeed provides the necessary, automatic stabilization. Furthermore, the boundary conditions were re-formulated to solve the problem of coupling patches by a trim-and-glue approach, which is briefly explained in section 3.4.1.

Journal Publication

title:	Parameter-free, weak imposition of Dirichlet boundary conditions and coupling of trimmed and non-conforming patches
authors:	S. Kollmannsberger, A. Özcan, J. Baiges, M. Ruess, E. Rank, A. Reali
published at:	*International Journal for Numerical Methods in Engineering*
publisher:	Wiley
year:	2015
volume:	101
pages:	670–699
doi:	https://doi.org/10.1002/nme.4817

INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN ENGINEERING
Int. J. Numer. Meth. Engng 0000; **00**:1–33
Published online in Wiley InterScience (www.interscience.wiley.com). DOI: 10.1002/nme

Parameter-free, weak imposition of Dirichlet boundary conditions and coupling of trimmed patches

S. Kollmannsberger[1*], A. Özcan[1], J. Baiges[2], M. Ruess[3], E. Rank[1], A. Reali[4]

[1]*Computation in Engineering, Technische Universität München; Munich, Germany*
[2]*International Center for Numerical Methods in Engineering, Universitat Politcnica de Catalunya, Barcelona, Spain*
[3]*Aerospace Structures and Computational Mechanics, Delft University of Technology, Delft, Netherlands*
[4]*Department of Civil Engineering and Architecture, University of Pavia, Italia*

SUMMARY

We present a parameter-free domain sewing approach for low- as well as high-order finite elements. Its final form contains only primal unknowns, i.e., the approach does not introduce additional unknowns at the interface. Additionally, it does not involve problem dependent parameters which require an estimation. The presented approach is symmetry-preserving, i.e. the resulting discrete form of an elliptic equation will remain symmetric and positive definite. It preserves the order of the underlying discretization and we demonstrate high order accuracy for problems of non-matching discretizations concerning the mesh size h as well as the polynomial degree of the order of discretization p. We also demonstrate how the method may be used to model material interfaces which may be curved and for which the interface does not coincide with the underlying mesh. This novel approach is presented in the context of the p- and B-spline versions of the finite cell method, an embedded domain method of high order, and compared to more classical methods such as the penalty method or Nitsche's method. Copyright © 0000 John Wiley & Sons, Ltd.

Received . . .

KEY WORDS: Dirichlet boundary conditions, embedded domain, trimmed patches, Isogeometric Analysis, p-FEM, finite cell method

1. INTRODUCTION

We reconsider the imposition of Dirichlet boundary conditions in the context of the finite element method. We focus on the case, where the Dirichlet boundary conditions are imposed independent of the underlying discretization, i.e., they do not coincide with nodes, edges or faces of the mesh but may cut elements in an arbitrary fashion.

*Correspondence to: Computation in Engineering, Technische Universität München; Munich, Germany. E-mail: kollmannsberger@bv.tu-muenchen.de

Contract/grant sponsor: S. Kollmannsberger and A. Reali greatly acknowledge the funds granted under DAAD grant No. 54577652 'Patching trimmed surfaces in isogeometric analysis'. The last author also acknowledges the funds granted under the ERC grant 'ISOBIO', No. '259229'. These grants made this contribution possible.

Journal publication: https://doi.org/10.1002/nme.4817

Whereas this problem has been investigated for decades, it has gained increased attention recently, for example in the context of Isogeometric Analysis (IGA) [1, 2]. The main motivation in IGA is to facilitate the transition from geometric models to computational analysis and thereby render mesh-generation unnecessary. This is achieved by an isoparametric approach, based on applying typical functions (e.g., B-Splines and NURBS) from Computer Aided Design (CAD) systems directly in order to discretize the differential equations that describe the underlying physics. The Dirichlet boundary conditions, however, are not always associated with the boundary of the geometric model.

Another problem class in which the need for a mesh independent imposition of Dirichlet boundary conditions arises are embedded domain methods. By definition, they do not represent the underlying physical domain with a boundary conforming mesh. In this contribution, we consider the finite cell method [3, 4], an embedded domain method of high-order which is also applicable to IGA [5].

Numerous approaches and a vast body of literature exist for the imposition of Dirichlet boundary conditions, each of them with different properties, advantages and disadvantages. None of them is perfect for all applications. Our wish-list for an ideal imposition of Dirichlet boundary conditions would require the method

- to be independent of the underlying discretization not only geometrically but such that, for example, a change of the approximation basis does not require a modification of the method;
- to be numerically stable and variationally consistent;
- to preserve a symmetric, positive definite system matrices for elliptic problems;
- to be applicable to hyperbolic and parabolic problems;
- not to introduce additional degrees of freedom;
- not to deteriorate the order of approximation of the underlying discretization;
- to be suitable for high-order accuracy;
- to be free from problem-dependent parameters.

We will first provide a quick and non-exhaustive review of some methods which comply with at least some of the properties mentioned above. For a more general review see, for example, [6, 7]. In the contribution at hand we are only interested in methods which were shown to be suitable for high-order discretizations and/or set the stage for the approach to be presented.

In principle, there are two choices for the imposition of Dirichlet boundary conditions. First, the Dirichlet boundary conditions may be directly built into the basis functions by modification, extrapolation or augmentation of the basis. Methods belonging to the first group include web-splines [8], i-splines [9, 10], max-ent interpolation [11], the extended finite element method (X-FEM), the generalized finite element method (GFEM) and level-set like approaches [12, 13, 14]. Second, one may enforce Dirichlet boundary conditions by augmenting the governing equations with suitable constraint conditions. This second group emanates from enforcing constraints in variational problems and is, thus, very general. The most commonly used is the penalty method [15, 16], which is easy to implement and which produces a positive definite system matrix for elliptic problems. However, it is not a variationally consistent method which renders the results to be strongly dependent on the choice of the penalty parameter. Additionally, the need to select a high penalty value results in an ill-conditioned system of equations.

A classic alternative to the penalty method is to enforce the constraints through Lagrange multipliers [17]. This produces a variationally consistent method with additional degrees of

Prepared using nmeauth.cls DOI: 10.1002/nme

Journal publication: https://doi.org/10.1002/nme.4817

freedom. However, the Lagrange multiplier space must satisfy the inf-sup condition [17, 18]. Thus, the choice of the discretization of the Lagrange multiplier is not always obvious. It depends on the discretization including the mesh, which turns out to be quite restrictive. The restrictions can be circumvented by means of especially developed stabilized Lagrange multiplier methods. They can be interpreted as an attempt to combine penalty and Lagrange multiplier methods and have become a popular choice in the context of embedded domain modeling [19, 20, 21, 22, 23].

Another attractive and very general alternative is Nitsche's method [24]. It can be interpreted as a stabilized Lagrange multiplier technique in which the Lagrange multiplier is directly identified to be the flux normal to the Dirichlet boundary. The close relation of Nitsche's method to the Lagrange multipliers method was first analyzed by Stenberg [25]. Even though Nitsche's method eliminates the need to discretize the Lagrange multipliers with additional variables, the stabilization term remains in the formulation and continues to contain a problem-dependent parameter. Nitsche's method is much less sensitive to the choice of a stabilization parameter than the simple penalty method. However, for optimal performance, such a parameter still needs to be estimated. A suitable estimation for a lower bound is, for example, given by Griebel and Schweitzer [26] who propose to solve an auxiliary generalized eigenvalue problem. Beyond the Poisson's equation originally investigated by Nitsche, his idea has been extended to other problem classes. The potential of Nitsche's method is nicely laid out in a review article by Hansbo [6]. It has been further extended, for example, to thin plate problems [27], three-dimensional elasticity [28], thermo-elasticity [29], and fluid dynamics [30].

Recently, different schemes have been proposed that are neither dependent on additional degrees of freedom nor on the solution of auxiliary problems to estimate the stabilization terms. Among these are schemes using degrees of freedom of certain nodes of the finite element mesh to minimize the difference between the exact and the approximated boundary condition [31]. They are robust but yield a non-symmetric equation, also for symmetric problems. In the contribution at hand, we follow a different approach, namely to use Discontinuous Galerkin techniques. Herein, the material law is discretized with functions being discontinuous across element boundaries. The additional degrees of freedom can then be condensed out at the discrete level prior to the solution of the system matrix, so they will not appear in the final formulation. These approaches can yield automatic stabilization terms, i.e., terms that add stability without the need to compute auxiliary eigenvalue problems, see e.g. [7]. The method proposed in [32] is a recently proposed variant but again leads to unsymmetric systems for symmetric problems. However, the formulation was symmetrized in [33]. This formulation will mark the point of departure in the paper at hand in which we will start by investigating its suitability for high-order embedded domain discretizations.

Imposing Dirichlet boundary conditions is also of importance for weak coupling of different discretizations. Geometric models in CAD systems, for example, are usually comprised of many trimmed patches of splines of all types, such as B-Splines, NURBS, T-Splines, etc. These trimmed patches should serve as the geometric model in IGA. If this model is to be directly transferred to computational analysis without mesh generation, the trimmed patches have to be glued at their common and usually non-matching interface by means of enforcing continuity of the unknowns and/or their derivatives to yield a consistent computational model.

The same task needs to be addressed in embedded domain analysis. They are popular for geometries which are difficult to mesh or for cases in case a geometry and/or topology changes

Journal publication: https://doi.org/10.1002/nme.4817

over time. Chimera approaches, for example, resolve the domain with one or multiple overlapping meshes each of which represents a different physical domain. The continuity at their common boundary is then enforced either iteratively or monolithically. Throughout this treatise, we only consider monolithic coupling, i.e., the coupling terms will directly appear in the resulting system matrix and the solution of the coupled system is obtained in one step only.

Monolithic approaches using Lagrange multipliers to satisfy continuity constraints along the domain interfaces are mostly named Mortar methods. Many variants of this technique have been proposed [34, 35, 36, 37, 38], to list just a few. As in the case of the weak imposition of Dirichlet boundary conditions with Lagrange multipliers, its straightforward application results in rather restrictive stability conditions, which is why stabilized versions are advocated. Again, an alternative for monolithic coupling is to use Nitsche's method to impose the constraints. The use of Nitsche's method to couple subdomains was extensively analysed in [39, 40]. Hansbo et al. [41] extended this concept to overlapping domains. It was also used to represent strong and weak singularities [42]. The concept of using Nitsche's method for coupling domains has been applied to a variety of fields such as linear elasticity [43, 44], contact [45] and fluid-structure interaction [46]. As in the case of imposing Dirichlet boundary conditions, using Nitsche's idea for coupling requires the solution of an auxiliary generalized eigenvalue problem for an estimation of the optimal stabilization parameter. In this contribution, we derive a method which sets out to eliminate the need for solving auxiliary equations to estimate stabilization parameters and, at the same time, leads to symmetric system matrices for elliptic problems.

In order to set up our main field of application of our proposed technique we begin with the introduction of the finite cell method in section 2 before continuing with the weak imposition of Dirichlet boundary conditions in section 3, embarking from classical Nitsche's formulation. Next, we recall the recently introduced boundary conditions for Poisson's problem as presented in [33] but put them in the context of the finite cell method. We show that the concept in [33] can be straightforwardly utilized in the framework of high-order finite elements in general. We investigate the performance of the boundary conditions and compare their results to the penalty method as well as to Nitsche's method for Poisson's equation in one and two dimensions. The two-dimensional examples are computed by discretizing the continuous problems with the classical p-version of the finite element method utilizing the hierarchical integrated Legendre polynomials as introduced in [47]. Additionally, we evaluate the performance of this new type of boundary conditions in the context of patches of B-Splines which form the basis for Isogeometric Analysis.

Section 4 is devoted to the new, parameter-free and symmetry-preserving weak coupling of domains. Again, we start by recalling the more classical Nitsche's method for the problem under consideration. The derivation of the new scheme is carried out in sections 4.3 and 4.4. We prove the scheme to be stable in section 4.5 and show that in all tested cases the resulting discrete system is positive definite. Utilizing the presented methodology, we demonstrate high convergence rates in all cases. Specifically, we first evaluate the new method's capabilities for Poisson's equation with non-conform discretizations in h and p and constant coefficients in a boundary-conform setting (section 4.6.1) before we move to an overlapping discretization with an inclined coupling boundary (section 4.6.2). We then demonstrate the method for a bimetal strip. Naturally, this example possesses discontinuous coefficients and we apply non-conforming discretizations in h and p (section 4.6.3). The last example is an inclusion problem. It has discontinuous coefficients and

Prepared using nmeauth.cls DOI: 10.1002/nme

Journal publication: https://doi.org/10.1002/nme.4817

we discretize it in a Chimera-like fashion by using an overlapping discretizations (section 4.6.4). All examples are computed by using B-Spline patches and classical p-FEM and are compared to Nitsche's method. Finally, we draw conclusions in section 5.

2. FINITE CELL METHOD

2.1. Applications of the finite cell method

The finite cell method was first introduced in [3, 4], where its potential was demonstrated for linear-elastic examples in two and three dimensions. Various extensions of the FCM confirm its versatility in the context of topology optimization [48], geometrically nonlinear continuum mechanics [49], adaptive mesh-refinement [50, 51, 52], computational steering [53, 54], biomedical engineering [55], numerical homogenization [56], elastoplasticity [57], wave propagation in heterogeneous materials [58], local enrichment for material interfaces [14], convection-diffusion problems [59, 60], thin-walled structures [61], design-through-analysis and isogeometric analysis [51, 52, 5], and multi-physical applications [29].

A free FCM-MATLAB-toolbox is provided at `http://fcmlab.cie.bgu.tum.de/` under the GNU Licence along with extensive online documentation [62] as well as numerous examples. It serves as a library to solve one-, two-, and three-dimensional elasto-static and elasto-dynamic problems. It utilizes advanced software development techniques such as Design Patterns [63] to provide maximum flexibility to users as well as Unit Testing to ensure code consistency throughout the development process.

2.2. Formulation

The finite cell method is an embedded domain method based on higher order shape functions. The primary goal of the method is to avoid the generation of boundary-conforming meshes. Nevertheless, it provides high-order convergence rates. For this purpose, it assumes a three-dimensional physical domain Ω and an embedding domain Ω_e such that $\Omega_e \supset \Omega$ whereby Ω_e is of a simpler, typically rectangular shape. Their boundaries are defined as $\partial\Omega = \Gamma$ and $\partial\Omega_e = \Gamma_e$.

Let us consider the following bilinear form $(\cdot,\cdot)_\Omega$ defined on Ω_e:

$$(\boldsymbol{u},\boldsymbol{v})^{\alpha}_{\Omega_e} = \int_{\Omega_e} [\boldsymbol{L}\boldsymbol{v}]^T \boldsymbol{C}^{\alpha}_e [\boldsymbol{L}\boldsymbol{u}] \, d\Omega \tag{1}$$

in which $\boldsymbol{u}$ is a scalar- or vector-valued function, $\boldsymbol{v}$ is a test function, and $\boldsymbol{L}$ is a differential operator. The material matrix of the embedding domain is defined as $\boldsymbol{C}^{\alpha}_e = \alpha\boldsymbol{C}$ where $\boldsymbol{C}$ is the material matrix of the physical domain Ω and α is an indicator function defined as:

$$\alpha(\boldsymbol{x}) = \begin{cases} 1 & \forall \boldsymbol{x} \in \Omega \\ 0 & \forall \boldsymbol{x} \in \Omega_e \setminus \Omega \end{cases} \tag{2}$$

The finite cell method is depicted in fig. 1. It is important to note that eq. (2) implies $\alpha(\boldsymbol{x}) = 1.0 \quad \forall \boldsymbol{x} \in \Gamma$ as, otherwise, it would not be possible to apply non homogeneous boundary conditions

Journal publication: https://doi.org/10.1002/nme.4817

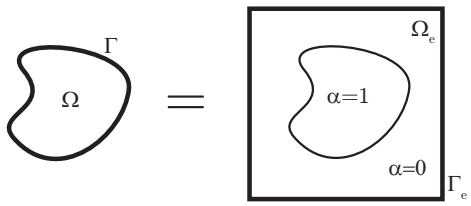


Figure 1. The finite cell method: Physical domain Ω and embedding domain Ω_e.

on Γ. Furthermore, the definition of $\alpha(\boldsymbol{x})$ in eq. (2) ensures that the bilinear forms defined on the embedding domain Ω_e and on the physical domain Ω are equivalent:

$$(\boldsymbol{u},\boldsymbol{v})^{\alpha}_{\Omega_e} = \int_{\Omega_e} [\boldsymbol{L}\boldsymbol{v}]^T \alpha \boldsymbol{C} [\boldsymbol{L}\boldsymbol{u}] \, d\Omega \tag{3}$$

$$= \int_{\Omega} [\boldsymbol{L}\boldsymbol{v}]^T \boldsymbol{C} [\boldsymbol{L}\boldsymbol{u}] \, d\Omega + \int_{\Omega_e \setminus \Omega} [\boldsymbol{L}\boldsymbol{v}]^T \boldsymbol{0} [\boldsymbol{L}\boldsymbol{u}] \, d\Omega \tag{4}$$

$$= \int_{\Omega} [\boldsymbol{L}\boldsymbol{v}]^T \boldsymbol{C} [\boldsymbol{L}\boldsymbol{u}] \, d\Omega \tag{5}$$

$$= (\boldsymbol{u},\boldsymbol{v})_{\Omega} \tag{6}$$

The linear functional $\mathcal{F}$

$$\langle \mathcal{F}, \boldsymbol{v} \rangle = \int_{\Omega_e} \boldsymbol{v}^T (\alpha \boldsymbol{f}) \, d\Omega + \int_{\Gamma_n} \boldsymbol{v}^T \boldsymbol{t} \, d\Gamma \equiv \langle \mathcal{F}, \boldsymbol{v} \rangle^{\alpha}_{e} \tag{7}$$

is identical for the extended and the physical domain and also takes the volume sources $\boldsymbol{f}$ inside Ω and prescribed fluxes $\bar{\boldsymbol{t}} = \boldsymbol{\sigma} \cdot \boldsymbol{n}$ on the Neumann part of the boundary of the physical domain Γ into account. The solution of

$$(\boldsymbol{u},\boldsymbol{v})^{\alpha}_{\Omega_e} = \langle \mathcal{F}, \boldsymbol{v} \rangle^{\alpha}_{e} \tag{8}$$

is equivalent to solving $(\boldsymbol{u},\boldsymbol{v})_{\Omega} = \langle \mathcal{F}, \boldsymbol{v} \rangle$. In the sequel we will, therefore, drop the subindex $_e$ unless we want to specifically stress one viewpoint or the other.

A discretized view of the finite cell method is depicted in fig. 2. For the numerical approximation of eq. (8) it is convenient to define a computational grid on the extended domain Ω_e. This grid forms the support of the basis functions. In principle, we can use a uniform Cartesian grid, as was applied in the first implementations of the FCM [3, 4]. Hierarchically refined grids can be defined alike [64, 65, 66], if local features of the solution are to be resolved in more detail. Note that the boundary of the physical domain is not resolved or even approximated by the grid. In order not to change the problem, the geometry needs to be recovered at the integration level. A natural strategy is the following:

Let us denote elements of the computational grid such as cells Ω_C to distinguish them from classical finite elements. It is convenient to categorize cells into three different types which are depicted in fig. 2:

DOI: 10.1002/nme

Journal publication: https://doi.org/10.1002/nme.4817

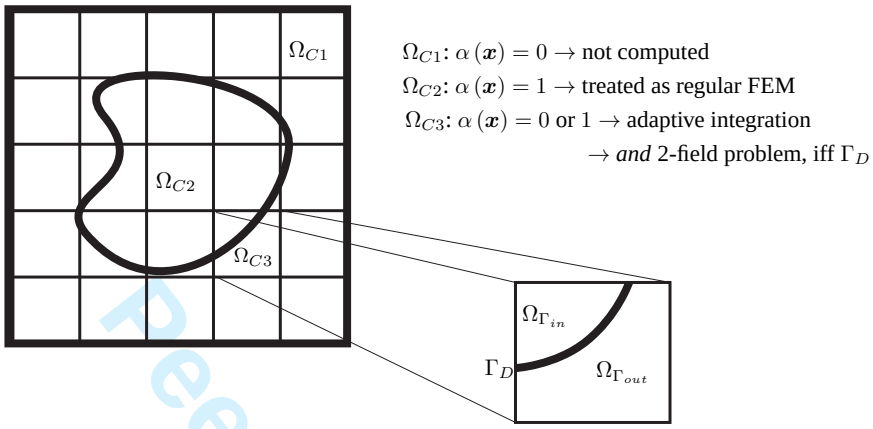


Figure 2. The discrete point of view of the finite cell method. The cells Ω_C provide the support of the shape functions much like finite elements do. The important case is Ω_{C3} In addition to an adaptive integration we advocate solving the 2-field problem on Ω_{C3} if and only if a Dirichlet boundary condition is imposed on Γ parts of it (see section 3.2.). The part of Γ on which Dirichlet boundary conditions are imposed is labeled Γ_D.

1. Ω_{C1}: Cells where $\alpha(\boldsymbol{x}) = 0 \quad \forall \boldsymbol{x} \in \Omega_e$, which are not computed.
2. Ω_{C2}: Cells where $\alpha(\boldsymbol{x}) = 1 \quad \forall \boldsymbol{x} \in \Omega_e$, which are computed as if they were regular finite elements.
3. Ω_{C3}: Cells cut by the boundary Γ, which receive special treatment as described in the following.

Cells of type Ω_{C3} are decomposed into two parts $\Omega_{\Gamma_{in}}$ in which $\alpha(\boldsymbol{x}) = 1$, and $\Omega_{\Gamma_{out}}$ in which $\alpha(\boldsymbol{x}) = 0$. In practical computations, we assume the parameter $\alpha(\boldsymbol{x})$ not to be exactly zero. Instead, we chose $\epsilon \ll 1$ to ease numerical round-off issues in the solution process. Thus, we explicitly accept a modeling error of the order ϵ which is controllable as it tends to zero if ϵ approaches zero. We usually choose ϵ to have very small magnitude, typically about 10^{-8} smaller than the material constant.

The numerical integration of the bilinear form must be carried out accurately enough on the cut cells as to recover the bilinear form of eq. (6) at the discrete level. For this purpose, the integration of the bilinear form is ideally only carried out on $\Omega_{\Gamma_{in}}$. One way to perform the domain-integration numerically is to approximate $\Omega_{\Gamma_{in}}$ adaptively by recursive bisection, allowing for a simple and efficient refinement towards the boundary [64, 5, 67]. Geometrically more involved, but also more efficient schemes utilize straight sided triangles, —see, e.g., [28]. Precise numerical integration schemes for complex geometries based on the blending function method [68] are also possible.

Homogeneous Neumann boundary conditions are easy to deal with. They are equivalent to assuming material with zero stiffness in the domain $\Omega_e \setminus \Omega$ and, therefore, need no special treatment in the framework of the finite cell method. Inhomogeneous Neumann boundary conditions are realized by including the second term in eq. (7), i.e., by directly integrating over Γ. The integrand

Journal publication: https://doi.org/10.1002/nme.4817

is simply a product of a function living on the surface (the load) and the shape functions $\boldsymbol{u}$ defined over the cells, yet evaluated at the surface Γ.

The imposition of Dirichlet type boundary conditions on Γ or on parts of it, i.e., Γ_D cuts Ω_{C3}, is a central topic of this paper and is discussed in section 3.

3. WEAK IMPOSITION OF DIRICHLET BOUNDARY CONDITIONS

3.1. Nitsche's method

We consider the following scalar-valued single field problem:

$$-k\Delta u = f \quad \text{in } \Omega \tag{9}$$

$$u = \overline{u} \quad \text{on } \Gamma = \partial\Omega \tag{10}$$

For simplicity, we assume a pure Dirichlet-problem, i.e. $\Gamma_D = \Gamma$. If Dirichlet conditions are imposed only on parts of the boundary, corresponding integrals in the following formulations have to be taken only over these parts. If the boundary conditions eq. (10) are imposed by Nitsche's method [24], the potential corresponding to eqs. (9) and (10) reads:

$$\Pi\left(u,\boldsymbol{\sigma}\right) = \int_{\Omega}\left(\frac{1}{2}k\left(\nabla u\right)^2 - fu\right)d\Omega - \int_{\Gamma}\boldsymbol{n}\cdot\boldsymbol{\sigma}\left(u-\overline{u}\right)d\Gamma + \frac{1}{2}\beta\int_{\Gamma}\left(u-\overline{u}\right)^2 d\Gamma \tag{11}$$

The first term in eq. (11) is clearly the potential associated to eq. (9). The second term enforces the boundary conditions of eq. (10) by means of a Lagrange multiplier $\boldsymbol{\lambda}$ which has already been identified to be equal to the fluxes at the boundary $\boldsymbol{\lambda} = -\boldsymbol{n}\cdot\boldsymbol{\sigma}$. This allows a formulation in primal variables, i.e., no additional degrees of freedom are needed to approximate $\boldsymbol{\lambda}$. However, the negative sign of the Lagrange multiplier may lead to a loss of coercivity which is restored by the third term in the potential of eq. (11) which again enforces u to be equal to $\overline{u}$ at Γ by means of a penalization. Coercivity is only restored for a sufficiently high choice of β. Unfortunately, a closer look reveals that the penalty parameter β is problem-dependent. Fortunately, it has turned out that the precise choice of β does not have a large influence on the solution above a certain threshold. However, an optimal value for β is not easily obtained and the threshold above which its influence on the result is of only minor importance is unknown a priori.

The weak form of the imposition of the boundary conditions by Nitsche's method may either be obtained by multiplication of eqs. (9) and (10) by a suitable test function and integration by parts, or by variation of the potential given in eq. (11). In any case, both approaches lead to the following weak form [24]:

$$k\left(\nabla u,\nabla v\right)_{\Omega} - k\left(\boldsymbol{n}\cdot\nabla u, v\right)_{\Gamma} - k\left(u,\boldsymbol{n}\cdot\nabla v\right)_{\Gamma} + k\beta\left(u,v\right)_{\Gamma} = \left\langle f,v\right\rangle_{\Omega} - k\langle\overline{u},\boldsymbol{n}\cdot\nabla v\rangle_{\Gamma} + k\beta\langle\overline{u},v\rangle_{\Gamma} \tag{12}$$

where we used the common short hand notation of the L^2 product $(\cdot,\cdot)$ and $<\cdot,\cdot>$ denotes the integral of two functions over a domain.

Journal publication: https://doi.org/10.1002/nme.4817

A lower bound to estimate the parameter β in eq. (12) to ensure coercivity may be obtained by solving the following auxiliary eigenvalue problem as, for example, suggested in [69, 28]:

$$\boldsymbol{A}\boldsymbol{x} = \Lambda \boldsymbol{B}\boldsymbol{x} \tag{13}$$

$$\boldsymbol{A} = \int_\Gamma (\nabla u \cdot \boldsymbol{n})(\nabla v \cdot \boldsymbol{n}) d\Gamma$$

$$\boldsymbol{B} = \int_\Omega (\nabla u \cdot \nabla v) d\Omega$$

3.2. *Alternative to Nitsche's method*

The symmetric form of the imposition of boundary conditions presented in this section closely follows the work of Baiges et al. [33] but puts the methodology into the context of the finite cell method. Even though the presentation starts from a mixed-field formulation, its final form contains only the primal unknowns.

3.2.1. Weak form Equations (9) and (10) can be recast into the following two-field problem:

$$-k\Delta u = f \qquad \text{in } \Omega \tag{14}$$

$$\frac{1}{k}\boldsymbol{\sigma} = \nabla u \qquad \text{in } \Omega_{\Gamma_{in}} \tag{15}$$

$$u = \overline{u} \qquad \text{on } \Gamma \tag{16}$$

As usual for mixed-field formulations, u and $\boldsymbol{\sigma}$ in eqs. (14) to (16) are discretized independently. The associated potential to eqs. (14) to (16) is

$$\Pi(u, \boldsymbol{\sigma}) = \int_\Omega \left(\frac{1}{2} k (\nabla u)^2 - f u \right) d\Omega - \int_\Gamma \boldsymbol{n} \cdot \boldsymbol{\sigma} (u - \overline{u}) \, d\Gamma - \frac{1}{2nk} \int_{\Omega_{\Gamma_{in}}} (\boldsymbol{\sigma} - k\nabla u)^2 \, d\Omega_{\Gamma_{in}} \tag{17}$$

where n is any real number greater than two. The only difference between eq. (11) and eq. (17) is that the latter enforces the boundary condition with the help of the material law of eq. (15) which is imposed weakly on $\Omega_{\Gamma_{in}}$. In essence, the potential eq. (17) enforces the Lagrange multiplier to be the trace of the unknown in a least squares sense.

The variation of the potential given in eq. (17) leads to the following weak form of eqs. (14) to (16):

$$k(\nabla u, \nabla v)_\Omega - \langle \boldsymbol{n} \cdot \boldsymbol{\sigma}, v \rangle_\Gamma + \frac{1}{n} (\nabla v, \boldsymbol{\sigma})_{\Omega_{\Gamma_{in}}} - \frac{1}{n} k (\nabla v, \nabla u)_{\Omega_{\Gamma_{in}}} = \langle f, v \rangle_\Omega \tag{18}$$

$$-\frac{1}{nk} (\boldsymbol{\tau}, \boldsymbol{\sigma})_{\Omega_{\Gamma_{in}}} + \frac{1}{n} (\boldsymbol{\tau}, \nabla u)_{\Omega_{\Gamma_{in}}} - \langle \boldsymbol{\tau} \cdot \boldsymbol{n}, u \rangle_\Gamma = -\langle \boldsymbol{\tau} \cdot \boldsymbol{n}, \overline{u} \rangle_\Gamma \tag{19}$$

This formulation still contains the primal unknown u and the flux variable $\boldsymbol{\sigma}$. However, as we shall see later, $\boldsymbol{\sigma}$ can be condensed out at the discrete level in such a way that no additional unknowns will be necessary to impose the boundary conditions.

DOI: 10.1002/nme

Journal publication: https://doi.org/10.1002/nme.4817

3.2.2. Discretization in the context of the finite cell method. The finite element subspaces are constructed to ensure that $V_h \subset H^1(\Omega_h)^d$, so V_h consists of continuous functions. Further, we construct $S_h \subset L^2(\Omega_h)^d$. Functions in S_h are considered to be discontinuous only within the elements cut by Γ and are zero elsewhere. The discretized weak form given in eqs. (18) and (19) now reads:

$$k\left(\nabla u_h, \nabla v_h\right)_\Omega - \langle \boldsymbol{n}\cdot\boldsymbol{\sigma}_h, v_h\rangle_\Gamma + \frac{1}{n}\left(\nabla v_h, \boldsymbol{\sigma}_h\right)_{\Omega_{\Gamma_{in}}} - \frac{1}{n}k\left(\nabla v_h, \nabla u_h\right)_{\Omega_{\Gamma_{in}}} = \langle f, v_h\rangle_\Omega \qquad \forall v_h \in V_h \tag{20}$$

$$-\frac{1}{nk}\left(\boldsymbol{\tau}_h, \boldsymbol{\sigma}_h\right)_{\Omega_{\Gamma_{in}}} + \frac{1}{n}\left(\boldsymbol{\tau}_h, \nabla u_h\right)_{\Omega_{\Gamma_{in}}} - \langle \boldsymbol{\tau}_h\cdot\boldsymbol{n}, u_h\rangle_\Gamma = -\langle \boldsymbol{\tau}_h\cdot\boldsymbol{n}, \overline{u}\rangle_\Gamma \quad \forall \boldsymbol{\tau}_h \in S_h \tag{21}$$

At this point it is worthwhile to discuss the meaning of discontinuity in the context of the finite cell method. The primary goal is to simplify the implementation process as far as possible by reusing the shape functions from V_h for S_h. In the context of the finite cell method, the first term in eq. (20) reads

$$\left(\nabla u_h, \nabla v_h\right)^\alpha_{\Omega_e} = \sum_{c=1}^{n_c}\int_{\Omega_c}\left[\boldsymbol{L}v_h\right]^T \alpha\boldsymbol{C}\left[\boldsymbol{L}u_h\right]\, d\Omega \tag{22}$$

The following, alternative view is naturally possible:

$$\left(\nabla u_h, \nabla v_h\right)^\alpha_{\Omega_e} = \sum_{c=1}^{n_c}\int_{\Omega_c}\left[\boldsymbol{L}v_h\sqrt{\alpha}\right]^T \boldsymbol{C}\left[\boldsymbol{L}u_h\sqrt{\alpha}\right]\, d\Omega \tag{23}$$

Hereby, the shape and the weighting functions $\boldsymbol{u}$ and $\boldsymbol{v}$ are both multiplied by the square root of the indicator function α defined in eq. (2). We now have two views on the discontinuity of $\boldsymbol{u}$ and $\boldsymbol{v}$: (a) we may view them to *smoothly extend* across Γ into the embedding domain under the condition that this extension does not contribute to the energy described by the bilinear form. Clearly, this view is expressed in eq. (22). (b) we might view the shape functions u_h and v_h to exhibit a jump to zero from whatever value that satisfies the conditions on Γ internal to Ω_{C3}. This view is represented in eq. (23). From the implementational point of view there is, thus, no need to define a new set of discontinuous shape functions for the discretization of either $\boldsymbol{\sigma}_h$ or $\boldsymbol{\tau}_h$. Instead, one may simply reuse the shape functions from the discretization of u_h and v_h for the discretization of $\boldsymbol{\sigma}_h$ and $\boldsymbol{\tau}_h$. Thus, we discretize u_h and v_h in a Bubnov-Galerkin sense as follows:

$$u_h = \boldsymbol{N}\boldsymbol{U} = \begin{bmatrix} N_{i=1} & \dots & N_{i=n}\end{bmatrix}\begin{bmatrix}\hat{u}_{i=1}\\ \vdots \\ \hat{u}_{i=n}\end{bmatrix} \tag{24}$$

$$v_h = \boldsymbol{N}\boldsymbol{V} = \begin{bmatrix} N_{i=1} & \dots & N_{i=n}\end{bmatrix}\begin{bmatrix}\hat{v}_{i=1}\\ \vdots \\ \hat{v}_{i=n}\end{bmatrix} \tag{25}$$

$$\boldsymbol{\sigma}_h = \boldsymbol{N}\boldsymbol{\Sigma} = \begin{bmatrix} N_{i=1} & \dots & N_{i=n} \\ N_{i=1} & \dots & N_{i=n}\end{bmatrix}\begin{bmatrix}\hat{\sigma}_{i=1x} & \hat{\sigma}_{i=1y}\\ \vdots & \vdots \\ \hat{\sigma}_{i=nx} & \hat{\sigma}_{i=ny}\end{bmatrix} \tag{26}$$

DOI: 10.1002/nme

Journal publication: https://doi.org/10.1002/nme.4817

variational eqn.	$k\left(\nabla u_h, \nabla v_h\right)_{\Omega,\Omega_\Gamma in}$	$-\frac{1}{nk}\left(\boldsymbol{\tau}_h, \boldsymbol{\sigma}\right)_{\Omega_{\Gamma_{in}}}$	$\frac{1}{n}\left(\boldsymbol{\tau_h}, \nabla u\right)_{\Omega^I_{\Gamma_{in}}}$	$\frac{1}{n}\left(\nabla v_h, \boldsymbol{\sigma}_h\right)_{\Omega_{\Gamma_{in}}}$
algebraic form	$\boldsymbol{K}_{uu}\boldsymbol{U}$	$\boldsymbol{K}_{\sigma\sigma}\boldsymbol{\Sigma}$	$\boldsymbol{K}_{\sigma u}\boldsymbol{U}$	$\boldsymbol{K}_{u\sigma}\boldsymbol{\Sigma}$
variational eqn.	$-\langle\boldsymbol{\sigma}_h\cdot\boldsymbol{n}, v_h\rangle_\Gamma$	$-\langle\boldsymbol{\tau}_h\cdot\boldsymbol{n}, u\rangle_\Gamma$	$\text{-}\langle\boldsymbol{n}\cdot\boldsymbol{\tau}_h, \bar{u}\rangle_\Gamma$	$\langle f, v_h\rangle_\Omega$
algebraic form	$\boldsymbol{G}_{u\sigma}\boldsymbol{\Sigma}$	$\boldsymbol{G}_{\sigma u}\boldsymbol{U}$	$\boldsymbol{g}_{\sigma\bar{u}}$	$\boldsymbol{f}$

Table I. Terms concerned with the internal part of each domain of the discrete variational and algebraic form of eqs. (20) and (21).

$$\boldsymbol{\tau}_h = \boldsymbol{nNT} = \begin{bmatrix} \hat{n}_1 \\ \hat{n}_2 \end{bmatrix} \begin{bmatrix} N_{i=1} & \dots & N_{i=n} \end{bmatrix} \begin{bmatrix} \hat{\tau}_{i=1\,x} & \hat{\tau}_{i=1\,y} \\ \vdots & \vdots \\ \hat{\tau}_{i=n\,x} & \hat{\tau}_{i=n\,y} \end{bmatrix} \tag{27}$$

for scalar-valued Poisson's problem. In eqs. (24) to (27) N_i denote the i^{th} shape function and $n = (p+1)^d$ for a tensor product space in d dimensions. The corresponding coefficients $\hat{u}_i, \hat{v}_i, \hat{\sigma}_{ix}, \hat{\sigma}_{iy}, \hat{\tau}_{ix}, \hat{\tau}_{iy}$ are organized in the respective matrices $\boldsymbol{U}$, $\boldsymbol{V}$, $\boldsymbol{\Sigma}$, and $\boldsymbol{T}$. We would like to emphasize that, in the following, we will use an equal order discretization for all fields and that, therefore, all $\boldsymbol{N}$'s given in eqs. (24) to (27) are completely identical.

Note that the definition of $\boldsymbol{\tau}_h$ includes the multiplication by the normal vector $\boldsymbol{n}$. We apply the following definition for $\boldsymbol{n}$: $\boldsymbol{n}$ at Γ_D is the outward pointing normal vector. Logically, then, inside $\Omega_{\Gamma_{in}}$, $\boldsymbol{n}$ is defined to deliver a smooth extension of Γ into $\Omega_{\Gamma_{in}}$ which is obtained by taking $\boldsymbol{n}$ to be the normal vector towards Γ from the considered point in $\Omega_{\Gamma_{in}}$[†].

3.2.3. Static condensation and final formulation. With the definitions given in table I the system given by eqs. (20) and (21) can be written in compact form as:

$$\begin{bmatrix} \left(1-\frac{1}{n}\right)\boldsymbol{K}_{uu} & \boldsymbol{K}_{u\sigma}+\boldsymbol{G}_{u\sigma} \\ \boldsymbol{K}_{\sigma u}+\boldsymbol{G}_{\sigma u} & \boldsymbol{K}_{\sigma\sigma} \end{bmatrix} \begin{bmatrix} \boldsymbol{U} \\ \boldsymbol{\Sigma} \end{bmatrix} = \begin{bmatrix} \boldsymbol{f} \\ \boldsymbol{g}_{\sigma\bar{u}} \end{bmatrix} \tag{28}$$

The fluxes may be computed as follows:

$$\boldsymbol{\Sigma} = \boldsymbol{K}_{\sigma\sigma}^{-1}\left(-\left(\boldsymbol{K}_{\sigma u}+\boldsymbol{G}_{\sigma u}\right)\boldsymbol{U}+\boldsymbol{g}_{\sigma\bar{u}}\right) \tag{29}$$

$\boldsymbol{K}_{\sigma\sigma}$ is always positive definite and, therefore, invertible. Additionally, $\boldsymbol{K}_{\sigma\sigma}$ must only be computed on the elements cut by Γ. The fluxes may, thus, be condensed out of eq. (28) analytically on element level prior to computation. Equation (28), therefore, simplifies to:

$$\left[\left(1-\frac{1}{n}\right)\boldsymbol{K}_{uu}-\left(\boldsymbol{G}_{u\sigma}+\boldsymbol{K}_{u\sigma}\right)\boldsymbol{K}_{\sigma\sigma}^{-1}\left(\boldsymbol{K}_{\sigma u}+\boldsymbol{G}_{\sigma u}\right)\right]\boldsymbol{U} = \left[\boldsymbol{f}-\left(\boldsymbol{G}_{u\sigma}+\boldsymbol{K}_{u\sigma}\right)\boldsymbol{K}_{\sigma\sigma}^{-1}\boldsymbol{g}_{\sigma\bar{u}}\right] \tag{30}$$

[†]This definition is applied, for example, in the evaluation of $\boldsymbol{K}_{\sigma\sigma}$ in eqs. (35), (53) and (54).

Journal publication: https://doi.org/10.1002/nme.4817

variational eqn.	$-k\left(\boldsymbol{n}\cdot\nabla u, v\right)_\Gamma$	$\frac{\beta k}{h}\left(v_h, u_h\right)_\Gamma$	$-k\left(\boldsymbol{n}\nabla v_h, \bar{u}\right)_\Gamma$	$\frac{\beta k}{h}\left(v_h, \bar{u}\right)_\Gamma$
algebraic form	$\boldsymbol{G}_{uu}\boldsymbol{U}$	$\boldsymbol{G}^{\beta}_{uu}\boldsymbol{U}$	$\boldsymbol{g}_{u\bar{u}}$	$\boldsymbol{g}^{\beta}_{u\bar{u}}$

Table II. Nitsche's method: Linear and bilinear forms used in eq. (36).

The system given in eq. (30) can be written in a simpler way with the help of the following identities:

$$\frac{1}{n}\boldsymbol{K}_{uu}\boldsymbol{U} = -\left[\boldsymbol{K}_{u\sigma}\boldsymbol{K}^{-1}_{\sigma\sigma}\boldsymbol{K}_{\sigma u}\right]\boldsymbol{U} \tag{31}$$

$$\boldsymbol{G}^{T}_{uu}\boldsymbol{U} = -\left[\boldsymbol{K}_{u\sigma}\boldsymbol{K}^{-1}_{\sigma\sigma}\boldsymbol{G}_{\sigma u}\right]\boldsymbol{U} \tag{32}$$

$$\boldsymbol{g}_{u\bar{u}}\boldsymbol{U} = -\left[\boldsymbol{K}_{u\sigma}\boldsymbol{K}^{-1}_{\sigma\sigma}\boldsymbol{g}_{\sigma\bar{u}}\right] \tag{33}$$

$$\boldsymbol{G}_{uu}\boldsymbol{U} = -\left[\boldsymbol{G}_{u\sigma}\boldsymbol{K}^{-1}_{\sigma\sigma}\boldsymbol{K}_{\sigma u}\right]\boldsymbol{U} \tag{34}$$

to give:

$$\left[\boldsymbol{K}_{uu} + \boldsymbol{G}_{uu} + \boldsymbol{G}^{T}_{uu} - \boldsymbol{G}_{u\sigma}\boldsymbol{K}^{-1}_{\sigma\sigma}\boldsymbol{G}_{\sigma u}\right]\boldsymbol{U} = \left[\boldsymbol{f} + \boldsymbol{g}_{\sigma\bar{u}} - \boldsymbol{G}_{\bar{u}\sigma}\boldsymbol{K}^{-1}_{\sigma\sigma}\boldsymbol{g}_{\sigma\bar{u}}\right] \tag{35}$$

Equation (35) is the final formulation of the method presented in [33]. For means of comparison, the discrete form of Nitsche's method is given in eq. (36)

$$\left[\boldsymbol{K}_{uu} + \boldsymbol{G}_{uu} + \boldsymbol{G}^{T}_{uu} + \boldsymbol{G}^{\beta}_{uu}\right]\boldsymbol{U} = \left[\boldsymbol{f} + \boldsymbol{g}_{u\bar{u}} + \boldsymbol{g}^{\beta}_{u\bar{u}}\right] \tag{36}$$

where the linear forms of $\boldsymbol{g}_{u\bar{u}}$ and $\boldsymbol{g}^{\beta}_{u\bar{u}}$ and the bilinear forms of $\boldsymbol{G}^{T}_{uu}$ and $\boldsymbol{G}^{\beta}_{uu}$ are given in table II. Now it can be clearly seen that eq. (35) and eq. (36) only differ in the last terms in the brackets. The new terms $\boldsymbol{G}_{u\sigma}\boldsymbol{K}^{-1}_{\sigma\sigma}\boldsymbol{G}_{\sigma u}$ and $\boldsymbol{G}_{\bar{u}\sigma}\boldsymbol{K}^{-1}_{\sigma\sigma}\boldsymbol{g}_{\sigma\bar{u}}$ are now responsible for counterbalancing the Nitsche terms resulting from the identified Lagrange multiplier $\boldsymbol{G}_{uu}$ and its transpose. Stability of eq. (35) was proven in the sense of an inf-sup condition in [33] for low order discretizations. We will provide examples to show that the system given by eq. (35) even remained positive definite.

It is interesting to note how the method given by eq. (35) avoids the estimation of a problem-dependent parameter. Let us consider the simple case in which $\Omega_{\Gamma in}$ is small. Then, the magnitude of $\boldsymbol{K}^{-1}_{\sigma\sigma}$ is large and the term $\boldsymbol{G}_{u\sigma}\boldsymbol{K}^{-1}_{\sigma\sigma}\boldsymbol{G}_{\sigma u}$ automatically adds the needed stabilization. In this process, the dependency of the penalty term on a characteristic mesh size h and a specific p is respected quite naturally.

3.3. Two-dimensional Poisson's problem

In this section, we investigate the 2D Poisson's problem proposed in [70]. The same example was studied in the context of the finite cell method in [28] as well, and is depicted in fig. 3. It consists of a square domain with edge $a = 1$ and $k = 1$. Homogeneous Dirichlet boundary conditions are imposed on the left, right and upper boundary of the domain while $u(x, 0) = sin(\pi x)$ is imposed on the lower boundary. We discretize the domain by 8×8 elements using both the p-version and the B-Spline version of the finite cell method and study pure p-refinements, keeping the discretization grid as it is. All boundary conditions are imposed weakly as presented above either by the Nitsche's method or by eq. (35). For Nitsche's method, we estimate the penalty parameter for each polynomial degree as suggested in [69] and multiply it by two. We expect the solution to converge to the

Journal publication: https://doi.org/10.1002/nme.4817

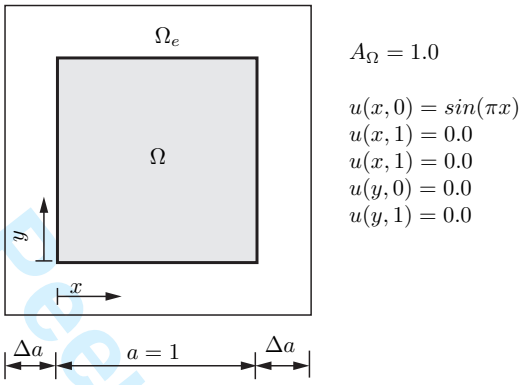


Figure 3. two-dimensional Poisson's problem as proposed in [70].

analytical one given in eq. (37) which has the strain energy given by eq. (38):

$$U_{ex}(x,y) = (cosh(\pi y) - coth(\pi)\, sinh(\pi y))\, sin(\pi x) \tag{37}$$

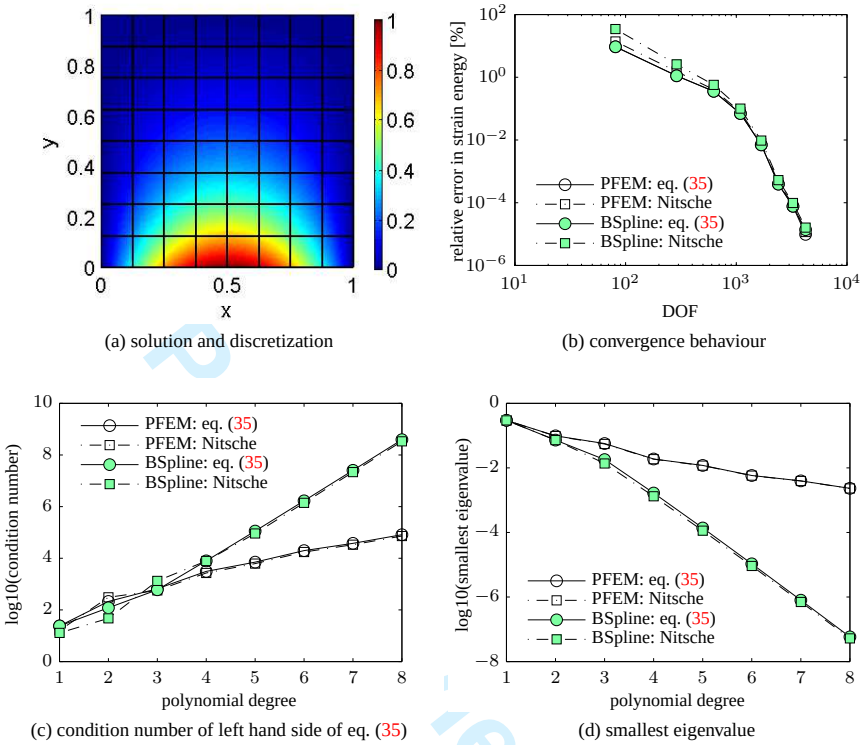
$$U_{ex} = \frac{k}{4}\pi\, coth(\pi) \tag{38}$$

With no singularities in the solution, we expect an exponential convergence in the energy norm under pure p-refinement.

We begin by studying the case where the discretization of the domain conforms to all boundaries. A first overall impression of the solution is given in fig. 4a, where the case $p = 3$ is considered. The exact solution is not depicted as the approximation is already so close to it. More insight is only gained by looking at the convergence of its energy, as depicted in fig. 4b. All methods converge exponentially as expected. With no even-odd behavior present, almost no difference in the p- or B-Spline version of the finite cell method is observable. This is not surprising as we are performing a pure p-extension on both meshes starting with p=1 and, therefore, stay C^0 continuous at inter-element level for both discretization schemes. It is explicitly pointed out that at p=8 the example is computed to numerical accuracy for all methods. Figure 4b depicts the error in the energy norm in per cent. The absolute difference of the analytic strain energy to its numeric approximation is in the order of 10^{-12}. Figure 4c depicts the condition number of the stiffness matrix. It is of no surprise that the p-version has a clear advantage over B-Splines in terms of conditioning for this boundary conforming example. The minimum eigenvalues of the overall stiffness matrix including its weak boundary conditions are depicted in fig. 4d. They decrease to small values but always stay positive.

We now look at the performance of the boundary conditions for an embedded problem which we solve using the finite cell method. We use the same mesh consisting of a single patch with 8×8 elements for the B-Spline version or 8×8 p-elements. The discretization is plotted along with the integration mesh and the solution in fig. 5a. Note that there are now all three types of cells Ω_{C1},

Journal publication: https://doi.org/10.1002/nme.4817



(a) solution and discretization

(b) convergence behaviour

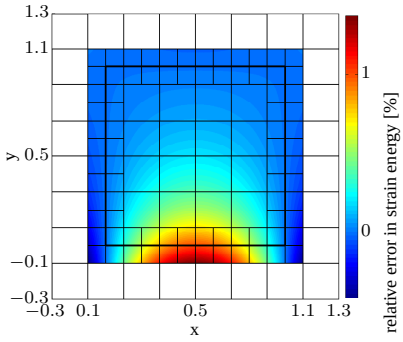
(c) condition number of left hand side of eq. (35)

(d) smallest eigenvalue

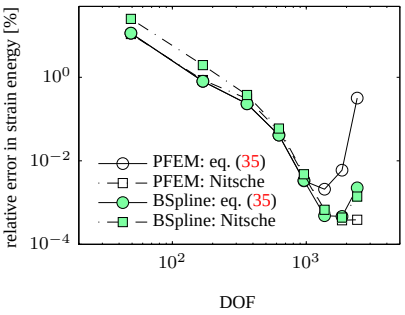
Figure 4. 2D $\Delta a = 0$ (conforming boundary conditions).

Ω_{C2}, and Ω_{C3} and that the computations are carried out as described in section 2: the cells in Ω_{C1} are located completely outside the physical domain and are neither computed nor assembled. The cells in Ω_{C2} are computed as regular finite elements of high order. The cells in Ω_{C3} are adaptively integrated. As we chose a very favorable situation — namely that the boundary Γ cuts the cells in Ω_{C3} exactly in half — an adaptive integration on a quadtree is chosen, leading to an exact integration already at the first level of refinement. Figure 5a explicitly depicts the entire solution, including the integration tree and the unphysical solution in the fictitious domain $\Omega_{\Gamma_{out}}$ outside the boundary depicted by a thick black line. The physical solution is similar to the non-embedded case, only the color-scale is shifted due to the smooth extension. The convergence behavior as percent error in the energy norm is depicted in fig. 5b, whereby the reference solution is the strain energy computed analytically from the solution given in eq. (37). The convergence is of higher order until p=6, where it levels off and, in case of the method given by eq. (35) version, increases. The increase is non-physical and clearly shows the limits of the numerics to compute this situation. At this level, the condition number depicted in fig. 5c rises above 10^{15} which imposes difficulties for the solver used. This deserves a closer look. Apart from the inversion of the stiffness matrix $\boldsymbol{K}$, there are more places where matrices have to be inverted. For the Nitsche's version of imposing boundary conditions, one must solve the eigenvalue problem given in eq. (13) as accurately as possible. The

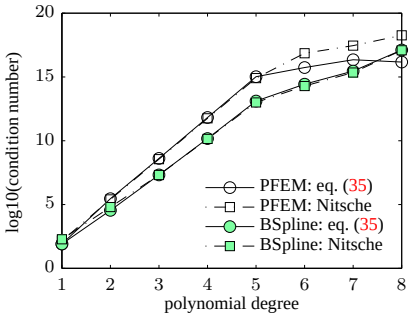
Journal publication: https://doi.org/10.1002/nme.4817



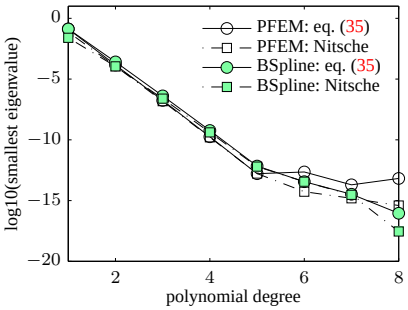
(a) solution, discretization and integration cells



(b) convergence behaviour



(c) condition number of left hand side of eq. (35)



(d) smallest eigenvalue

Figure 5. *non*-conforming boundary

eigenvalue problem is defined on the cut cells and therefore suffers from the conditioning problems as well.

The crucial point for the method given in eq. (35) is the inversion of $\boldsymbol{K}_{\sigma\sigma}$, also defined on the cut cells. It constitutes the core part of the lower bound for the stability term. In the current Matlab implementation of the method, the inversion of $\boldsymbol{K}_{\sigma\sigma}$ is most stable using the pseudo inverse instead of the inverse but it is still less accurate than the solution of the corresponding eigenvalue problem necessary for Nitsche's method. It is worthwhile noting that $\boldsymbol{K}_{\sigma\sigma}$ is similar to a mass matrix and, therefore, neither basis is optimized to render optimal conditioning here. At p=7, the condition of $\boldsymbol{K}_{\sigma\sigma}$ reaches 10^{18} and a correct solution is no longer guaranteed. However, all eigenvalues of the overall stiffness matrix of the discrete system are very small but always remain positive as depicted in fig. 5d.

Journal publication: https://doi.org/10.1002/nme.4817

3.4. Curved boundary

We compute the example depicted in fig. 6. It consists of a Poisson problem solved on a quarter of an annulus, embedded in a Cartesian mesh. Two Dirichlet boundary conditions are set weakly at the inner and outer ring to 1 and 3, respectively. The analytical solution in polar coordinates is given by

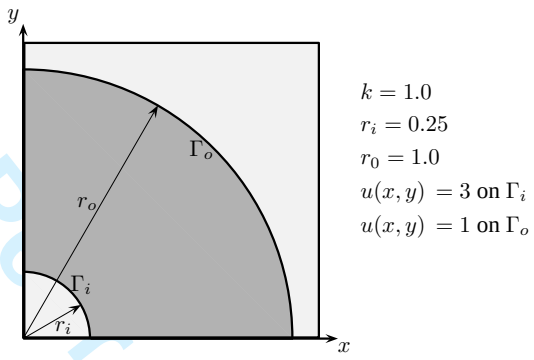


Figure 6. Quarter of an annulus.

$$u(r) = 1 - \frac{ln(r)}{ln(2)} \tag{39}$$

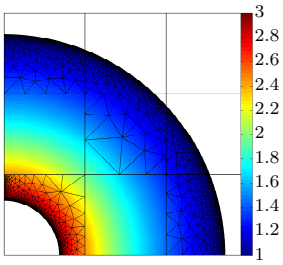
where r denotes the radial distance. It possesses a strain energy of

$$U_{ex} = \frac{\pi}{ln(4)} \approx 2.266180071.$$

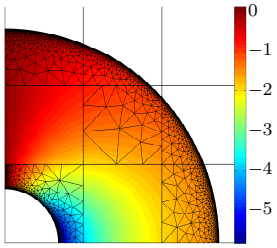
The results are plotted in fig. 7. In addition to the Cartesian mesh spanning the shape functions, we plot the integration mesh in figs. 7a to 7c. Note that, instead of the adaptive sub-cells we use sub-triangles here to integrate the cut-cells. This increases the accuracy of the integration and the solution, see e.g.[28]. The results are snapshots for $p = 3$ whereby we do not plot the smooth extension for the sake of clarity.

The error in the energy norm, the condition number and the smallest eigenvalues are depicted in figs. 7d to 7f, respectively. The new method practically delivers results of comparible accuracy and conditioning for non-conforming, curved boundaries.

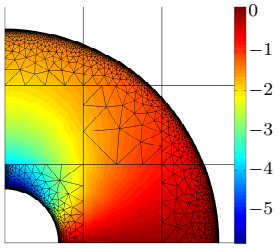
Journal publication: https://doi.org/10.1002/nme.4817



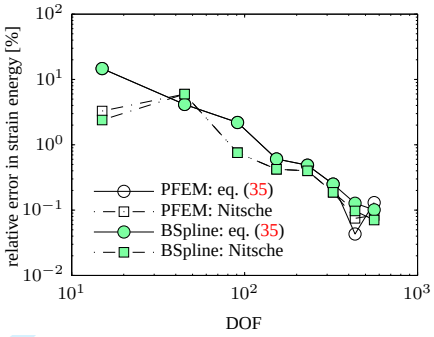
(a) Numerical solution for $p = 3$, discretization, and integration domains



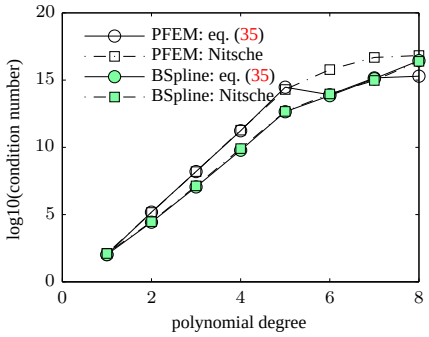
(b) Numerical flux for $p = 3$ in x-direction



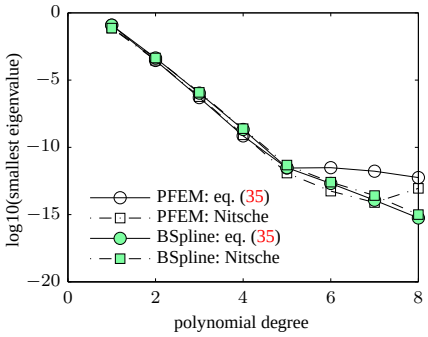
(c) Numerical flux for $p = 3$ in y-direction



(d) convergence behaviour



(e) Condition number of left hand side of eq. (35)



(f) Smallest eigenvalue

Figure 7. Example: Quarter ring, results.

4. PARAMETER-FREE, WEAK COUPLING OF TWO DOMAINS

4.1. General setting

In the following, we consider two domains Ω^I and Ω^{II}, which are sewed together along the common boundary $\Gamma = \Gamma^I = \Gamma^{II}$. On this common boundary, we will demand the continuity of

Journal publication: https://doi.org/10.1002/nme.4817

the displacements as well as the continuity of the fluxes in a weak from. The continuous view of this general setting is depicted in fig. 8. The coupling conditions of Ω^I and Ω^{II} at the common boundary

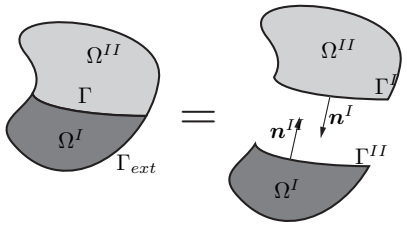


Figure 8. Two physical domains Ω^I and Ω^{II} weakly coupled at Γ.

Γ in strong form are:

$$u^I - u^{II} = 0 \quad \text{on } \Gamma \tag{40}$$

$$\boldsymbol{\sigma}^I \cdot \boldsymbol{n}^I - \boldsymbol{\sigma}^{II} \cdot \boldsymbol{n}^{II} = 0 \quad \text{on } \Gamma \tag{41}$$

4.2. Weak coupling using Nitsche's method

Nitsche's method applied to the coupled Poisson's system is given by:

$$k^I \left(\nabla u^I, \nabla v^I\right)_{\Omega^I} - \frac{1}{2}\left(\boldsymbol{n}^I \cdot k^I \nabla u^I - \boldsymbol{n}^{II} \cdot k^{II} \nabla u^{II}, v^I\right)_\Gamma - \frac{1}{2}\left(u^I, \boldsymbol{n}^I \cdot k^I \nabla v^I - \boldsymbol{n}^{II} \cdot k^{II} \nabla v^{II}\right)_\Gamma + \beta\left(u^I - u^{II}, v^I\right)_\Gamma = \langle f, v^I \rangle_{\Omega^I} \tag{42}$$

$$k^{II} \left(\nabla u^{II}, \nabla v^{II}\right)_{\Omega^{II}} - \frac{1}{2}\left(\boldsymbol{n}^{II} \cdot k^{II} \nabla u^{II} - \boldsymbol{n}^I \cdot k^I \nabla u^I, v^{II}\right)_\Gamma - \frac{1}{2}\left(u^{II}, \boldsymbol{n}^{II} \cdot k^{II} \nabla v^{II} - \boldsymbol{n}^I \cdot k^I \nabla v^I\right)_\Gamma + \beta\left(u^{II} - u^I, v^{II}\right)_\Gamma = \langle f, v^{II} \rangle_{\Omega^{II}} \tag{43}$$

where the weak boundary conditions on $\Gamma_{ext} = \left(\partial\Omega^I \cup \partial\Omega^{II}\right) \setminus \Gamma$ are not written down for the sake of simplicity. The use of Nitsche's method for coupling domains weakly has already extensively been studied, for example in [39]. We refer to [6], for an overview of its properties and to the introduction for further references. Again, an estimator for β is needed and we use the one provided by eq. (13), wherein now

$$\boldsymbol{A} = \begin{bmatrix} \boldsymbol{A}^I & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{A}^{II} \end{bmatrix} \tag{44}$$

and likewise for $\boldsymbol{B}$.

DOI: 10.1002/nme
http://mc.manuscriptcentral.com/nme

Journal publication: https://doi.org/10.1002/nme.4817

variational eqn.	$k\left(\nabla u_h, \nabla v_h\right)_{\Omega,\Omega_\Gamma in}$	$-\frac{1}{nk}\left(\boldsymbol{\tau}_h, \boldsymbol{\sigma}\right)_{\Omega_{\Gamma_{in}}}$	$\frac{1}{n}\left(\boldsymbol{\tau_h}, \nabla u\right)_{\Omega^I_{\Gamma_{in}}}$	$\frac{1}{n}\left(\nabla v_h, \boldsymbol{\sigma}_h\right)_{\Omega_{\Gamma_{in}}}$
algebraic form	$\boldsymbol{K}_{uu}\boldsymbol{U}$	$\boldsymbol{K}_{\sigma\sigma}\boldsymbol{\Sigma}$	$\boldsymbol{K}_{\sigma u}\boldsymbol{U}$	$\boldsymbol{K}_{u\sigma}\boldsymbol{\Sigma}$
variational eqn.	$-\langle\boldsymbol{\sigma}_h\cdot\boldsymbol{n}, v_h\rangle_\Gamma$	$-\langle\boldsymbol{\tau}_h\cdot\boldsymbol{n}, u\rangle_\Gamma$		$\langle f, v_h\rangle_\Omega$
algebraic form	$\boldsymbol{G}_{u\sigma}\boldsymbol{\Sigma}$	$\boldsymbol{G}_{\sigma u}\boldsymbol{U}$		$\boldsymbol{f}$

Table III. Terms concerned with the internal part of each domain of the discrete variational and algebraic form of eqs. (45) to (48).

4.3. *New formulation for parameter-free, symmetry-preserving weak coupling*

We first reconsider the weak form given in eqs. (20) and (21) and note that the parts of the equations on Ω and $\Omega_{\Gamma_{in}}$ only concern the interior Ω. They may thus be written independently of each other for each subdomain Ω^I and Ω^{II}. On the contrary, the coupling conditions at Γ i.e., eqs. (40) and (41), are concerned with both domains. For reasons of consistency, we equally distribute their contribution between both subdomains. Let us define the subspace for the unknowns in each subdomain as $V_h^I \subset H^1(\Omega_h^I)$ and $V_h^{II} \subset H^1(\Omega_h^{II})$. Similarly, we define $S_h^I \subset L^2(\Omega_h^I)$ and $S_h^{II} \subset L^2(\Omega_h^{II})$. Utilizing eqs. (40) and (41), the variational form of the problem consists of finding $u_h^I \in V_h^I$, $\boldsymbol{\sigma}_h^I \in S_h^I$, $u_h^{II} \in V_h^{II}$, $\boldsymbol{\sigma}_h^{II} \in S_h^{II}$ such that:

$$\begin{aligned} &k\left(\nabla u_h^I, \nabla v_h^I\right)_{\Omega^I} - \frac{1}{2}\langle\boldsymbol{\sigma_h^{II}}\cdot\boldsymbol{n}^I + \boldsymbol{\sigma_h^I}\cdot\boldsymbol{n}^I, v_h^I\rangle_\Gamma + \frac{1}{n}\left(\nabla v_h^I, \boldsymbol{\sigma_h^I}\right)_{\Omega^I_{\Gamma_{in}}} + \\ &-\frac{1}{n}k\left(\nabla v_h^I, \nabla u_h^I\right)_{\Omega^I_{\Gamma_{in}}} = \langle f, v_h\rangle_{\Omega^I} \qquad \forall v_h^I \in V_h^I \end{aligned} \tag{45}$$

$$-\frac{1}{nk}\left(\boldsymbol{\tau}_h^I, \boldsymbol{\sigma}_h^I\right)_{\Omega^I_{\Gamma_{in}}} + \frac{1}{n}\left(\boldsymbol{\tau}_h^I, \nabla u_h^I\right)_{\Omega^I_{\Gamma_{in}}} - \frac{1}{2}\langle\boldsymbol{\tau}_h^I\cdot\boldsymbol{n}^I, u_h^I - \boldsymbol{u_h^{II}}\rangle_\Gamma = 0 \qquad \forall\boldsymbol{\tau}_h^I \in S_h^I \tag{46}$$

$$\begin{aligned} &k\left(\nabla u_h^{II}, \nabla v_h^{II}\right)_{\Omega^{II}} - \frac{1}{2}\langle\boldsymbol{\sigma_h^I}\cdot\boldsymbol{n}^{II} + \boldsymbol{\sigma}_h^{II}\cdot\boldsymbol{n}^{II}, v_h^{II}\rangle_\Gamma + \frac{1}{n}\left(\nabla v_h^{II}, \boldsymbol{\sigma}_h^{II}\right)_{\Omega^I_{\Gamma_{in}}} + \\ &-\frac{1}{n}k\left(\nabla v_h^{II}, \nabla u_h^{II}\right)_{\Omega^I_{\Gamma_{in}}} = \langle f, v_h^{II}\rangle_{\Omega^{II}} \qquad \forall v_h^{II} \in V_h^{II} \end{aligned} \tag{47}$$

$$-\frac{1}{nk}\left(\boldsymbol{\tau}_h, \boldsymbol{\sigma}_h\right)_{\Omega^{II}_{\Gamma_{in}}} + \frac{1}{n}\left(\boldsymbol{\tau}_h, \nabla u_h\right)_{\Omega^{II}_{\Gamma_{in}}} - \frac{1}{2}\langle\boldsymbol{\tau}_h^{II}\cdot\boldsymbol{n}^{II}, u_h^{II} - \boldsymbol{u_h^I}\rangle_\Gamma = 0 \qquad \forall\boldsymbol{\tau}_h^{II} \in S_h^{II} \tag{48}$$

where the coupling terms from the respective other domain are marked in blue. The first two equations correspond to the first subdomain, while the last two equations correspond to the second subdomain.

Let $\boldsymbol{U}$ and $\boldsymbol{\Sigma}$ be the vectors of the unknown coefficients of u_h and $\boldsymbol{\sigma}_h$, respectively. For the algebraic version of eqs. (45) to (48) we utilize the notation given in table III for the terms only concerned with one domain. The new terms concerned with the sewing of the domains are collected in table IV, where we have used the fact that $\boldsymbol{n}^I = -\boldsymbol{n}^{II}$. Further, we utilize the fact that $\boldsymbol{G}_{\sigma u}^{I,II} = \boldsymbol{G}_{u\sigma}^{II,I^T}$ and $\boldsymbol{G}_{\sigma u}^{II,I} = \boldsymbol{G}_{u\sigma}^{I,II^T}$. The coupled problem written in matrix form then reads:

DOI: 10.1002/nme

Journal publication: https://doi.org/10.1002/nme.4817

Discrete variational eqn.	$-\langle \boldsymbol{\sigma}^{II} \cdot \boldsymbol{n}^I, v_h^I \rangle_\Gamma$	$-\langle \boldsymbol{\sigma}^{I} \cdot \boldsymbol{n}^{II}, v_h^{II} \rangle_\Gamma$
Algebraic form	$\boldsymbol{G}_{u\sigma}^{I,II} \boldsymbol{\Sigma}^{II}$	$\boldsymbol{G}_{u\sigma}^{II,I} \boldsymbol{\Sigma}^{I}$
Discrete variational eqn.	$-\langle \boldsymbol{\tau}_h^I \cdot \boldsymbol{n}^{II}, u^{II} \rangle_\Gamma$	$-\langle \boldsymbol{\tau}_h^{II} \cdot \boldsymbol{n}^{I}, u^{I} \rangle_\Gamma$
Algebraic form	$\boldsymbol{G}_{\sigma u}^{I,II} \boldsymbol{U}^{II}$	$\boldsymbol{G}_{\sigma u}^{II,I} \boldsymbol{U}^{I}$

Table IV. Coupling terms of the discrete variational and algebraic form eqs. (45) to (48).

$$\left[\begin{array}{cc:cc} \left(1-\frac{1}{n}\right)\boldsymbol{K}_{uu}^I & \boldsymbol{K}_{u\sigma}^I+\frac{1}{2}\boldsymbol{G}_{u\sigma}^I & \boldsymbol{0} & \frac{1}{2}\boldsymbol{G}_{u\sigma}^{I,II} \\ \boldsymbol{K}_{\sigma u}^I+\frac{1}{2}\boldsymbol{G}_{\sigma u}^I & \boldsymbol{K}_{\sigma\sigma}^I & \frac{1}{2}\boldsymbol{G}_{\sigma u}^{I,II} & \boldsymbol{0} \\ \hdashline \boldsymbol{0} & \frac{1}{2}\boldsymbol{G}_{u\sigma}^{II,I} & \left(1-\frac{1}{n}\right)\boldsymbol{K}_{uu}^{II} & \boldsymbol{K}_{u\sigma}^{II}+\frac{1}{2}\boldsymbol{G}_{u\sigma}^{II} \\ \frac{1}{2}\boldsymbol{G}_{\sigma u}^{II,I} & \boldsymbol{0} & \boldsymbol{K}_{\sigma u}^{II}+\frac{1}{2}\boldsymbol{G}_{\sigma u}^{II} & \boldsymbol{K}_{\sigma\sigma}^{II} \end{array}\right] \left[\begin{array}{c} \boldsymbol{U}^I \\ \boldsymbol{\Sigma}^I \\ \hdashline \boldsymbol{U}^{II} \\ \boldsymbol{\Sigma}^{II} \end{array}\right] = \left[\begin{array}{c} \boldsymbol{f}^I \\ \boldsymbol{0} \\ \boldsymbol{f}^{II} \\ \boldsymbol{0} \end{array}\right] \tag{49}$$

4.4. Static condensation

We may compute the fluxes in eq. (49) as:

$$\boldsymbol{\Sigma}^I = \left(\boldsymbol{K}_{\sigma\sigma}^I\right)^{-1} \left(-\left(\boldsymbol{K}_{\sigma u}^I+\frac{1}{2}\boldsymbol{G}_{\sigma u}^I\right)\boldsymbol{U}^I - \frac{1}{2}\boldsymbol{G}_{\sigma u}^{I,II}\boldsymbol{U}^{II}\right) \tag{50}$$

$$\boldsymbol{\Sigma}^{II} = \left(\boldsymbol{K}_{\sigma\sigma}^{II}\right)^{-1} \left(-\left(\boldsymbol{K}_{\sigma u}^{II}+\frac{1}{2}\boldsymbol{G}_{\sigma u}^{II}\right)\boldsymbol{U}^{II} - \frac{1}{2}\boldsymbol{G}_{\sigma u}^{II,I}\boldsymbol{U}^{I}\right) \tag{51}$$

and condense $\boldsymbol{\Sigma}^I$ and $\boldsymbol{\Sigma}^{II}$ out of the system eq. (49). The equation for subdomain I reads:

$$\begin{aligned} &\left(\left(1-\frac{1}{n}\right)\boldsymbol{K}_{uu}^I - \left(\frac{1}{2}\boldsymbol{G}_{u\sigma}^I+\boldsymbol{K}_{u\sigma}^I\right)\left(\boldsymbol{K}_{\sigma\sigma}^I\right)^{-1}\left(\boldsymbol{K}_{\sigma u}^I+\frac{1}{2}\boldsymbol{G}_{\sigma u}^I\right) - \frac{1}{4}\boldsymbol{G}_{u\sigma}^{I,II}\left(\boldsymbol{K}_{\sigma\sigma}^{II}\right)^{-1}\boldsymbol{G}_{\sigma u}^{II,I}\right)\boldsymbol{U}^I \\ &\left(-\left(\boldsymbol{K}_{u\sigma}^I+\frac{1}{2}\boldsymbol{G}_{u\sigma}^I\right)\left(\boldsymbol{K}_{\sigma\sigma}^I\right)^{-1}\frac{1}{2}\boldsymbol{G}_{\sigma u}^{I,II} - \frac{1}{2}\boldsymbol{G}_{u\sigma}^{I,II}\left(\boldsymbol{K}_{\sigma\sigma}^{II}\right)^{-1}\left(\boldsymbol{K}_{\sigma u}^{II}+\frac{1}{2}\boldsymbol{G}_{\sigma u}^{II}\right)\right)\boldsymbol{U}^{II} \end{aligned} \tag{52}$$

We then use the definitions given in eqs. (31), (32) and (34) to simplify the terms only concerned with domain I. Equation (52) then reduces to:

$$\begin{aligned} &\left(\boldsymbol{K}_{uu}^I+\frac{1}{2}\left(\boldsymbol{G}_{uu}^I+\left(\boldsymbol{G}_{uu}^I\right)^T\right) - \frac{1}{4}\boldsymbol{G}_{u\sigma}^I\left(\boldsymbol{K}_{\sigma\sigma}^I\right)^{-1}\boldsymbol{G}_{\sigma u}^I - \frac{1}{4}\boldsymbol{G}_{u\sigma}^{I,II}\left(\boldsymbol{K}_{\sigma\sigma}^{II}\right)^{-1}\boldsymbol{G}_{\sigma u}^{II,I}\right)\boldsymbol{U}^I \\ &\left(-\frac{1}{2}\left(\boldsymbol{K}_{u\sigma}^I+\frac{1}{2}\boldsymbol{G}_{u\sigma}^I\right)\left(\boldsymbol{K}_{\sigma\sigma}^I\right)^{-1}\boldsymbol{G}_{\sigma u}^{I,II} - \frac{1}{2}\boldsymbol{G}_{u\sigma}^{I,II}\left(\boldsymbol{K}_{\sigma\sigma}^{II}\right)^{-1}\left(\boldsymbol{K}_{\sigma u}^{II}+\frac{1}{2}\boldsymbol{G}_{\sigma u}^{II}\right)\right)\boldsymbol{U}^{II} \end{aligned} \tag{53}$$

Likewise, for subdomain II we have:

$$\begin{aligned} &\left(-\frac{1}{2}\boldsymbol{G}_{u\sigma}^{II,I}\left(\boldsymbol{K}_{\sigma\sigma}^I\right)^{-1}\left(\boldsymbol{K}_{\sigma u}^I+\frac{1}{2}\boldsymbol{G}_{\sigma u}^I\right) - \frac{1}{2}\left(\boldsymbol{K}_{u\sigma}^{II}+\frac{1}{2}\boldsymbol{G}_{u\sigma}^{II}\right)\left(\boldsymbol{K}_{\sigma\sigma}^{II}\right)^{-1}\boldsymbol{G}_{\sigma u}^{II,I}\right)\boldsymbol{U}^I \\ &\left(\boldsymbol{K}_{uu}^{II} - \frac{1}{4}\boldsymbol{G}_{u\sigma}^{II}\left(\boldsymbol{K}_{\sigma\sigma}^{II}\right)^{-1}\boldsymbol{G}_{\sigma u}^{II} + \frac{1}{2}\left(\boldsymbol{G}_{uu}^{II}+\left(\boldsymbol{G}_{uu}^{II}\right)^T\right) - \frac{1}{4}\boldsymbol{G}_{u\sigma}^{II,I}\left(\boldsymbol{K}_{\sigma\sigma}^I\right)^{-1}\boldsymbol{G}_{\sigma u}^{I,II}\right)\boldsymbol{U}^{II} \end{aligned} \tag{54}$$

Journal publication: https://doi.org/10.1002/nme.4817

variational eqn.	$-k^I\langle\nabla {u_h}^I\cdot\boldsymbol{n}^{II}, v_h^{II}\rangle_\Gamma$	$-k^{II}\langle\nabla {u_h}^{II}\cdot\boldsymbol{n}^{I}, v_h^{I}\rangle_\Gamma$
algebraic form	$\boldsymbol{G}_{uu}^{II,I}\boldsymbol{U}^I$	$\boldsymbol{G}_{uu}^{I,II}\boldsymbol{U}^{II}$
variational eqn.	$-k^I\langle\nabla {v_h}^I\cdot\boldsymbol{n}^{II}, u_h^{II}\rangle_\Gamma$	$-k^{II}\langle\nabla {v_h}^{II}\cdot\boldsymbol{n}^{I}, u_h^{I}\rangle_\Gamma$
algebraic form	$(\boldsymbol{G}_{uu}^{II,I})^T\boldsymbol{U}^{II}$	$(\boldsymbol{G}_{uu}^{I,II})^T\boldsymbol{U}^{I}$

Table V. Coupling terms from Nitsche's method.

Further, we introduce the following identities:

$$-\boldsymbol{K}_{u\sigma}^{II}(\boldsymbol{K}_{\sigma\sigma}^{II})^{-1}\boldsymbol{G}_{\sigma u}^{II,I}\boldsymbol{U}^I=(\boldsymbol{G}_{uu}^{I,II})^T\boldsymbol{U}^I \tag{55}$$

$$-\boldsymbol{G}_{u\sigma}^{II,I}(\boldsymbol{K}_{\sigma\sigma}^{I})^{-1}\boldsymbol{K}_{\sigma u}^{I}\boldsymbol{U}^I=\boldsymbol{G}_{uu}^{II,I}\boldsymbol{U}^I \tag{56}$$

$$-\boldsymbol{K}_{u\sigma}^{I}(\boldsymbol{K}_{\sigma\sigma}^{I})^{-1}\boldsymbol{G}_{\sigma u}^{I,II}\boldsymbol{U}^{II}=(\boldsymbol{G}_{uu}^{II,I})^T\boldsymbol{U}^{II} \tag{57}$$

$$-\boldsymbol{G}_{u\sigma}^{I,II}(\boldsymbol{K}_{\sigma\sigma}^{II})^{-1}\boldsymbol{K}_{\sigma u}^{II}\boldsymbol{U}^{II}=\boldsymbol{G}_{uu}^{I,II}\boldsymbol{U}^{II} \tag{58}$$

which arise from Nitsche's method as defined in table V. Equation (52) then reads (terms in blue now mark the terms which differ from Nitsche's method, that is, the stabilization terms):

$$\begin{aligned}&\left(\boldsymbol{K}_{uu}^{I}+\frac{1}{2}\left(\boldsymbol{G}_{uu}^{I}+(\boldsymbol{G}_{uu}^{I})^T\right)-\frac{1}{4}\boldsymbol{G}_{u\sigma}^{I}\left(\boldsymbol{K}_{\sigma\sigma}^{I}\right)^{-1}\boldsymbol{G}_{\sigma u}^{I}-\frac{1}{4}\boldsymbol{G}_{u\sigma}^{I,II}\left(\boldsymbol{K}_{\sigma\sigma}^{II}\right)^{-1}\boldsymbol{G}_{\sigma u}^{II,I}\right)\boldsymbol{U}^I\\&\left(\frac{1}{2}\left(\boldsymbol{G}_{uu}^{I,II}+(\boldsymbol{G}_{uu}^{II,I})^T\right)-\frac{1}{4}\boldsymbol{G}_{u\sigma}^{I}\left(\boldsymbol{K}_{\sigma\sigma}^{I}\right)^{-1}\boldsymbol{G}_{\sigma u}^{I,II}-\frac{1}{4}\boldsymbol{G}_{u\sigma}^{I,II}\left(\boldsymbol{K}_{\sigma\sigma}^{II}\right)^{-1}\boldsymbol{G}_{\sigma u}^{II}\right)\boldsymbol{U}^{II}\end{aligned} \tag{59a}$$

Likewise, for subdomain II we have:

$$\begin{aligned}&\left(\frac{1}{2}\left(\boldsymbol{G}_{uu}^{II,I}+(\boldsymbol{G}_{uu}^{I,II})^T\right)-\frac{1}{4}\boldsymbol{G}_{u\sigma}^{II,I}\left(\boldsymbol{K}_{\sigma\sigma}^{I}\right)^{-1}\boldsymbol{G}_{\sigma u}^{I}-\frac{1}{4}\boldsymbol{G}_{u\sigma}^{II}\left(\boldsymbol{K}_{\sigma\sigma}^{II}\right)^{-1}\boldsymbol{G}_{\sigma u}^{II,I}\right)\boldsymbol{U}^I\\&\left(\boldsymbol{K}_{uu}^{II}+\frac{1}{2}\left(\boldsymbol{G}_{uu}^{II}+(\boldsymbol{G}_{uu}^{II})^T\right)-\frac{1}{4}\boldsymbol{G}_{u\sigma}^{II}\left(\boldsymbol{K}_{\sigma\sigma}^{II}\right)^{-1}\boldsymbol{G}_{\sigma u}^{II}-\frac{1}{4}\boldsymbol{G}_{u\sigma}^{II,I}\left(\boldsymbol{K}_{\sigma\sigma}^{I}\right)^{-1}\boldsymbol{G}_{\sigma u}^{I,II}\right)\boldsymbol{U}^{II}\end{aligned} \tag{59b}$$

It is interesting to note that the penalty terms have a contribution which arises from subdomain I and a contribution from subdomain II. They ensure the stability of the method independently of the geometry of the meshes in subdomain I and II, as proven in section 4.5.

4.5. *Stability Analysis*

In this subsection we prove that the formulation for weak coupling given by eqs. (59a) and (59b) is stable. To do so, we follow a strategy similar to the one presented in [33]. In order to show the stability of the method for weak coupling, we will consider the unknowns u^I to vanish on the external boundary Γ^{ext} of subdomain I, and the normal component of the stresses $\boldsymbol{\sigma}^{II}$ to be null in the external boundary in subdomain II. Further, to keep the proof as concise as possible, we consider only the case in which the material constants k are the same for both domains. Additionally, we assume the common boundary Γ^{ext} to be a straight line.

Journal publication: https://doi.org/10.1002/nme.4817

$$u^I = 0 \quad \text{on } \Gamma^{ext} \tag{60}$$
$$\boldsymbol{\sigma}^{II} \cdot \boldsymbol{n}^{II} = 0 \quad \text{on } \Gamma^{ext} \tag{61}$$

In this setting, we redefine $V_h^I \subset H_0^1(\Omega_h^I)$, where the zero subscript denotes functions vanishing in the external boundary Γ^{ext}. We define the norm:

$$|||[u^I, \boldsymbol{\sigma}^I, u^{II}, \boldsymbol{\sigma}^{II}]|||^2 =$$
$$k||\nabla u^I||^2_{L^2(\Omega^I)} + k||\nabla u^{II}||^2_{L^2(\Omega^{II})} + \frac{k}{h}||u^I - u^{II}||^2_{L^2(\Gamma)} + \frac{1}{k}||\boldsymbol{\sigma}^I||^2_{L^2(\Omega^I_{\Gamma,in})} + \frac{1}{k}||\boldsymbol{\sigma}^{II}||^2_{L^2(\Omega^{II}_{\Gamma,in})} \tag{62}$$

where h is the cell size. The bilinear form for the global problem defined on both subdomains is:

$$B\left(\left[u_h^I, \boldsymbol{\sigma}_h^I, u_h^{II}, \boldsymbol{\sigma}_h^{II}\right], \left[v_h^I, \boldsymbol{\tau}_h^I, v_h^{II}, \boldsymbol{\tau}_h^{II}\right]\right) =$$
$$\begin{aligned}
&k\left(\nabla u_h^I, \nabla v_h^I\right)_{\Omega^I} - \frac{1}{2}\langle \boldsymbol{\sigma}_h^{II} \cdot \boldsymbol{n}^I + \boldsymbol{\sigma}_h^I \cdot \boldsymbol{n}^I, v_h^I\rangle_\Gamma + \frac{1}{n}\left(\nabla v_h^I, \boldsymbol{\sigma}_h^I\right)_{\Omega^I_{\Gamma_{in}}} - \frac{1}{n}k\left(\nabla v_h^I, \nabla u_h^I\right)_{\Omega^I_{\Gamma_{in}}} + \\
&\quad - \frac{1}{nk}\left(\boldsymbol{\tau}_h^I, \boldsymbol{\sigma}_h^I\right)_{\Omega^I_{\Gamma_{in}}} + \frac{1}{n}\left(\boldsymbol{\tau}_h^I, \nabla u_h^I\right)_{\Omega^I_{\Gamma_{in}}} - \frac{1}{2}\langle \boldsymbol{\tau}_h^I \cdot \boldsymbol{n}^I, u_h^I - u_h^{II}\rangle_\Gamma + \\
&k\left(\nabla u_h^{II}, \nabla v_h^{II}\right)_{\Omega^{II}} - \frac{1}{2}\langle \boldsymbol{\sigma}_h^I \cdot \boldsymbol{n}^{II} + \boldsymbol{\sigma}_h^{II} \cdot \boldsymbol{n}^{II}, v_h^{II}\rangle_\Gamma + \frac{1}{n}\left(\nabla v_h^{II}, \boldsymbol{\sigma}_h^{II}\right)_{\Omega^{II}_{\Gamma_{in}}} - \frac{1}{n}k\left(\nabla v_h^{II}, \nabla u_h^{II}\right)_{\Omega^{II}_{\Gamma_{in}}} + \\
&\quad - \frac{1}{nk}\left(\boldsymbol{\tau}_h^{II}, \boldsymbol{\sigma}_h^{II}\right)_{\Omega^{II}_{\Gamma_{in}}} + \frac{1}{n}\left(\boldsymbol{\tau}_h^{II}, \nabla u_h^{II}\right)_{\Omega^{II}_{\Gamma_{in}}} - \frac{1}{2}\langle \boldsymbol{\tau}_h^{II} \cdot \boldsymbol{n}^{II}, u_h^{II} - u_h^I\rangle_\Gamma
\end{aligned} \tag{63}$$

We suppose that V_h and S_h are such that

$$\forall v_h \in V_h \quad \exists \boldsymbol{\tau}_h \in S_h \;\mid\; \delta_1 \|v_h\|^2_{L^2(\Gamma)} \leq \langle \boldsymbol{\tau}_h \cdot \boldsymbol{n}, v_h\rangle_\Gamma + \delta_0 h \|\nabla v_h\|^2 \tag{64}$$
$$\|\boldsymbol{\tau}_h\|_{L^2(\Gamma)} = \|v_h\|_{L^2(\Gamma)}, \quad \|\boldsymbol{\tau}_h\|^2 \leq \delta_2 h \|v_h\|^2_{L^2(\Gamma)} \tag{65}$$

where $\delta_0, \delta_1, \delta_2$ are positive non-dimensional constants which depend on the geometry of the mesh. Conditions in eqs. (64) and (65) are an assumption of the formulation, but these conditions hold for the most common interpolation spaces (see [33]).

We will show the stability of eq. (63) by obtaining an inf-sup condition in the norm of eq. (62). Let us take $\left[v_h^I, \boldsymbol{\tau}_h^I, v_h^{II}, \boldsymbol{\tau}_h^{II}\right] = \left[u_h^I, -\boldsymbol{\sigma}_h^I - \frac{\beta}{h}k\widetilde{\boldsymbol{\tau}}_h^I, u_h^{II}, -\boldsymbol{\sigma}_h^{II} - \frac{\beta}{h}k\widetilde{\boldsymbol{\tau}}_h^{II}\right]$, where $\widetilde{\boldsymbol{\tau}}_h^I$ is the function in S_h^I which makes eqs. (64) and (65) hold for $u_h^{II} - u_h^I$, and $\widetilde{\boldsymbol{\tau}}_h^{II}$ is the function in S_h^{II} which makes eqs. (64) and (65) hold for $u_h^{II} - u_h^I$. β is a dimensionless constant to be defined. We have:

$$\begin{aligned}
&B\left(\left[u_h^I, \boldsymbol{\sigma}_h^I, u_h^{II}, u_h^{II}, \boldsymbol{\sigma}_h^{II}\right], \left[u_h^I, -\boldsymbol{\sigma}_h^I - \frac{\beta}{h}k\widetilde{\boldsymbol{\tau}}_h^I, u_h^{II}, -\boldsymbol{\sigma}_h^{II} - \frac{\beta}{h}k\widetilde{\boldsymbol{\tau}}_h^{II}\right]\right) = \\
&k\left(\nabla u_h^I, \nabla u_h^I\right)_{\Omega^I} - \frac{1}{2}\langle \boldsymbol{\sigma}_h^{II} \cdot \boldsymbol{n}^I + \boldsymbol{\sigma}_h^I \cdot \boldsymbol{n}^I, u_h^I\rangle_\Gamma + \frac{1}{n}\left(\nabla u_h^I, \boldsymbol{\sigma}_h^I\right)_{\Omega^I_{\Gamma_{in}}} - \frac{1}{n}k\left(\nabla u_h^I, \nabla u_h^I\right)_{\Omega^I_{\Gamma_{in}}} + \\
&+ \frac{1}{nk}\left(\boldsymbol{\sigma}_h^I, \boldsymbol{\sigma}_h^I\right)_{\Omega^I_{\Gamma_{in}}} - \frac{1}{n}\left(\boldsymbol{\sigma}_h^I, \nabla u_h^I\right)_{\Omega^I_{\Gamma_{in}}} + \frac{1}{2}\langle \boldsymbol{\sigma}_h^I \cdot \boldsymbol{n}^I, u_h^I - u_h^{II}\rangle_\Gamma + \\
&+ \frac{\beta}{nh}\left(\widetilde{\boldsymbol{\tau}}_h^I, \boldsymbol{\sigma}_h^I\right)_{\Omega^I_{\Gamma_{in}}} - \frac{\beta k}{nh}\left(\widetilde{\boldsymbol{\tau}}_h^I, \nabla u_h^I\right)_{\Omega^I_{\Gamma_{in}}} + \frac{\beta k}{2h}\langle \widetilde{\boldsymbol{\tau}}_h^I \cdot \boldsymbol{n}^I, u_h^I - u_h^{II}\rangle_\Gamma +
\end{aligned}$$

$$
\begin{aligned}
&k\left(\nabla u_h^{II},\nabla u_h^{II}\right)_{\Omega^{II}} - \frac{1}{2}\langle \boldsymbol{\sigma}_h^I\cdot\boldsymbol{n}^{II}+\boldsymbol{\sigma}_h^{II}\cdot\boldsymbol{n}^{II},u_h^{II}\rangle_\Gamma + \frac{1}{n}\left(\nabla u_h^{II},\boldsymbol{\sigma}_h^{II}\right)_{\Omega^{II}_{\Gamma_{in}}} - \frac{1}{n}k\left(\nabla u_h^{II},\nabla u_h^{II}\right)_{\Omega^{II}_{\Gamma_{in}}} + \\
&+\frac{1}{nk}\left(\boldsymbol{\sigma}_h^{II},\boldsymbol{\sigma}_h^{II}\right)_{\Omega^{II}_{\Gamma_{in}}} - \frac{1}{n}\left(\boldsymbol{\sigma}_h^{II},\nabla u_h^{II}\right)_{\Omega^{II}_{\Gamma_{in}}} + \frac{1}{2}\langle\boldsymbol{\sigma}_h^{II}\cdot\boldsymbol{n}^{II},u_h^{II}-u_h^I\rangle_\Gamma + \\
&+\frac{\beta}{nh}\left(\widetilde{\boldsymbol{\tau}}_h^{II},\boldsymbol{\sigma}_h^{II}\right)_{\Omega^{II}_{\Gamma_{in}}} - \frac{\beta k}{nh}\left(\widetilde{\boldsymbol{\tau}}_h^{II},\nabla u_h^{II}\right)_{\Omega^{II}_{\Gamma_{in}}} + \frac{\beta k}{2h}\langle\widetilde{\boldsymbol{\tau}}_h^{II}\cdot\boldsymbol{n}^{II},u_h^{II}-u_h^I\rangle_\Gamma \geq \\
&\left(1-\frac{1}{n}\right)k||\nabla u_h^I||^2+\frac{1}{nk}||\boldsymbol{\sigma}_h^I||^2 + \\
&-\frac{\beta\delta_2^{1/2}}{nh^{1/2}}\|u_h^{II}-u_h^I\|_{L^2(\Gamma)}||\boldsymbol{\sigma}_h^I|| - \frac{\beta k\delta_2^{1/2}}{nh^{1/2}}\|u_h^{II}-u_h^I\|_{L^2(\Gamma)}||\nabla u_h^I|| + \frac{\beta k\delta_1}{2h}\|u_h^{II}-u_h^I\|^2_{L^2(\Gamma)} - \frac{\beta k\delta_0}{2}\|\nabla(u_h^{II}-u_h^I)\|^2 + \\
&\left(1-\frac{1}{n}\right)k||\nabla u_h^{II}||^2+\frac{1}{nk}||\boldsymbol{\sigma}_h^{II}||^2 + \\
&-\frac{\beta\delta_2^{1/2}}{nh^{1/2}}\|u_h^{II}-u_h^I\|_{L^2(\Gamma)}||\boldsymbol{\sigma}_h^{II}|| - \frac{\beta k\delta_2^{1/2}}{nh^{1/2}}\|u_h^{II}-u_h^I\|_{L^2(\Gamma)}||\nabla u_h^{II}|| + \frac{\beta k\delta_1}{2h}\|u_h^{II}-u_h^I\|^2_{L^2(\Gamma)} - \frac{\beta k\delta_0}{2}\|\nabla(u_h^{II}-u_h^I)\|^2 \geq \\
&\left(1-\frac{1}{n}\right)k||\nabla u_h^I||^2+\frac{1}{nk}||\boldsymbol{\sigma}_h^I||^2+\left(1-\frac{1}{n}\right)k||\nabla u_h^{II}||^2+\frac{1}{nk}||\boldsymbol{\sigma}_h^{II}||^2+\frac{\beta k\delta_1}{h}\|u_h^{II}-u_h^I\|^2_{L^2(\Gamma)} + \\
&-\frac{\delta_2^{1/2}}{2\gamma h^{1/2}}\frac{\beta\delta_2^{1/2}}{nh^{1/2}}\|u_h^{II}-u_h^I\|_{L^2(\Gamma)} - \frac{\gamma h^{1/2}}{2\delta_2^{1/2}}\frac{\beta\delta_2^{1/2}}{nh^{1/2}}||\boldsymbol{\sigma}_h^I|| - \frac{\delta_2^{1/2}}{2\gamma h^{1/2}}\frac{\beta\delta_2^{1/2}}{nh^{1/2}}\|u_h^{II}-u_h^I\|_{L^2(\Gamma)} - \frac{\gamma h^{1/2}}{2\delta_2^{1/2}}\frac{\beta\delta_2^{1/2}}{nh^{1/2}}||\nabla u_h^I|| + \\
&-\frac{\delta_2^{1/2}}{2\gamma h^{1/2}}\frac{\beta\delta_2^{1/2}}{nh^{1/2}}\|u_h^{II}-u_h^I\|_{L^2(\Gamma)} - \frac{\gamma h^{1/2}}{2\delta_2^{1/2}}\frac{\beta\delta_2^{1/2}}{nh^{1/2}}||\boldsymbol{\sigma}_h^{II}|| - \frac{\delta_2^{1/2}}{2\gamma h^{1/2}}\frac{\beta\delta_2^{1/2}}{nh^{1/2}}\|u_h^{II}-u_h^I\|_{L^2(\Gamma)} - \frac{\gamma h^{1/2}}{2\delta_2^{1/2}}\frac{\beta\delta_2^{1/2}}{nh^{1/2}}||\nabla u_h^{II}|| + \\
&-\beta k\delta_0\|\nabla u_h^{II}||^2-\beta k\delta_0\|\nabla u_h^I\|^2 \geq \\
&\left(1-\frac{1}{n}-\beta\left(\frac{\gamma}{2n}+\delta_0\right)\right)k||\nabla u_h^I||^2+\frac{1}{n}\left(1-\frac{\beta\gamma}{2}\right)\frac{1}{k}||\boldsymbol{\sigma}_h^I||^2 + \\
&\left(1-\frac{1}{n}-\beta\left(\frac{\gamma}{2n}+\delta_0\right)\right)k||\nabla u_h^{II}||^2+\frac{1}{n}\left(1-\frac{\beta\gamma}{2}\right)\frac{1}{k}||\boldsymbol{\sigma}_h^{II}||^2 + \\
&\beta\left(\delta_1-\frac{2\delta_2}{\gamma n}\right)\frac{k}{h}\|u_h^{II}-u_h^I\|_{L^2(\Gamma)}
\end{aligned}
$$

where γ is an arbitrary dimensionless constant. We now take

$$
n>1,\ \gamma>\frac{2\delta_2}{\delta_1 n},\ \beta<\min\left(\frac{1-\frac{1}{n}}{\left(\frac{\gamma}{2n}+\delta_0\right)},\frac{2}{\gamma}\right), \tag{66}
$$

and we take into account that:

$$
\begin{aligned}
&|||\left[u_h^I,-\boldsymbol{\sigma}_h^I-\frac{\beta}{h}k\widetilde{\boldsymbol{\tau}}_h^I,u_h^{II},-\boldsymbol{\sigma}_h^{II}-\frac{\beta}{h}k\widetilde{\boldsymbol{\tau}}_h^{II}\right]|||^2 = \\
&k||\nabla u_h^I||^2_{L^2(\Omega^I)}+k||\nabla u_h^{II}||^2_{L^2(\Omega^{II})}+\frac{k}{h}||u_h^I-u_h^{II}||^2_{L^2(\Gamma)}+\frac{1}{k}||\boldsymbol{\sigma}_h^I-\frac{\beta}{h}k\widetilde{\boldsymbol{\tau}}_h^I||^2+\frac{1}{k}||\boldsymbol{\sigma}_h^{II}-\frac{\beta}{h}k\widetilde{\boldsymbol{\tau}}_h^{II}||^2 \leq \\
&k||\nabla u_h^I||^2_{L^2(\Omega^I)}+k||\nabla u_h^{II}||^2_{L^2(\Omega^{II})}+\frac{k}{h}||u_h^I-u_h^{II}||^2_{L^2(\Gamma)}+\frac{2}{k}||\boldsymbol{\sigma}_h^I||^2+\frac{2k\beta^2}{h^2}||\widetilde{\boldsymbol{\tau}}_h^I||^2+\frac{2}{k}||\boldsymbol{\sigma}_h^{II}||^2+\frac{2k\beta^2}{h^2}||\widetilde{\boldsymbol{\tau}}_h^{II}||^2 \leq \\
&k||\nabla u_h^I||^2_{L^2(\Omega^I)}+k||\nabla u_h^{II}||^2_{L^2(\Omega^{II})}+\frac{k}{h}(1+4\beta^2\delta_2)||u_h^I-u_h^{II}||^2_{L^2(\Gamma)}+\frac{2}{k}||\boldsymbol{\sigma}_h^I||^2+\frac{2}{k}||\boldsymbol{\sigma}_h^{II}||^2 \leq \\
&\max(1+4\beta^2\delta_2,2)|||[u_h^I,\boldsymbol{\sigma}_h^I,u_h^{II},\boldsymbol{\sigma}_h^{II}]|||^2
\end{aligned}
$$

which gives us the following theorem:

Theorem. Suppose that eqs. (64) and (65) hold and $n>1$. Then, the bilinear form eq. (63) satisfies that for all $[u_h^I,\boldsymbol{\sigma}_h^I,u_h^{II},\boldsymbol{\sigma}_h^{II}]$ there exist $\left[v_h^I,\boldsymbol{\tau}_h^I,v_h^{II},\boldsymbol{\tau}_h^{II}\right]$ and $\alpha>0$ such that

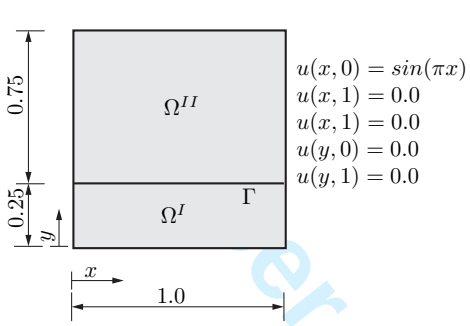
$$
B\left(\left[u_h^I,\boldsymbol{\sigma}_h^I,u_h^{II},\boldsymbol{\sigma}_h^{II}\right],\left[v_h^I,\boldsymbol{\tau}_h^I,v_h^{II},\boldsymbol{\tau}_h^{II}\right]\right)\geq\alpha|||\left[u_h^I,\boldsymbol{\sigma}_h^I,u_h^{II},\boldsymbol{\sigma}_h^{II}\right]|||\,|||\left[v_h^I,\boldsymbol{\tau}_h^I,v_h^{II},\boldsymbol{\tau}_h^{II}\right]|||
$$

4.6. *Examples*

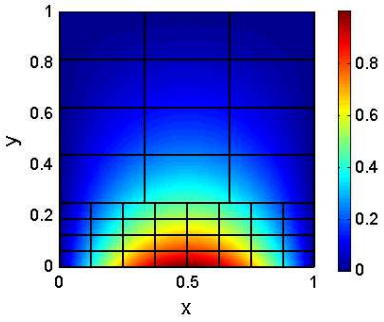
In this section we present various examples, where the performance of the new method is compared to Nitsche's method in the framework of both p- and B-spline version of the FCM.

Journal publication: https://doi.org/10.1002/nme.4817

4.6.1. Constant coefficients, non-conform discretizations in h *and* p. This introductory example is to provide a first insight into the performance of the weak coupling method given by eqs. (59a) and (59b). The starting point is the 2D Poisson's problem with a smooth solution already presented in section 3.3. This time, however, the problem is split into two domains Ω^I and Ω^{II} which are weakly coupled. The continuous situation is depicted in fig. 9a. We choose to discretize Ω^I with



(a) Continuous view



(b) Numerical solution with the BSpline version and a *non*-conforming discretization in h and p (Ω^I with $p = 3$ and Ω^{II} with $p = 2$) computed with eqs. (59a) and (59b)

Figure 9. 2D Poisson's problem as in fig. 3 but consisting of two domains Ω^I and Ω^{II}, weakly coupled at Γ.

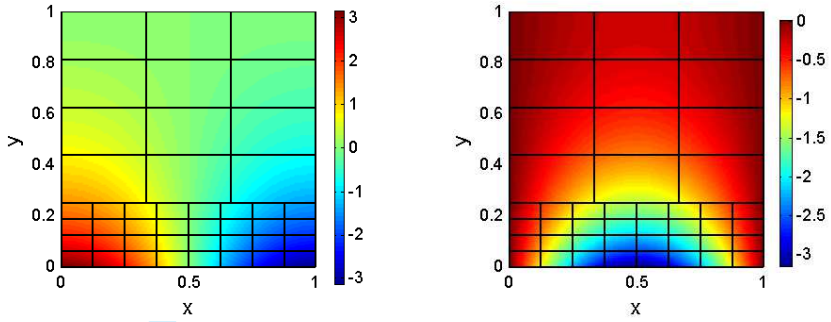
8×4 elements while Ω^{II} is discretized by 3×4 elements. The discretization is depicted in fig. 9b among with a snapshot of the solution. The *non*-conformity of the discretization on Γ manifests itself not only in the appearance of hanging nodes. Additionally, we discretize Ω^I with p and Ω^{II} with $p - 1$. The discretizations are, thus, *non*-conforming in h and p across the interface. However, in a first step, the domains do not overlap, i.e., they are coupled along Γ. The coupled solution depicted in fig. 9b does not show any (visible) kinks or jumps across Γ. This demonstrates that eq. (40) is satisfied.

The corresponding fluxes depicted for both directions x and y are depicted in fig. 10. They are smooth across Γ which demonstrates that eq. (41) is also satisfied.

Figure 11a depicts the convergence in the energy norm as we increase p from 2 to 9 for Ω^I and from 1 to 8 for Ω^{II}. We consider the p- and B-spline version of FCM, using both standard Nitsche and the method proposed here for patch coupling. The rate of convergence is exponential until numerical accuracy is reached and the curves for all methods are practically congruent to one another. The condition numbers are depicted in fig. 11b. They remain quite low for the p-version and are once again higher for the B-Spline version due to the sub-optimality of the BSpline basis with respect to the condition numbers. The plot of the smallest eigenvalue of the stiffness matrix fig. 11c again shows that all discrete eigenvalues remain positive.

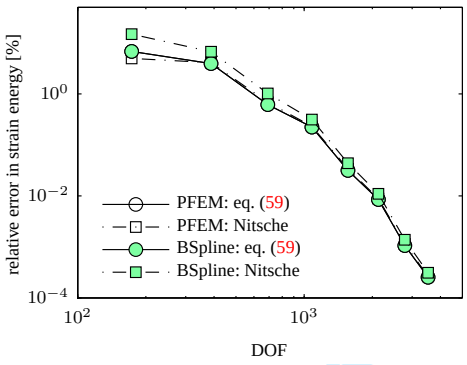
4.6.2. Constant coefficients, overlapping discretizations. We now change the discretization of the example given in section 4.6.1, (fig. 9a) such that the domains Ω^I and Ω^{II} are overlapping in the region $(0 \leq x < 1.0) \wedge (0.45 \leq y < 0.55)$. We weakly enforce the continuity (eqs. (40) and (41)) at the boundary Γ and explicitly note that Γ is now inclined and internal to the discretization of both

Journal publication: https://doi.org/10.1002/nme.4817

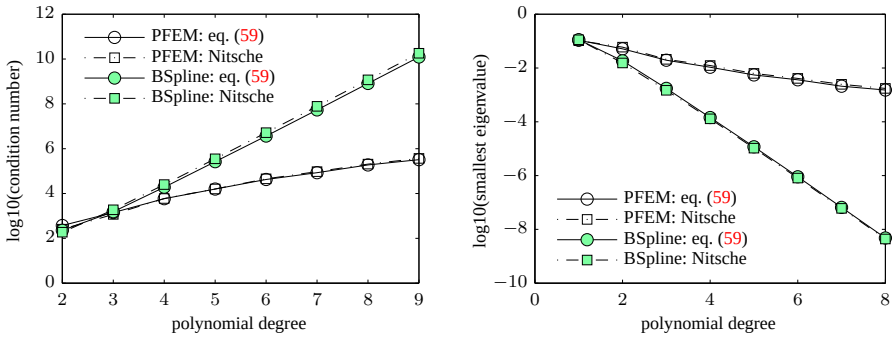


(a) Numerical flux in x-direction (b) Numerical flux in y-direction

Figure 10. Flux corresponding to fig. 9b.



(a) convergence behaviour

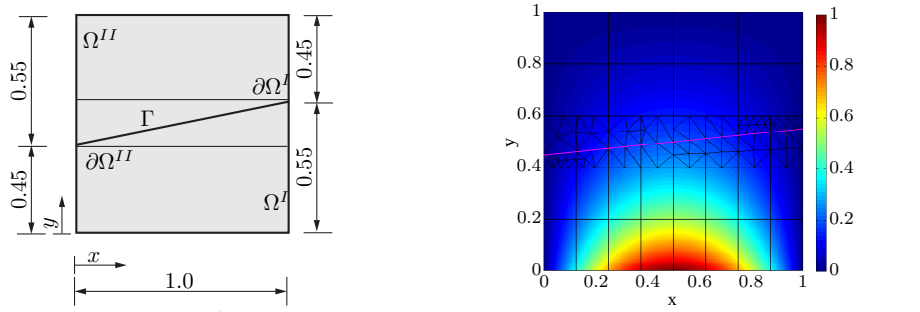


(b) condition number of system stiffness matrix (c) smallest eigenvalue

Figure 11. analysis corresponding to fig. 9b

domains. The continuous situation is depicted in fig. 12a. The solution is depicted in fig. 12b along with its Cartesian discretization and the triangular integration mesh. The integration is carried out on

Journal publication: https://doi.org/10.1002/nme.4817



(a) Continuous view

(b) Numerical solution with the B-Spline version (Ω^I with $p = 3 \wedge \Omega^{II}$ with $p = 2$)

Figure 12. Constant coefficients, overlapping discretizations: Both computational domains Ω^I and Ω^{II} have an extension of $x \times y = 1.0 \times 0.55$. They overlap on $(0 \leq x < 1.0) \wedge (0.45 \leq y < 0.55)$.

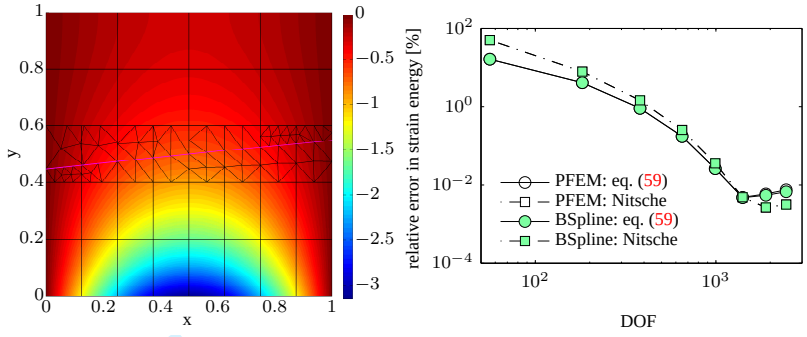
$\Omega_{\Gamma_{in}}$ only. Note that $\Omega_{\Gamma_{in}} \in \Omega^I$ lies below Γ and $\Omega_{\Gamma_{in}} \in \Omega^{II}$ lies above Γ. The obtained numerical solution is smooth, already at low polynomial orders. This is particularly eminent on the plot of the flux in y-direction in fig. 13a. The plot of the fluxes in x-direction is omitted as it provides no further insight. Again, we note an exponential convergence in fig. 13b before it levels off at a low strain energy due to the bad conditioning depicted in fig. 13c. The corresponding lowest eigenvalues are depicted in fig. 13d.

4.6.3. Bimetal strip: discontinuous coefficients, non-conforming discretizations in h *and* p. This example serves to show that the proposed method has no difficulties in representing discontinuities in the derivatives of the solution across material interfaces which arise if the material coefficients jump as well. The first example with jumping coefficients is a bimetal strip on which the Laplace problem is solved. Its solution is depicted in fig. 14a. Dirichlet boundary conditions are applied at the lower end $u(x, 0) = 0$ and on the upper end $u(x, 1) = 1$. We split the domain into a lower part Ω_I in which the material coefficient is $k^I = 1 \, \forall x, y \in \Omega^I$ and an upper part Ω^{II} in which $k^{II} = 2 \, \forall x, y \in \Omega^{II}$. The interface between the two domains $\Gamma : y = 0.5$ is a straight line, parallel to the x-axis. We discretize Ω^I with 5×4 elements with $p = 1$ organized in one patch covering all of Ω^I and likewise Ω^{II} with 3×3 elements with $p = 1$. This results in hanging nodes at Γ.

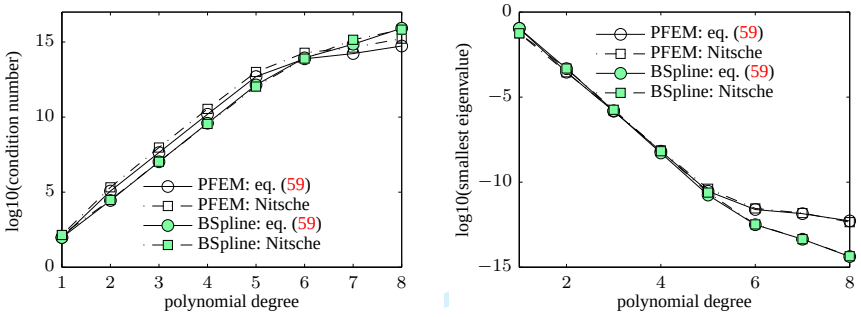
We expect a linear solution in y and a constant solution in x with a kink at the interface Γ. Since the solution is already represented exactly at $p = 1$, convergence-studies are not carried out. Figure 14a depicts the numerical solution along with the discretization. The fluxes are expected to exhibit a jump at Γ and should remain constant within the domains. Figure 14b depicts a view of the fluxes in y-direction along with the chosen discretization. The exact solution is matched up to machine-precision, as expected.

4.6.4. Inclusion problem: discontinuous coefficients and embedded domain. The last example aims at demonstrating the performance of the method given by eqs. (59a) and (59b) in an embedded domain situation. The problem consists of an infinite medium, where the material coefficient is $k^I = 1$, with a circular inclusion, where the material coefficient is $k^{II} = 0.2$. A constant flux is

Journal publication: https://doi.org/10.1002/nme.4817

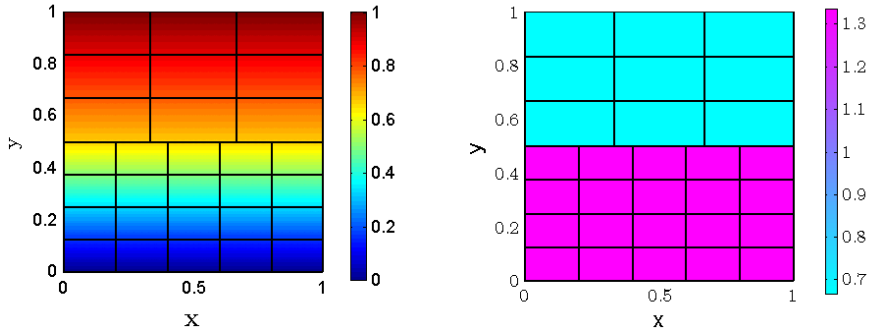


(a) Numerical flux in y-direction (b) convergence behaviour



(c) Condition number of system stiffness matrix (d) Smallest eigenvalue

Figure 13. Analysis corresponding to fig. 12a.



(a) discretization and solution (b) discretization and derivative of $u(x, y)$ in y

Figure 14. Bimetal strip: solution $u(x, y)$ and derivative of $u(x, y)$ in y along with discretization

applied along the x-direction at large distances from the inclusion. The analytical solution is given

Journal publication: https://doi.org/10.1002/nme.4817

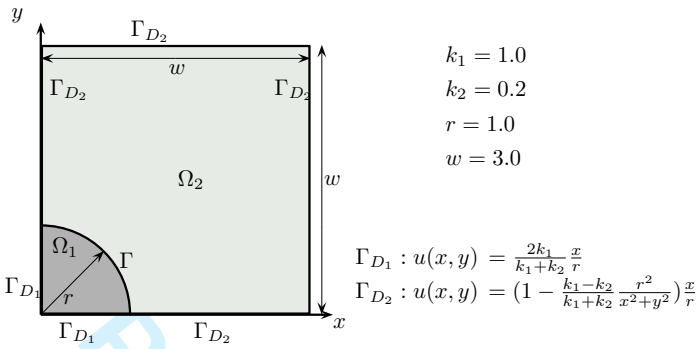


Figure 15. Example: circular inclusion

by

$$u(x,y) = \begin{cases} \frac{2k_1}{k_1+k_2}\frac{x}{r} & \text{if } x^2+y^2 < r^2 \\ (1 - \frac{k_1-k_2}{k_1+k_2}\frac{r^2}{x^2+y^2})\frac{x}{r} & \text{if } x^2+y^2 \geq r^2 \end{cases} \tag{67}$$

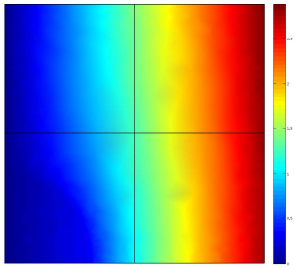
where r denotes the radius of the inclusion. Due to its symmetry, only a quarter of the domain is modeled and the Dirichlet boundary conditions are applied to the boundaries Γ_{D1} and Γ_{D2} according to the analytical solution. The continuous view of the example is depicted in fig. 15. The results are depicted in figs. 16a to 16f. The performance of the new method carries over from the rest of the examples provided in this section. This example shows how well discontinuities in the derivatives are approximated, also in an embedded domain situation (see figs. 16b and 16c). To clear the view, we omit depicting the integration mesh and only show the Cartesian discretization on which the shape functions are spanned.

5. SUMMARY, CONCLUSIONS AND SUGGESTIONS FOR FURTHER RESEARCH

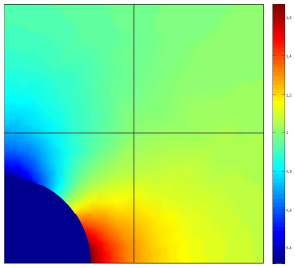
Starting point of this paper was the formulation given in [33] for the application of boundary conditions which must not necessarily conform to the finite element mesh. It is attractive in the sense that there is no need to compute a stabilization parameter. We demonstrated, that this method is able to provide results of similar accuracy as the more classical Nitsche's formulation which calls for the computation of stabilization parameters.

We have then used the fundamental ideas provided in [33] for the derivation of a new formulation for the weak coupling of domains. It is, as well, free from problem-dependent stabilization parameters which need estimation. The new, coupled formulation works well for discretizations of high order, too. It is not specific to the underlying discretization and preserves symmetry. It is thus well suited for the weak coupling of trimmed B-Spline patches and high-order embedded domain

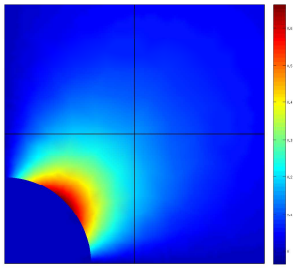
Journal publication: https://doi.org/10.1002/nme.4817



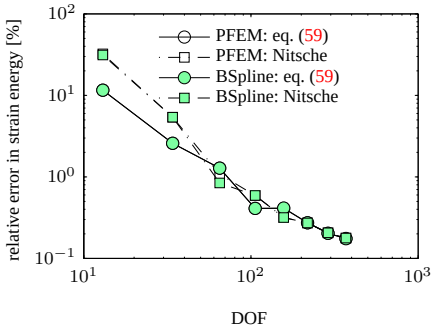
(a) solution, discretization and integration



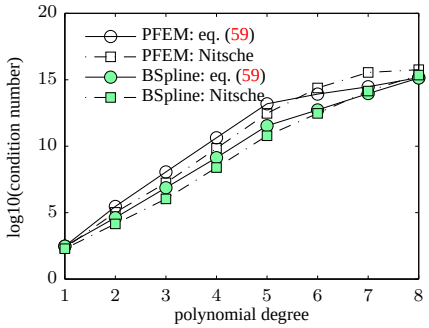
(b) derivative of $u(x, y)$ in x along



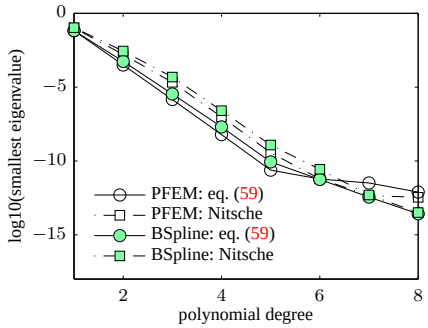
(c) derivative of $u(x, y)$ in y along



(d) convergence behaviour



(e) condition number of left hand side of eqs. (20) and (21)



(f) smallest eigenvalue of left hand side of eqs. (20) and (21)

Figure 16. Circular inclusion, results.

situations. We are able to prove stability and clearly point out the difference to Nitsche's method. While this are very nice features, we see some room for improvement as well.

The formulation carries the material matrix in the Nitsche terms as well as in the automatic stabilization terms. This poses no extra effort as long as geometric nonlinearities are not considered. In this case one would need the directional derivative of practically all terms involved in the

Journal publication: https://doi.org/10.1002/nme.4817

entire formulation. If one sets out to perform these analytically, this will result in a complicated formulation. Automatic differentiation would probably be the best way to go in this case, especially if one intends to avoid redoing the derivatives for each material law employed. The same is true, of course, for Nitsche's method but we would like to point out that the new formulation does not alleviate this burden.

The most important advantage compared to Nitsche's method is the avoidance of the solution of auxiliary eigenvalue problems for the estimation of the stability parameters. This is replaced by computing the inverse of a matrix similar to a mass matrix defined on the remaining part of the cut element. However, it is to be remarked that both problems are formulated on the cut elements and this leads to badly conditioned problems. Throughout the computations we noticed a very close correspondence between the results of Nitsche's method and the new method in the computed examples.

REFERENCES

1. Hughes TJR, Cottrell JA, Bazilevs Y. Isogeometric analysis: CAD, finite elements, NURBS, exact geometry and mesh refinement. *Computer Methods in Applied Mechanics and Engineering* 2005; **194**:4135–4195.
2. Cottrell JA, Hughes TJR, Bazilevs Y. *Isogeometric analysis: Towards Indegration of CAD and FEM*. John Wiley & Sons, 2009.
3. Parvizian J, Düster A, Rank E. Finite cell method – h- and p-extension for embedded domain problems in solid mechanics. *Computational Mechanics* 2007; **41**:121–133.
4. Düster A, Parvizian J, Yang Z, Rank E. The finite cell method for three-dimensional problems of solid mechanics. *Computer Methods in Applied Mechanics and Engineering* 2008; **197**:3768–3782.
5. Rank E, Ruess M, Kollmannsberger S, Schillinger D, Düster A. Geometric modeling, isogeometric analysis and the finite cell method. *Computer Methods in Applied Mechanics and Engineering* 2012; **249-252**:104–115, doi: doi:http://dx.doi.org/10.1016/j.cma.2012.05.022.
6. Hansbo P. Nitsche's method for interface problems in computational mechanics. *GAMM-Mitteilungen* 2005; **28**(2):183206.
7. Lew AJ, Buscaglia GC. A discontinuous-Galerkin-based immersed boundary method. *International Journal for Numerical Methods in Engineering* 2008; **76**(4):427–454, doi:10.1002/nme.2312. URL http://doi.wiley.com/10.1002/nme.2312.
8. Höllig K. *Finite Element Methods with B-Splines*. Frontiers in Applied Mathematics, SIAM Society for Industrial and Applied Mathematics, 2003.
9. Sanches R, Bornemann P, Cirak F. Immersed b-spline (i-spline) finite element method for geometrically complex domains. *Computer Methods in Applied Mechanics and Engineering* 2011; **200**(13-16):14321445, doi:10.1016/j.cma.2010.12.008. URL http://linkinghub.elsevier.com/retrieve/pii/S0045782510003531.
10. Rüberg T, Cirak F. Subdivision-stabilised immersed b-spline finite elements for moving boundary flows. *Computer Methods in Applied Mechanics and Engineering* 2012; **209-212**:266–283, doi:10.1016/j.cma.2011.10.007. URL http://linkinghub.elsevier.com/retrieve/pii/S0045782511003197.
11. Arroyo M, Ortiz M. Local maximum-entropy approximation schemes: a seamless bridge between finite elements and meshfree methods. *International Journal for Numerical Methods in Engineering* 2006; **65**(13):2167–2202, doi:10.1002/nme.1534. URL http://doi.wiley.com/10.1002/nme.1534.
12. Legrain G, Chevaugeon N, Drau K. High order x-FEM and levelsets for complex microstructures: Uncoupling geometry and approximation. *Computer Methods in Applied Mechanics and Engineering* 2012; **241-244**:172189, doi:10.1016/j.cma.2012.06.001. URL http://linkinghub.elsevier.com/retrieve/pii/S0045782512001880.
13. Duarte CA, Reno LG, Simone A. A high-order generalized FEM for through-the-thickness branched cracks. *International Journal for Numerical Methods in Engineering* 2007; **72**(3):325–351, doi:10.1002/nme.2012. URL http://doi.wiley.com/10.1002/nme.2012.

Journal publication: https://doi.org/10.1002/nme.4817

14. Joulaian M, Düster A. Local enrichment of the finite cell method for problems with material interfaces. *Computational Mechanics* 2013; doi:10.1007/s00466-013-0853-8. URL http://link.springer.com/10.1007/s00466-013-0853-8.
15. Babuška. The finite element method with penalty. *Mathematics of Computation* April 1973; **27**(122):221.
16. Zhu T, Atluri SN. A modified collocation method and a penalty formulation for enforcing the essential boundary conditions in the element free galerkin method. *Computational Mechanics* April 1998; **21**(3):211–222.
17. Babuška I. The Finite Element Method with Lagrangian Multipliers. *Numerische Mathematik* 1973; **20**:179–192.
18. Brezzi F. On the existence, uniqueness and approximation of saddle-point problems arising from lagrangian multipliers. *ESAIM: Mathematical Modelling and Numerical Analysis - Modlisation Mathmatique et Analyse Numrique* 1974; **8**(R2):129–151.
19. Barbosa HJC, Hughes TJR. Boundary Lagrange multipliers in finite element methods: Error analysis in natural norms. *Numerische Mathematik* 1992; **62**(1):1–15.
20. Glowinski R, Pan TW, Periaux J. A fictitious domain method for dirichlet problem and applications. *Computer Methods in Applied Mechanics and Engineering* 1994; **111**(34):283–303.
21. Hansbo P, Lovadina C, Perugia I, Sangalli G. A Lagrange multiplier method for the finite element solution of elliptic interface problems using non-matching meshes. *Numerische Mathematik* 2005; **100**(1):91–115, doi: 10.1007/s00211-005-0587-4. URL http://link.springer.com/10.1007/s00211-005-0587-4.
22. Mourad HM, Dolbow J, Harari I. A bubble-stabilized finite element method for dirichlet constraints on embedded interfaces. *International Journal for Numerical Methods in Engineering* 2007; **69**(4):772–793, doi:10.1002/nme.1788. URL http://doi.wiley.com/10.1002/nme.1788.
23. Burman E, Hansbo P. Fictitious domain finite element methods using cut elements: I. a stabilized Lagrange multiplier method. *Computer Methods in Applied Mechanics and Engineering* 2010; **199**(4144):2680–2686.
24. Nitsche J. Über ein variationsprinzip zur Lösung von Dirichlet-Problemen bei Verwendung von Teilräumen, die keinen Randbedingungen unterworfen sind. *Abhandlungen aus dem Mathematischen Seminar der Universität Hamburg* 1971; **36**(1):9–15.
25. Stenberg R. On some techniques for approximating boundary conditions in the finite element method. *Journal of Computational and Applied Mathematics* 1995; **63**(1-3):139148, doi:10.1016/0377-0427(95)00057-7.
26. Griebel M, Schweitzer M. A Particle-Partition of Unity Method Part v: Boundary Conditions. *Geometric Analysis and Nonlinear Partial Differential Equations*, Hildebrandt S, Karcher H (eds.). Springer Berlin Heidelberg, 2003; 519–542, doi:10.1007/978-3-642-55627-2_27. URL http://dx.doi.org/10.1007/978-3-642-55627-2_27.
27. Harari I, Shavelzon E. Embedded kinematic boundary conditions for thin plate bending by Nitsche's approach. *International Journal for Numerical Methods in Engineering* 2012; **92**(1):99–114, doi:10.1002/nme.4337. URL http://doi.wiley.com/10.1002/nme.4337.
28. Ruess M, Schillinger D, Bazilevs Y, Varduhn V, Rank E. 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* 2013; **95**(10):811–846, doi:10.1002/nme.4522. URL http://doi.wiley.com/10.1002/nme.4522.
29. Zander N, Kollmannsberger S, Ruess M, Yosibash Z, Rank E. The Finite Cell Method for Linear Thermoelasticity. *Computers & Mathematics with Applications* 2012; **64**(11):3527–3541, doi:10.1016/j.camwa.2012.09.002. URL http://linkinghub.elsevier.com/retrieve/pii/S0898122112005688.
30. Bazilevs Y, Hughes T. Weak imposition of Dirichlet boundary conditions in fluid mechanics. *Computers & Fluids* 2007; **36**(1):12–26, doi:10.1016/j.compfluid.2005.07.012.
31. Codina R, Baiges J. Approximate imposition of boundary conditions in immersed boundary methods. *International Journal for Numerical Methods in Engineering* 2009; **80**:13791405, doi:10.1002/nme.2662.
32. Gerstenberger A, Wall W. An embedded Dirichlet formulation for 3D continua. *International Journal for Numerical Methods in Engineering* 2010; **82**:537–563.
33. Baiges J, Codina R, Henke F, Shahmiri S, Wall WA. A symmetric method for weakly imposing Dirichlet boundary conditions in embedded finite element meshes. *International Journal for Numerical Methods in Engineering* 2012; **90**(5):636–658, doi:10.1002/nme.3339. URL http://dx.doi.org/10.1002/nme.3339.
34. Bernardi C, Maday Y, Patera A. A new non conforming approach to domain decomposition: The mortar element method. *Collège de France Seminar*, Brezis H, Lions JL (eds.). Pitman, 1994; 53–63.
35. Flemisch B, Puso M, Wohlmuth B. A new dual mortar method for curved interfaces:2d linear elasticity. *International Journal for Numerical Methods in Engineering* 2000; **63**:813–832.
36. Wohlmuth B. A mortar finite element method using dual spaces for the Lagrange multiplier. *SIAM Journal on Numerical Analysis* 2000; **38**:989–1012.

Journal publication: https://doi.org/10.1002/nme.4817

37. Puso MA. A 3d mortar method for solid mechanics. *International Journal for Numerical Methods in Engineering* 2004; **59**:315–336.
38. Lamichhane B, Stevenson R, Wohlmuth B. Higher order mortar finite element methods in 3d with dual Lagrange multiplier bases. *Numerische Mathematik* 2005; **102**:93–121.
39. Hansbo A, Hansbo P. An unfitted finite element method, based on Nitsche's method, for elliptic interface problems. *Computer Methods in Applied Mechanics and Engineering* 2002; **191**(47-48):5537 – 5552, doi:DOI:10.1016/S0045-7825(02)00524-8. URL http://www.sciencedirect.com/science/article/pii/S0045782502005248.
40. Becker R, Hansbo P, Stenberg R. A finite element method for domain decomposition with non-matching grids. *ESAIM: Mathematical Modelling and Numerical Analysis* November 2003; **37**(2):209–225.
41. Hansbo A, Hansbo P, Larson MG. A finite element method on composite grids based on Nitsche's method. *ESAIM: Mathematical Modelling and Numerical Analysis* 2003; **37**(03):495–514.
42. Hansbo A, Hansbo P. A finite element method for the simulation of strong and weak discontinuities in solid mechanics. *Computer Methods in Applied Mechanics and Engineering* 2004; **193**(33-35):3523–3540.
43. Sanders JD, Laursen T, Puso M. A Nitsche embedded mesh method. *Computational Mechanics* September 2011; **49**(2):243257.
44. Ruess M, Schillinger D, Özcan A, Rank E. Weak coupling for isogeometric analysis of non-matching and trimmed multi-patch geometries. *Computer Methods in Applied Mechanics and Engineering* 2014; :46–71.
45. Wriggers P, Zavarise G. A formulation for frictionless contact problems using a weak form introduced by Nitsche. *Computational Mechanics* Februar 2008; **41**(3):407–420.
46. Hansbo P, Hermansson J. Nitsche's method for coupling non-matching meshes in fluid-structure vibration problems. *Computational Mechanics* September 2003; **32**(1-2):134–139.
47. Szabó B, Babuška I. *Finite element analysis*. John Wiley & Sons, 1991.
48. Parvizian J, Düster A, Rank E. Topology optimization using the finite cell method. *Optimization and Engineering* 2011; :1–22URL http://dx.doi.org/10.1007/s11081-011-9159-x, 10.1007/s11081-011-9159-x.
49. Schillinger D, Ruess M, Zander N, Bazilevs Y, Düster A, Rank E. Small and large deformation analysis with the p- and B-spline versions of the finite cell method. *Computational Mechanics* 2012; **50**:445–478, doi:DOI: 10.1007/s00466-012-0684-z.
50. Schillinger D, Düster A, Rank E. The hp-d-adaptive finite cell method for geometrically nonlinear problems of solid mechanics. *International Journal for Numerical Methods in Engineering* 2012; **89**:1171–1202, doi: DOI10.1002/nme.3289.
51. Schillinger D, Dedè L, Scott MA, Evans JA, Borden MJ, Rank E, Hughes TJ. 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* 2012; **249–252**:116–150, doi: 10.1016/j.cma.2012.03.017.
52. Schillinger D. 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 2012.
53. Yang Z, Kollmannsberger S, Düster A, Ruess M, Burgkart R, Garcia E, Rank E. Computational Steering for Orthopeadics. *Computing and Visualization in Science* 2012; **14**:207–216.
54. Yang Z, Ruess M, Kollmannsberger S, Düster A, Rank E. An efficient integration technique for the voxel-based Finite Cell Method. *International Journal for Numerical Methods in Engineering* 2012; **91**:457–471.
55. Ruess M, Tal D, Trabelsi N, Yosibash Z, Rank E. The finite cell method for bone simulations: verification and validation. *Biomechanics and modeling in mechanobiology* 2012; **11**:425–437.
56. Düster A, Sehlhorst HG, Rank E. Numerical homogenization of heterogeneous and cellular materials utilizing the finite cell method. *Computational Mechanics* 2012; **50**:413–431, doi:doi:10.1007/s00466-012-0681-2.
57. Abedian A, Parvizian J, Düster A, Rank E. The finite cell method for the j2 flow theory of plasticity. *Finite Elements in Analysis and Design* 2013; **69**:37–47, doi:10.1016/j.finel.2013.01.006.
58. Duczek S, Joulaian M, Düster A, Gabbert U. Modeling of ultrasonic guided waves using the finite and spectral cell method. *International Journal for Numerical Methods in Engineering* 2014, accepted; .
59. Cai Q, Kollmannsberger S, Mundani R, Rank E. The finite cell method for spatially varying dispersions in coupled multispecies reactive transprot problems. *Proceedings of the International Conference on Computational Methods for Coupled Problems in Science and Engineering*, Kos Island, Greece, 2011.
60. Cai Q, Kollmannsberger S, Mundani R, Rank E. The finite cell method for solute transport problems in porous media. *Proceedings of the International Conference on Finite Elements in Flow Problems*, Garching, Germany, 2011.

Journal publication: https://doi.org/10.1002/nme.4817

61. Rank E, Kollmannsberger S, Sorger C, Düster A. Shell Finite Cell Method: A High Order Fictitious Domain Approach for Thin-Walled Structures. *Computer Methods in Applied Mechanics and Engineering* 2011; **200**:3200–3209, doi:DOI:10.1016/j.cma.2011.06.005. URL http://www.sciencedirect.com/science/article/pii/S0045782511002234.
62. Zander N, Bog S, Elhaddad M, Espinoza R, Hu H, Joly A, Wu C, Zerbe P, Düster A, Kollmannsberger S, *et al.*. FCMLab: A Finite Cell Research Toolbox for MATLAB. *Advances in Engineering Software* 2014; doi: 10.1016/j.advengsoft.2014.04.004.
63. Gamma E, Helm R, Johnson R. *Design patterns: Elements of reusable object-oriented software*. Addison-Wesley, 1995.
64. Schillinger D, Rank E. 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* 2011; **200**:3358–3380.
65. Schillinger D, Düster A, Rank E. The hp-d-adaptive finite cell method for geometrically nonlinear problems of solid mechanics. *International Journal for Numerical Methods in Engineering* 2012; **89**:1171–1202, doi: DOI10.1002/nme.3289.
66. Schillinger D, Kollmannsberger S, Mundani R, Rank E. The Finite Cell Method for Geometrically Nonlinear Problems of Solid Mechanics. *The World Congress on Computational Mechanics*, Sidney, Australia, 2010, doi: 10.1088/1757-899X/10/1/012170.
67. Abedian A, Parvizian J, Düster A, Khademyzadeh H, Rank E. Performance of different integration schemes in facing discontinuites in the Finite Cell Method. *International Journal of Computational Methods* 2013; **10**(03):1350 002–1–1350 002–24.
68. Gordon W, Hall C. Transfinite element methods: Blending function interpolation over arbitrary curved element domains. *Numerische Mathematik* 1973; **21**:109–129.
69. Embar A, Dolbow J, Harari I. Imposing Dirichlet boundary conditions with Nitsche's method and spline-based finite elements. *International Journal for Numerical Methods in Engineering* 2010; **83**(7):877–898.
70. Fernández-Méndez S, Huerta A. Imposing essential boundary conditions in mesh-free methods. *Computer Methods in Applied Mechanics and Engineering* 2004; **193**(12-14):1257 – 1275.

Journal publication: https://doi.org/10.1002/nme.4817

Further remarks Other possibilities of avoiding the estimation of parameters emerged later, see e.g. [48]. However, they do not maintain the symmetry of the system matrices. This is, of course, not an issue e.g. for convection problems which are unsymmetric by nature. Another drawback common to all Nitsche-like schemes, including the one published in [47] as well as [45, 39, 48], is the fact that the material law appears in some form or another in the boundary conditions. This dependency is inconvenient – not only because every new material description needs to be linearized also for the imposition of the boundary conditions, but also because the number of terms needed for a stable formulation becomes disproportionally high, see e.g. [49] for a thorough treatment of the subject in the context of fluid mechanics. This emerging complexity is not in line with the original idea of simplifying the numerical analysis by means of the FCM. Fortunately, many of the mentioned stability issues appear only in convection-dominated problems, only where Dirichlet boundary conditions need to be imposed (i.e. almost nowhere in structural mechanics problems), or in cases of very involved non-linearities. Therefore, the method of choice still is the penalty approach, which suffices in most cases. A *simple* approach to impose Dirichlet boundary conditions to provide stability for all types of physics on cut cells is still a very active line of research.

3.4 Enrichments

Commonly, the discretization of a domain with finite elements leads to an approximate and continuously differentiable solution. However, the analytic solution might exhibit local irregularities or local features which then have to be reflected in the approximation as well. These local features might be located arbitrarily within the finite elements. To be able to resolve these local features accurately, enrichments may be used.

The enrichments discussed in this chapter are not restricted to the FCM but naturally extend to boundary-conforming methods of high order in general. The following schemes are distinguished: direct local enrichments which require the knowledge of the type of the irregularity and its exact location and ones which do not. The former are briefly discussed in the paragraph Direct, local enrichments and how to avoid them, the latter are laid out in the paragraph Enrichments of h or hp-type.

3.4.1 Direct, local enrichments and how to avoid them

A classic representative of this type of methods are the X-FEM (see [50] for a review) and the Partition of Unity Method (PUM) [51]. In the context of the FCM, a direct local enrichment of this type was first introduced in [52]. In this line of research, the underlying idea is to discretize the field of unknowns u such that

$$u_{FE} = \sum_i \hat{u}_i \phi_i + \text{enrichment}.$$

The enrichment is constructed to directly represent the irregularity in the solution. In the PUM, it takes the form $\text{enrichment} = \sum_{j=1}^{n} N_j F \hat{a}_i$, where the functions N_j build a partition of unity, $\hat{a}_j$ are the new degrees of freedom corresponding to the enrichment, and F is the enrichment function itself. The most important advantage in this approach is that F can be any function that is linearly independent of $\sum_i \hat{u}_i \phi_i$. A disadvantage is that F is not easily

constructable for geometrically complex cases in the framework of high-order methods.

For specific situations, however, such as in contact or embedded interface problems, it is possible to avoid an explicit enrichment. The common feature in these problems is that it is sufficient to think of F as being a C^0 continuous function i.e. a function with a kink at the interface. This kink can be introduced equivalently by using two meshes, one for each body under consideration, to trim them at the interface in an FCM-like manner, and to glue them back together along their trimming surface. This *trim-and-glue* procedure does not add degrees of freedom to the problem, and it is not necessary to construct F explicitly. Figure 16 depicts a comparison between these different ideas for embedded interface problems in the context of the FCM.

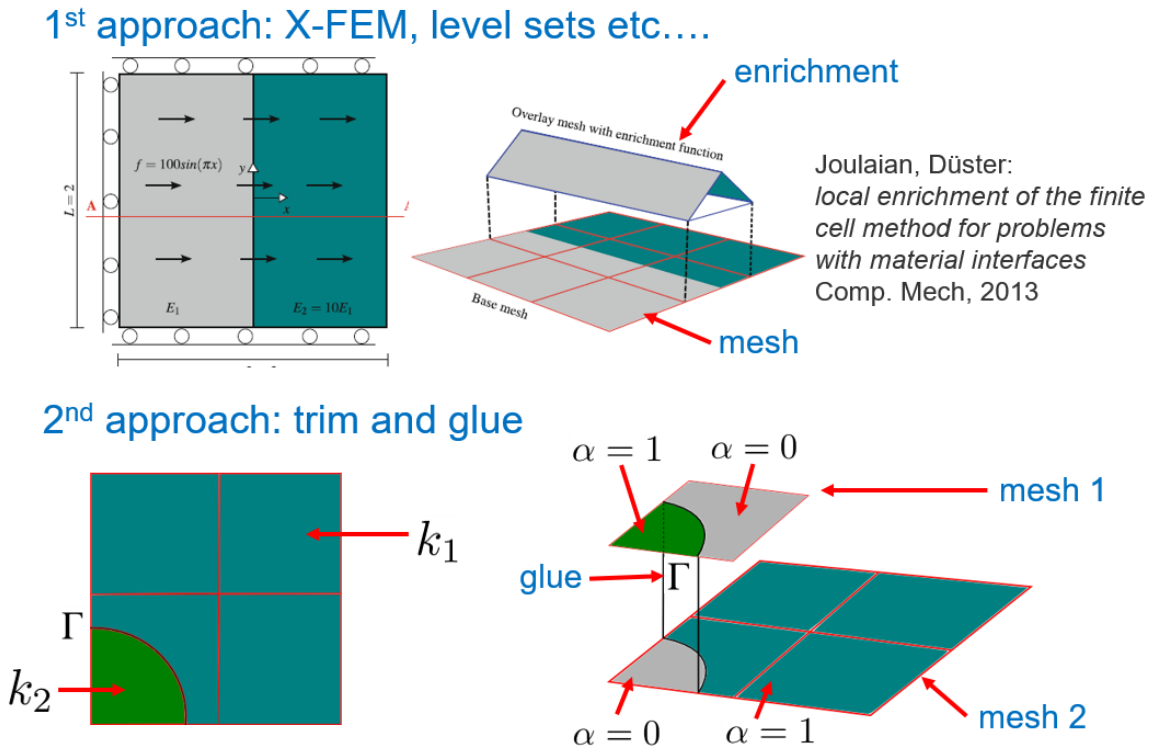
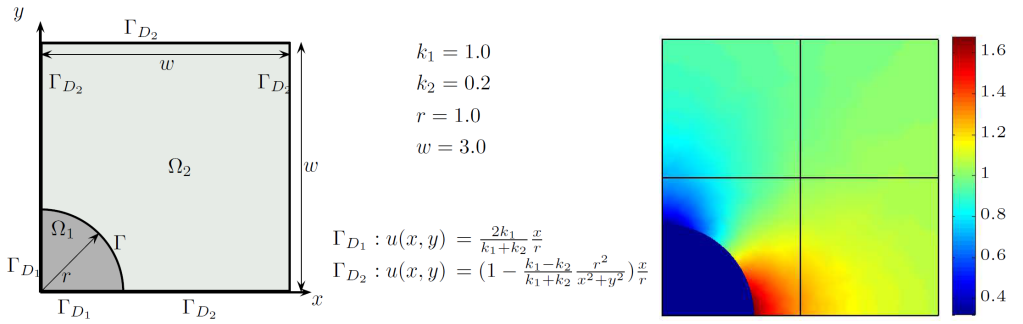


Figure 16: Two approaches to treat embedded interface problems, presented at [53]

The trim-and-glue approach is published in [47], using a combination of the newly developed coupling technique described in section 3.3. The approach to trim-and-glue is a popular choice and natural to embedded domain problems. A similar formulation to [47] is, for example, derived in [54]. However, this formulation leads to an asymmetric system of equations for originally symmetric problems, and it was only used for low orders. The extension in the context of this treatise lies in the fact that a new and variationally consistent coupling approach was developed for the FCM which preserves the possibility to maintain high-order convergence rates in embedded interface problems. Further, it is provably stable without stabilization parameters and delivers symmetric matrices for originally symmetric problems. Figure 17 shows the concrete problem setup, discretization and solution corresponding to the situation depicted in the lower part of fig. 16.

The trim-and-glue approach has also become very popular for solving multi-patch problems

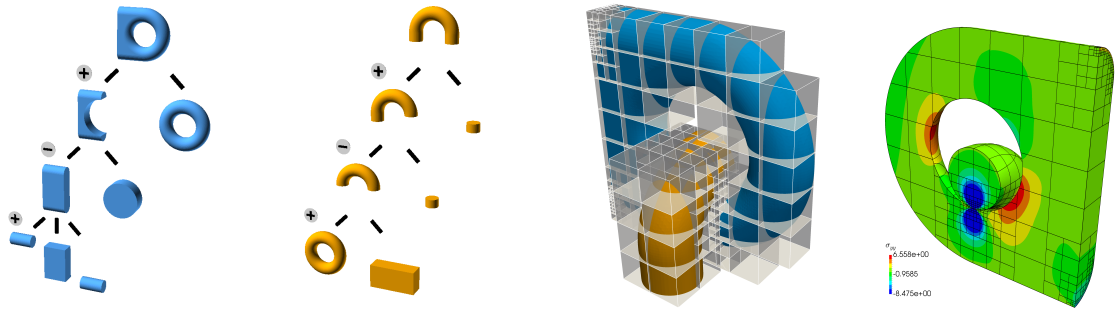


(a) setup

(b) discretization and solution of $u(x, y)$ in x

Figure 17: inclusion problem: setup, solution $u(x, y)$ and derivative of $u(x, y)$ in y along with discretization

in IGA, see e.g. [55] for one of its latest incarnations. Apart from that, it has become the method of choice in the FCM for contact problems, see fig. 18 as published in [56].



(a) CSG tree of the chain lock

(b) CSG tree of the chain link

(c) Each physical body is embedded in a refined FCM grid of order $p = 5$.

(d) stresses in vertical direction on a cutting plane bisecting the computational domain

Figure 18: Trim-and-glue approach for contact problems using CSG models and automatically recovered interfaces. [56]

Further, this approach obviously also exhibits the convenient capability to integrate different geometric models in one FCM application. As already mentioned in section 2, it is well suited for the bio-mechanical application depicted in fig. 10. Herein, the non-boundary conforming C^0 is represented by the trim-and-glue procedure, while the locally high gradients are treated by hierarchical enrichments of *hp* type, which will be discussed next.

3.4.2 Enrichments of *h* or *hp*-type

This paragraph addresses the case where the position or the type of the irregularity in the solution can not easily be identified or is not explicitly known. Indeed, the original motivation to introduce refinements of *hp*-type in the FCM is the fact that the FCM exhibits oscillations in strongly non-linear problems. These can be pronounced when shape functions spanning the fictitious domain are not able to adequately represent the change of one boundary condition

at one end of the cut cell to the boundary condition at the other end of the same cell. In these cases, the discretization is insufficiently fine. However, the implementation of classical *h*- or *hp*-finite elements is demanding. Part of the underlying challenge in classical *hp*-finite elements is that an element marked for refinement is completely replaced by new elements. This does not only change connectivities in the mesh data structure, but also induces the necessity of rewiring the local element degrees of freedoms. In this process, hanging nodes, edges, and faces produce cascades of dependencies which go well beyond the scope of one single finite element. This renders the necessary data-structures incomprehensible [57, 58].

Enrichment by hierarchical overlays A conceptually simpler remedy to *hp* refinement is offered by the *hp-d* method. Its underlying idea is to view the total discrete solution u_{FE} as being composed of a part which represents the smooth solution u_p and another part u_h which represents an irregularity, i.e.:

$$u_{FE} = u_p + u_h \tag{2}$$

Herein, u_h acts as an enrichment[4]. The smooth part of the solution u_p is spanned on a base mesh with relatively large elements, and u_h is spanned on a finer mesh overlaying the base mesh. This idea was first published in [59] and has been picked up numerous times, see e.g. [60] for an in-depth review of its variants.

The basic concept of the *hp-d* method was first applied to the FCM in [61] with the additional enhancement to enrich towards the irregularity of the solution by organizing these enrichments on hierarchical grids. To this end, B-spline basis functions were used. The concept was then transferred to classical *hp*-methods in [62]. Figure 19 presents the one-dimensional situation and its two-dimensional counterpart.

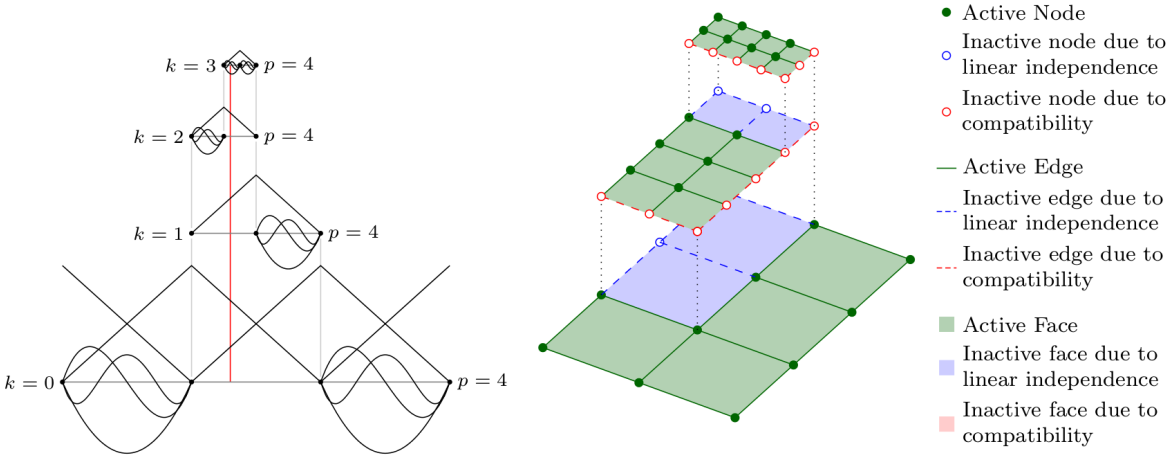


Figure 19: The multi-level hp-FEM concept in 1D and 2D as presented in [62]

In general, the enrichment requires a) linear independence of u_p and u_h, and b) the imposition of homogenous Dirichlet boundary conditions on the boundary of the overlay. The following simple rules for the construction of hierarchical higher order *hp*-discretizations suffice to obey

[4] The same enrichment is, of course, carried out for the test functions v as well, even if only u is mentioned explicitly in the sequel

conditions a) and b). Compatibility is ensured by applying homogeneous boundary conditions on all boundaries of the overlay mesh. In one dimension, this translates to deactivating all nodal degrees of freedom on the overlay meshes which correspond to the boundary of the overlay. Linear independence is guaranteed by deactivating the high-order modes on the lower levels. Thereby, high-order shape functions are *h*-refined as well. This is a hidden but important point in the construction of efficient *hp* discretizations of this type, see e.g. the investigations in [63]. The simple rule set of activating and de-activating nodal and edge modes directly translates to two- and three-dimensional discretizations.

The rule set can be kept simple because the connectivity of the base mesh never changes. Instead, the elements of the base mesh always remain in the discretization. This concept of *refine-by-overlay* is contrary to the *refine-by-replacement* paradigm usually followed in *hp*-methods, and it has many implementational advantages. For example, the calls for evaluating the shape functions can simply be implemented as recursive calls initiated from the base element. This concept also eases the management of *hp* discretizations with multi-level hanging nodes transiently, i.e. it allows to accurately capture moving singularities, see [60, 64].

The trade-off for the comparatively simple implementation is the interlink of many shape functions, due to the remaining linear modes of the base elements that are refined. This can lead to a high number of degrees of freedom within one base element. The complexity of the algorithm was examined in [62], where it was demonstrated that multi-level *hp* discretizations lead to an exponential convergence of the error *with respect to the run time* concerning the computation of the stiffness matrix. Thus, the complexity of the involved algorithms does not limit the convergence with respect to the run time. A further important aspect is that the conditioning of the resulting linear system could suffer due to the loss of orthogonality of the shape functions. This case occurs, for example, with non-uniform distributions of material parameters or skew meshes. Issues related to the solver were closely investigated in [65], but the effect on the necessary number of iterations of a preconditioned CG solver was found to be negligible for the investigated cases. Next, the multi-level *hp*-method is presented in-depth by including the original journal publication [62].

Journal Publication

title:	Multi-Level *hp*-Adaptivity: High-Order Mesh Adaptivity without the Difficulties of Constraining Hanging Nodes
authors:	N. Zander, T. Bog, S. Kollmannsberger, D. Schillinger, E. Rank
published at:	*Computational Mechanics*
publisher:	Springer
year:	2015
volume:	55
pages:	499-517
doi:	https://doi.org/10.1007/s00466-014-1118-x

Multi-Level hp-Adaptivity: High-Order Mesh Adaptivity without the Difficulties of Constraining Hanging Nodes

Nils Zander*[1], Tino Bog[1], Stefan Kollmannsberger[1], Dominik Schillinger[2], and Ernst Rank[1]

[1]Chair for Computation in Engineering, Technische Universität München, Arcisstr. 21, 80333 München, Germany
[2]Department of Civil Engineering, University of Minnesota, 500 Pillsbury Drive S.E., Minneapolis, 55455 MN, USA

Abstract

The implementation of *hp*-adaptivity is challenging as *hanging* nodes, edges, and faces have to be constrained to ensure compatibility of the shape functions. For this reason, most *hp*-code frameworks restrict themselves to 1-irregular meshes to ease the implementational effort. This work alleviates these difficulties by introducing a new formulation for high-order mesh adaptivity that provides full local *hp*-refinement capabilities at a comparably small implementational effort. Its main idea is the extension of the *hp-d*-method such that it allows for high-order overlay meshes yielding a *hierarchical*, multi-level *hp*-formulation of the Finite Element Method. This concept enables intuitive refinement and coarsening procedures, while linear independence and compatibility of the shape functions are guaranteed by construction. The proposed method is demonstrated to achieve exponential rates of convergence—both in terms of degrees of freedom and in run-time—for problems with non-smooth solutions. Furthermore, the scheme is used alongside the Finite Cell Method to simulate the heat flow around moving objects on a non-conforming background mesh and is combined with an energy-based refinement indicator for automatic *hp*-adaptivity.

Keywords: high-order FEM, automatic *hp*-adaptivity, arbitrary hanging nodes, Finite Cell Method

1 Introduction

One requirement for efficient numerical schemes is the ability to adapt the discretization locally in domains of interest to improve the quality of approximation. Within the framework of the Finite Element Method, a major challenge of this adaptivity is to guarantee the compatibility requirements of the shape functions. In particular, mesh *irregularities* caused by *hanging nodes* must be avoided in schemes based on the Finite Element Method.

Since the early 1980's, various refinement strategies have been introduced for this purpose. Their common idea is to ensure compatibility by appropriately constraining hanging nodes (see e.g [1; 2]). Although this approach has proven to work well for various applications, the implementation of the constraints bears challenges. This becomes more difficult in the context of *hp*-refinement, as here also edge- and face-modes have to be constrained correctly. For this reason, *hp*-algorithms are typically restricted to *1-irregular* meshes to reduce the implementational complexity (see e.g. [1; 3; 4; 5; 6; 7; 8; 9]). Only recently, advanced numerical schemes have been implemented that allow

*nils.zander@tum.de, Corresponding Author

Preprint submitted to *Computational Mechanics* November 10, 2014
The final publication is available at `http://link.springer.com/article/10.1007/s00466-014-1118-x`

Journal publication: https://doi.org/10.1007/s00466-014-1118-x

for arbitrary-level hanging nodes (see e.g. [10; 11; 12; 13]). However, the idea to post-constrain hanging nodes remains unchanged, which requires a sophisticated algorithmic treatment.

To overcome these challenges, the authors suggest an alternative refinement strategy that avoids the difficulties associated with arbitrary-level, high-order, hanging nodes. The approach is based on the idea of *hp-d*-refinement, in which a high-order base mesh is superposed with a finer *h*-overlay mesh in the domain of interest (see [14]). In this way, fine-scale solution characteristics can be captured accurately on the overlay mesh, while large-scale characteristics are represented by the high-order base mesh. By applying homogeneous Dirichlet boundary conditions on the overlay mesh, hanging nodes are avoided by definition. Recently, this approach has been extended in different directions.

Schillinger *et al.* show that using hierarchical *h*- and NURBS-overlay meshes allows for adaptive *h*-refinement (see [15; 16; 17; 18]). In [19], the *hp-d* idea is successfully combined with partition of unity approaches to better approximate discontinuities occurring at material interfaces. In the present work, the idea is extended to adaptive high-order-refinement by employing a *high-order* hierarchical overlay mesh. This *multi-level hp*-algorithm offers the full capabilities of *hp*-adaptive-methods, while allowing for arbitrary irregular meshes and a high flexibility in the discretization.

The essential ideas of this new refinement method will be outlined in detail in Section 2 and its implementation will be discussed briefly in Section 3. In Section 4, the benefits of this new approach are discussed. In a first example, it is shown that the suggested multi-level *hp*-approach yields exponential rates of convergence for problems with non-smooth solution characteristics. Two further examples demonstrate that the introduced agile data structure also allows to easily refine and coarsen the discretization in the course of transient simulations. The work closes with a concluding outlook in Section 5.

2 The Multi-Level *hp*-Algorithm

As mentioned in the introduction, a major challenge of adaptivity in the context of Finite Elements is the problem of mesh irregularities caused by *h*-refinement. In this section, the idea of the multi-level *hp*-algorithm is outlined, which allows for high-order mesh adaptivity without the difficulties of hanging nodes. Prior to this description, classical refinement schemes are briefly discussed to motivate the need for a novel discretization scheme.

2.1 Classical Refinement Schemes

Following [21], the most common *h*-refinement schemes can be categorized in the following three groups:

Methods based on *mesh regeneration* create a completely new mesh in the domain of interest. This approach has the advantage that the new mesh is independent of the old one, which allows for great flexibility and good mesh quality. In particular, the new mesh can be kept regular, and the algorithms of the FE-kernel can be kept simple. However, the approach has the disadvantage that automatic mesh generation in 3D can be difficult. Furthermore, the data transfer between the different meshes has to be handled.

An alternative possibility for adapting the discretization is to keep the number of unknowns constant but to adjust the node positions instead. This idea known as *r-refinement* has the advantage that the mesh can be kept regular (see e.g. [22]). However, this approach is difficult to automize [21].

The third category is *element subdivision*. Here, the idea is to *replace* elements with a high error contribution by smaller elements, which comply to the boundary of the original element. The advantage of this idea is that it provides flexibility while allowing for automization of the algorithm.

However, one major difficulty of this approach is the fulfillment of the *compatibility* requirement. This demands that patch shape functions are C^{m-1} continuous between interconnected elements and

Journal publication: https://doi.org/10.1007/s00466-014-1118-x

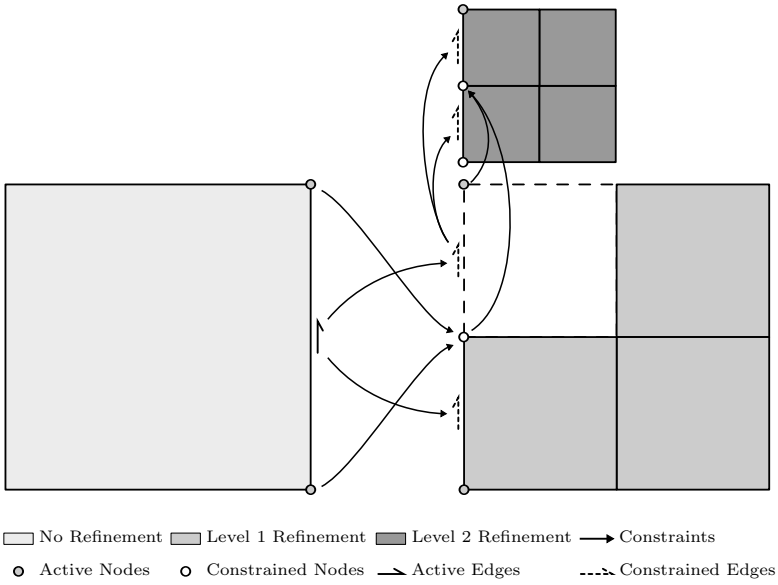


Figure 1: Multiply constrained modes occurring in classical *hp*-refinement strategies (following e.g. [1; 2; 20])

C^m continuous within an element. Thereby, m denotes the variational index of the problem under consideration (see e.g. [23; 24; 25; 26]). In the case of element subdivision, the mid-side nodes of a refined element have no suitable representation in coarse neighboring elements (see Figure 1). These *hanging nodes* render the mesh *irregular*, as the shape functions between the interconnected elements are no longer C^0 continuous. This means that the compatibility is no longer fulfilled for e.g. thermal or elastic problems ($m = 1$). As this requirement is essential for ensuring the convergence of the Finite Element Method, hanging nodes require an appropriate treatment.

One way of avoiding such mesh irregularities is to also subdivide the adjacent elements to provide a transition between the fine and the coarse discretization (see e.g. [27; 28]). Although mesh regularity can be ensured in this way, the transition elements introduce additional degrees of freedom, with the result that the refinement looses its local character. Furthermore, the transition elements can be highly distorted. Both effects possibly have a negative influence on the numerical accuracy and the rate of convergence (see e.g. [29; 28; 10; 30]). Moreover, an automatic generation of transition elements is challenging, in particular in case of structured three-dimensional meshes, which typically demands for templated solution strategies (see e.g. [31; 32]).

An alternative idea suggested by [33; 34] is to enhance the coarse elements with suitable shape modes, which recover the C^0 continuity between interconnected elements. Although this idea is very appealing and leads to optimal convergence, it has the disadvantage that an extension to high-order shape functions is not straightforward. Furthermore, the formulation requires that the maximum difference between the refinement level of adjacent elements is limited to one (i.e. 1-irregular meshes). Although an extension of the approach to k-irregular meshes is possible in principle, the authors do not know of any published work on this topic.

The most common idea to handle hanging nodes is to allow for irregular meshes during the refinement procedure and to ensure the compatibility requirement by appropriately constraining hanging nodes in a post-step. The essential idea is to express the fine-scale shape function of the hanging node in terms of the respective coarse-scale *parent* modes of the un-refined neighboring element. In the context of linear shape functions, this can be carried out easily, as the unknown of the hanging

Journal publication: https://doi.org/10.1007/s00466-014-1118-x

middle node can be expressed by a simple interpolation between the coarse nodal modes (see e.g. [1; 2; 20]).

Although the extension of this approach to higher-order Finite Elements (i.e. *hp*-FEM) follows the same idea, this step bears some challenges. The major difficulty is that in the presence of higher-order modes, not only hanging nodes but also hanging edges and faces have to be handled. The main difference to the linear case is that a simple interpolation between the coarse and fine-scale modes is not possible for edge and face modes. Instead, constraining them requires a suitable *projection* of the fine modes onto the coarse ones. In the literature, different methods have been presented how this can be achieved most efficiently (see e.g. [1; 2; 10; 11; 13; 20] for more details).

One additional technical challenge arises when hanging (nodal, edge, or face) modes have to be constrained using parent modes that are constrained themselves (see Figure 1). This typically occurs when the refinement algorithm requires for a sudden change in the discretizaton accuracy, i.e. a large difference in the refinement level of adjacent elements. As these *multiply constrained* modes significantly increase the implementational complexity, most *hp*-Finite Element codes restrict themselves to 1-irregular meshes such that only elements with unconstrained modes can be refined (see e.g. [1; 2; 3; 4; 5; 6; 7; 8; 9]). This commitment, however, imposes several limitations on the refinement flexibility. As noted by [1; 4], 1-irregular meshes introduce unwanted boundary layers as the refinement has to spread out to insure the compatibility of the basis function. As in the case of transition elements, the refinement thus loses its local character [4]. In their studies, [10; 13] show that schemes bound to 1-irregular refinements introduce significantly more degrees of freedom than actually necessary. One possibility to reduce this effect is to use an anisotropic refinement, which in turn might lead to dead-lock situations, which have to be handled explicitly (see e.g. [1; 35]). New methods have been published recently which allow for arbitrary hanging nodes and as such avoid refinement spreads (see e.g. [10; 11; 30; 12; 13]). The idea to post-constrain the hanging modes, however, remains unchanged so that these approaches still demand for a sophisticated algorithmic framework.

The present work aims at circumventing these implementational difficulties by avoiding hanging nodes by definition and in this way allowing for more flexibility, while achieving comparable approximation accuracy. The essential ideas of this approach are outlined next.

2.2 Refinement via Superposition

As discussed in the previous section, classical refinement schemes are based on the idea of *replacing* the coarse elements by finer elements in the domain of interest. An alternative approach—followed in this work—is to keep the base discretization unchanged and to improve the approximation accuracy by *superposing* a finer overlay mesh, on which small-scale solution characteristics can be captured.

This idea dates back to the pioneering work of Mote who, in 1971, introduced the Global-Local Finite Elements by combining the conventional and finite element Ritz method following the idea of mesh superposition (see [36] and e.g. [37] for a comprehensive review of the method). Since then, this idea has been extended and applied successfully in the context of e.g. the hierarchic Finite Element Method (see e.g. [38]), embedded localization zones (see e.g. [39; 40; 41]), the spectral overlay method (see e.g. [42]), the *hp-d*-refinement method (see e.g. [14; 43]), the *s*-version of the Finite Element Method (see e.g. [44; 45; 46; 47; 48]), and the adaptive local overlaying grid method (see e.g. [49]). Although originating from a different direction, also partition of unity-based methods such as the eXtended or the Generalized Finite Element Method can be regarded as superposition methods, as the standard Finite Element space is augmented by additional functions that capture special solution characteristics (see e.g. [50; 51; 52] for a review of these methods).

The unifying idea of all these approaches is to decompose the final approximation u into the base

Journal publication: https://doi.org/10.1007/s00466-014-1118-x

mesh part u_b and the overlay part u_o:

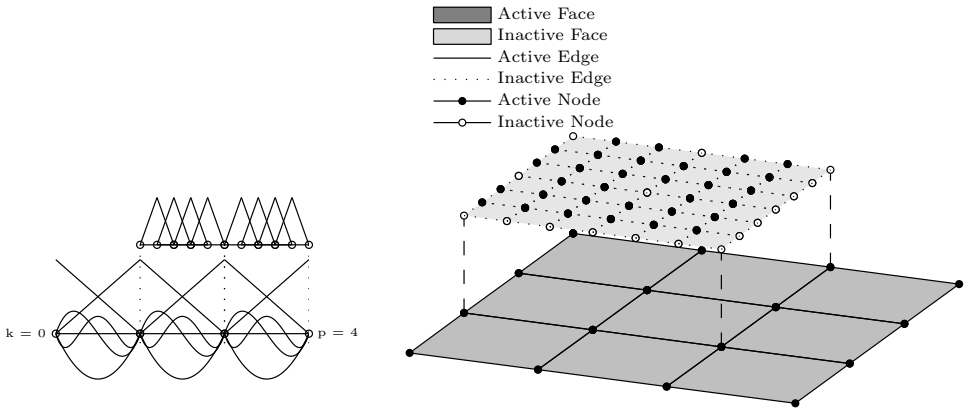
$$u = u_b + u_o.$$

The compatibly of the solution is ensured by constraining u_o via homogeneous Dirichlet boundary conditions on the boundary of the overlay mesh. In this way, the global and local scales can be captured separately without modifications of the base mesh. The idea of mesh superposition, therefore, allows for a high flexibility in the discretization. In particular, the previously described challenges of hanging nodes do not arise *by construction*, as the overlay solution is zero *by definition* on the transition between the base and overlay mesh.

The current work extends this idea to combine the simplicity of the superposition methods with the approximation power of *hp*-FEM. The aim is to construct hierarchical *hp*-like meshes to yield exponential convergence for non-smooth problems without the difficulties of hanging nodes. At the same time, the method is designed to deliver an agile data structure, such that the discretization can be refined and coarsened adaptively during transient simulations without any limitations due to mesh irregularities. Following the chronology of the method, this *multi-level hp*-approach is derived in three steps in the upcoming sections.

2.2.1 The *hp-d*-Idea

As outlined in the previous section, the essential idea of the *hp-d*-refinement algorithm is to overlay multiple meshes for approximation purposes. In contrast to the other methods mentioned, large-scale solution-characteristics are thereby captured on a coarse *high-order* base mesh, which spans the full domain of the boundary value problem under consideration. To also capture fine solution features, this base mesh is overlayed by a finer *h*-mesh in the domain of interest (see Figure 2).



(a) 1D-Case (b) 2D-Case

Figure 2: The *hp-d*-approach introduced in [14; 43]: The coarse high-order base mesh captures the large-scale solution, while fine-scale characteristics are captured using the finer *h*-mesh that overlays the base mesh in the respective domain of interest.

Due to this decomposition of the solution, two major aspects demand for special consideration: the *linear independence* and the *compatibility* of the basis functions. However, the hierarchical structure of the overlay elements renders both requirements easy to fulfill.

The linear independence of the shape functions is guaranteed by first ensuring that the elements

Journal publication: https://doi.org/10.1007/s00466-014-1118-x

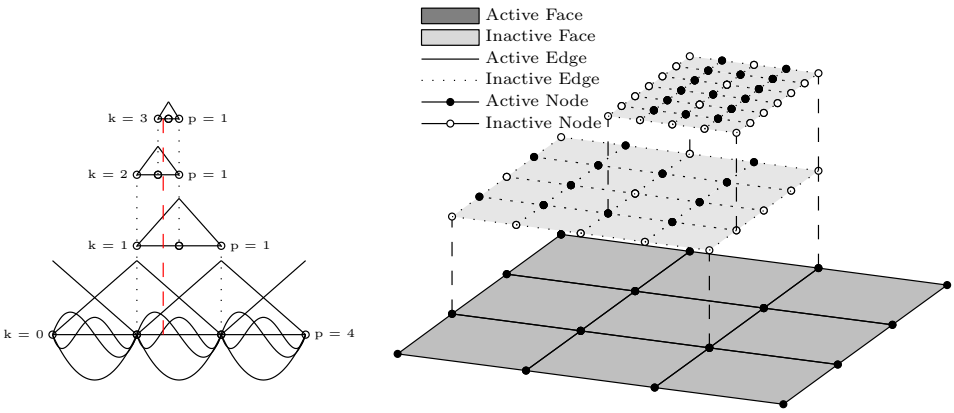
of the second mesh do not overlap the element boundaries of the base mesh and secondly by "deactivating" all nodal modes in the overlay mesh that are direct descendants of base mesh nodes.

The compatibility of the shape functions is ensured by applying homogeneous Dirichlet boundary conditions to the overlay mesh, which guarantees that the patch shape functions are C^0 continuous between interconnected elements of the respective patch. In this way, the compatibility is ensured *by construction*. This simple and yet sufficient approach has the additional advantage that the problem of hanging nodes can be circumvented completely, as no degrees of freedom are present on the boundary of the fine overlay mesh. This idea is shown in Figure 2b depicting inactive nodes as hollow circles, whereas active nodes are filled. Similarly, dashed edges and light-gray faces are inactive.

It was shown by [14; 43; 53; 54] that this approach can be successfully applied to structural mechanics, reaction-diffusion, and dimensionally reduced problems.

2.2.2 The Hierarchical *hp-d*-Idea

The *hp-d*-approach has been picked up by Schillinger *et al.* and extended in a hierarchical sense (see e.g. [15; 16; 17; 18]). This idea is depicted in Figure 3, which shows how the original approach is extended naturally by superposing different overlay meshes. As in Figure 2b, the direct descendant and the boundary nodes are deactivated on all levels, whereas the base mesh remains unchanged. In this way, arbitrary irregular meshes can be created easily by simply arranging the overlay meshes such that the respective mesh boundaries coincide.



(a) 1D-Case ($p = 4$, k: refinement level, dashed line: singular point)

(b) 2D-Case

Figure 3: Conceptional idea of the hierarchical *hp-d*-method (following e.g. [15])

Schillinger *et al.* show that this extension can be applied to significantly improve the approximation accuracy in the context of geometrically non-linear structural mechanics and fictitious domain problems. In [15; 18], this idea is extended to the context of isogeometric analysis (IGA) to adaptively refine B-spline and NURBS elements. In [55], it is shown that this approach also yields excellent results in the framework of collocation methods.

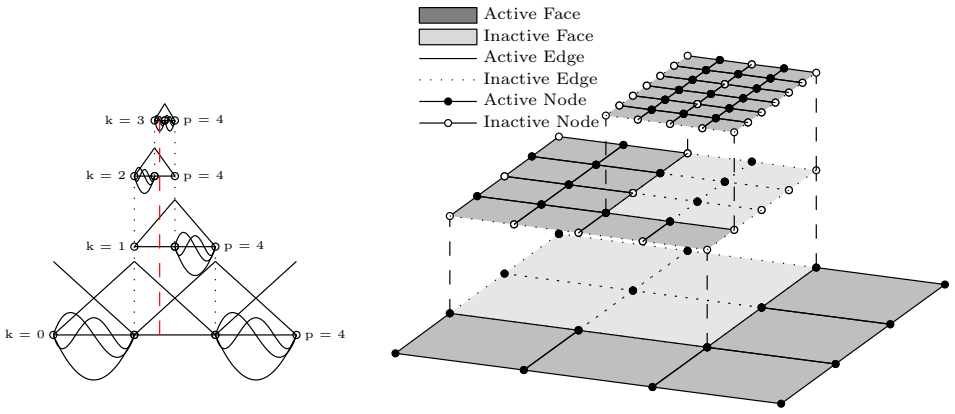
All these publications, however, focus on applying pure *h*-refinement on the overlay meshes. Although this approach allows to obtain exponential convergence for smooth problems when increasing the order of the base mesh, the convergence characteristics turn algebraic when the analytical solution incorporates singularities (see e.g. [17; 55]). To further improve the convergence properties for

Journal publication: https://doi.org/10.1007/s00466-014-1118-x

these non-smooth problems, the aforementioned approach needs to be extended. This is the focus of the multi-level *hp*-algorithm, whose details will be outlined in the next section.

2.2.3 The Multi-Level *hp*-Idea

The essential idea of the multi-level *hp*-approach is to employ hierarchical, *high-oder* overlay meshes to yield the full potential of *hp*-refinement schemes (see Figure 4). For this extension linear independence and compatibility of the shape functions have to be reconsidered.



(a) 1D-Case ($p = 4$, k: refinement level, dashed line: singular point)

(b) 2D-Case

Figure 4: Conceptional idea of multi-level *hp*-FEM

To guarantee a linearly independent basis, the high-order shapes introduced on a new overlay mesh have to be removed from the respective meshes on lower levels. As shown in Figure 4a for the one-dimensional case, the high-order shapes are thus distributed over the different levels and only elements with no further refinements are equipped with high-order shape functions. In contrast, on all lower elements only linear shape functions are spanned, which ensure the C^0 continuity between the different elements and levels. This idea naturally extends to two dimensions as shown in Figure 4b. Just as for the previous case, all refined edges and faces are deactivated and only non-refined elements incorporate the high-order modes. In this way, linear dependencies between the shape functions of the different levels can be easily avoided *a priori*.

Similar to the *hp-d*-approach, the compatibility of the shape functions is ensured by applying homogeneous Dirichlet boundary conditions on the overlay meshes. In the high-order case, however, this requires not only for "deactivating" the respective nodes but also the edges (and faces in 3D) on the mesh boundary. In Figure 4b, this is depicted by dashed lines on the mesh edges. Only on the domain boundary, higher-level edge modes can be active as no compatibility has to be ensured here. Again, this requires that the respective lower-level edges are disabled to ensure linear independence.

These simple extensions of the aforementioned *hp-d*-idea allow to formulate a simple *hp*-refinement algorithm. In particular, the deactivation of the respective boundary modes of the overlay meshes circumvents the difficulties of hanging nodes as these are avoided *by construction*. As depicted in Figure 4b, the refinement can therefore be adjusted flexibly without any restriction to 1-irregularities.

A similar multi-level superposition approach has been introduced by Fish in [48]. The current work, however, aims at constructing *hp*-FEM like discretizations via a clear hierarchy in the mesh structure. This allows to define a simple set of rules, which ensure compatibility and linear independence *by*

Journal publication: https://doi.org/10.1007/s00466-014-1118-x

construction. These lead to an agile data structure in which a change of the discretization in a transient simulation comes naturally without posing additional challenges. This process is outlined in the following section.

3 Aspects of Implementation

As outlined in the previous section, the main challenge of the multi-level *hp*-refinement is to ensure linear independence and compatibility of the basis functions. This demands for deactivating the "correct" topological components. The following section briefly outlines the computational implementation of this task in the context of an object oriented high-order FEM framework.

3.1 Data Structure

The data structure required for implementing the multi-level *hp*-algorithm is simple. All that is needed are four classes that represent the different types of topological components (node, edge, face, solid) and one class to represent the element (see Figure 5).

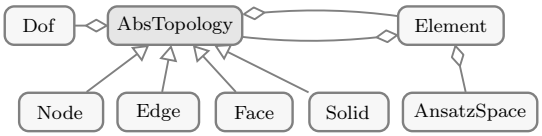


Figure 5: UML representation of the multi-level *hp*-data structure

As in most Finite Element programs, the topological components are used to represent the connectivity information of the mesh. In the context of high-order FEM, these classes additionally handle the degrees of freedom and the polynomial degree of the associated shape functions (nodal-, edge-, face-, and volume-/internal-modes). However, the topology does not actually create shape functions. This is the responsibility of the element class, which is defined *on* such a topological component (i.e. its "support").

Individual topological components can now be "deactivated" by simply setting the polynomial degree of their modes to zero. In this case, the respective topological component will not allocate any degrees of freedom, and, accordingly, the element will not create any shape functions associated with this component. In this way, selected shape functions can be removed easily from the ansatz space to ensure compatibility and linear independence.

To simplify the organization of this deactivation process, every element knows the topological component it is defined on, and every topological component holds a list of adjacent elements. This is depicted in Figure 6a as an example for one node and one edge by the arrows pointing to the respective neighboring elements.

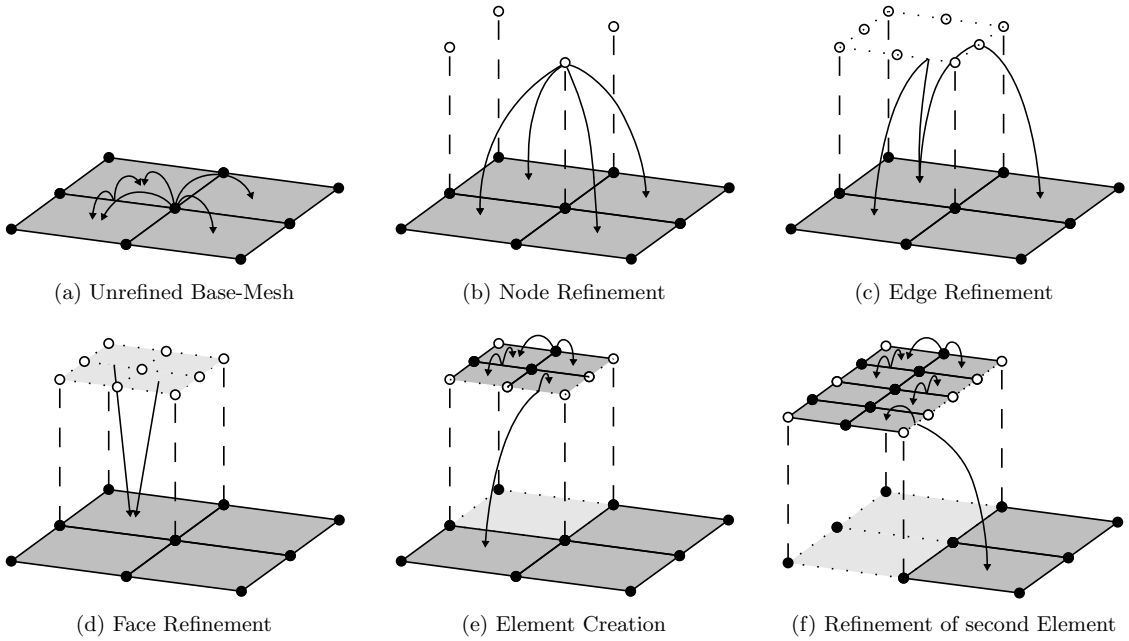
The data structure is extended in a tree-like manner to also allow for a hierarchical refinement. Thus, every topological component holds a reference to its sub-components and every element stores its sub-elements. Finally, every element also knows the refinement level it has been created on.

The individual steps to create these refined objects are described in the next subsection.

3.2 Refinement Procedure

Following the data structure described above, the refinement of the discretization requires to subdivide the respective topological components such that new elements can then be defined on this sub-topology. The different steps of this refinement procedure are discussed in the following.

Journal publication: https://doi.org/10.1007/s00466-014-1118-x



(a) Unrefined Base-Mesh (b) Node Refinement (c) Edge Refinement

(d) Face Refinement (e) Element Creation (f) Refinement of second Element

Figure 6: Implementation steps of multi-level *hp*-FEM. The arrows depict the adjacency relation between the created topological sub-components and the elements.

Node Refinement The first step is to refine the nodes of the element. For this purpose, four *new* nodes are created as sub-nodes of their respective parents (see Figure 6b). These sub-nodes are geometrically identical to their parents and, thus, overlay them on the next refinement level. Most importantly, the new sub-nodes inherit the list of adjacent elements of their parents and are, thus, still regarded as being connected to the original coarse element. In Figure 6b, this is illustrated by the arrows pointing from one new node to the elements from which this node is originating.

Edge Refinement In the second step, the edges are refined. As for the nodes, this requires to create (two) new sub-edges per parent edge and their corresponding middle nodes. Again, these new components (sub-edges *and* mid-nodes) inherit their adjacent elements from their parent edge and are, therefore, connected to the coarse elements. This is depicted in Figure 6c for one new edge and one new mid-node.

Face Refinement The third step is the face refinement. As in the previous steps, four new sub-faces are created together with their middle node and edges (see Figure 6d). It is important to note that, at this point, only the topological components have been subdivided and *no* new elements have been created yet. Just as in the previous step, the new components (sub-faces *and* mid-nodes and edges) are, therefore, still adjacent to the coarse elements.

Element Refinement In this final step, the actual element is refined. To this end, four new elements are created using the four sub-faces as their support (see Figure 6e). As outlined in Section 2.2.1, these new sub-elements do *not* replace the original element but overlay it instead. The coarse element is therefore *not* deleted but still exists on the base level. However, it deregisters itself from

Journal publication: https://doi.org/10.1007/s00466-014-1118-x

the newly created sub-topologies and registers the respective sub-elements instead. The new sub-components are thus now partly adjacent to elements on different levels. This information is essential to deactivating the correct components, which will be described in the following sub-section.

3.3 Component Deactivation

As discussed previously, linear independence and compatibility of the basis functions is ensured by deactivating the "correct" topological components. With the setup described in the previous paragraph, this can be done easily by applying the following simple rules:

Compatibility is ensured by deactivating all topological components of the overlay mesh whose adjacency list contains elements of different levels. In this way, components of the overlay mesh on the boundary of the refinement zone will have no degrees of freedom. This ensures C^0 continuity between interconnected elements and avoids hanging nodes and edges by construction.

Linear independence is ensured by deactivating all topological components of the base mesh that have (active) sub-components and all nodes on the overlay mesh directly descending from base nodes.

It is interesting to note that these simple ideas also allow to naturally activate the correct topological components on the domain boundary. As the topological components at the boundary are only adjacent to the refined elements, they are activated automatically without the necessity of any additional adjustments.

Furthermore, if now a second element is refined, the adjacency relations automatically update, and the new sub-components can be re- or deactivated without any difficulty (see Figure 6f). Finally, this procedure also allows for multiple levels of overlay meshes by simply applying the algorithm recursively on the newly created sub-elements.

3.4 Numerical Integration

In addition to the questions of compatibility and linear independence, also the correct numerical integration of the element stiffness matrix demands for special consideration when following the idea of mesh superposition. In the context of the *hp-d-* or multi-level *hp*-method, however, the clear hierarchical structure of the overlay elements can be exploited to also simplify this task. In particular, no complex subdivions of the overlay elements are required as these may only overlap *one* parent element. An exact integration only demands for subdividing the integration domain according to the inner-element boundaries. This poses no implementational difficulties, as the sub-elements are constructed by recursively bisecting the parent element (see Figure 7). On each of the resulting sub-domains, standard Gaussian quadrature points are distributed, on which the respective non-zero shape functions are evaluated to compute the element matrices.

The figure, however, also shows a potential performance bottleneck of the method. As the approximation accuracy is increased by superposing different elements, the number of shape functions with overlapping support increases. This results in a denser system of equations with a higher bandwidth compared to a conventional *hp*-scheme. Furthermore, the superposition leads to more shape functions being non-zero on each integration point. All three effects possibly have negative influences on the time required for integrating the element matrices and for solving the system of equations.

In the run-time analysis presented in Section 4.1, however, these possible disadvantages do *not* materialise. Instead, the multi-level *hp*-method yields exponential convergence characteristics in the run-time. This suggests that these negative effects—although present—are more of theoretical nature and have only a very limited relevance in practice.

Journal publication: https://doi.org/10.1007/s00466-014-1118-x

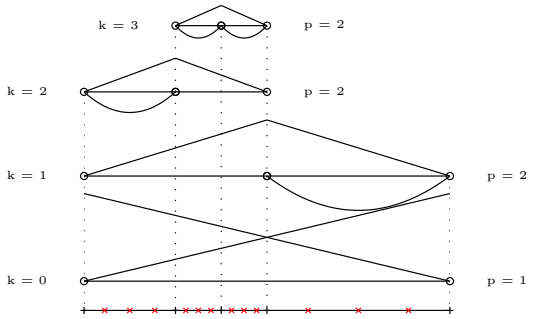


Figure 7: Integration of a multi-level element

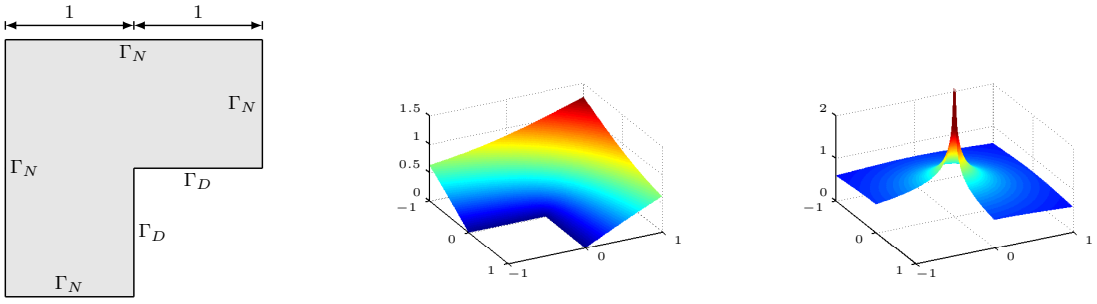
4 Numerical Examples

In the following section, the performance of the multi-level *hp*-approach described in Sections 2 and 3 is demonstrated. The convergence properties for applications with non-smooth solution characteristics are addressed by the first benchmark. The second and third examples demonstrate that the agile data structure of the method can be used to adaptively refine and coarsen the discretization during transient simulations.

4.1 Non-Smooth Solution Benchmark

The aim of this first example is to analyze the convergence properties of the proposed refinement methods in the context of non-smooth problems. For this purpose, the well known L-shaped domain problem is considered (see e.g. [56]).

4.1.1 Problem definition



(a) L shaped domain

(b) Analytical Temperature Solution ϕ

(c) Flux Magnitude $q = ||\nabla\phi||$

Figure 8: Setup and solution of L-shaped domain problem

Consider the temperature problem

$$\Delta\phi = 0 \qquad \forall\,(r,\theta) \in \Omega$$

Journal publication: https://doi.org/10.1007/s00466-014-1118-x

defined on the L-shaped domain depicted in Figure 8a, with the boundary conditions defined as

$$\begin{aligned} \phi &= 0 && \forall\,(r,\theta) \in \Gamma_D \\ \nabla\phi \cdot n &= \frac{2}{3} r^{-\frac{4}{3}} \begin{bmatrix} x\sin\left(\frac{2}{3}\theta\right) - y\cos\left(\frac{2}{3}\theta\right) \\ y\sin\left(\frac{2}{3}\theta\right) + x\cos\left(\frac{2}{3}\theta\right) \end{bmatrix} \cdot n && \forall\,(r,\theta) \in \Gamma_N. \end{aligned}$$

Therein, Γ_D and Γ_N define the Dirichlet and Neumann boundaries, respectively, such that

$$\Gamma_D \cup \Gamma_N = \partial\Omega \quad \text{and} \quad \Gamma_D \cap \Gamma_N = \emptyset.$$

As shown in e.g. [1], the analytical temperature distribution ϕ can be given in polar coordinates r and θ as

$$\begin{aligned} &\phi(r,\theta) = r^{\lambda} \sin\left(\frac{2}{3}\theta\right) && \forall\,(r,\theta) \in \Omega, \\ &\text{with } \lambda = 2/3. \end{aligned}$$

4.1.2 Numerical Results

To approximate the analytical temperature distribution ϕ numerically, the depicted domain is discretized with an initial element size $h = 1/2$. This base mesh is then refined using the hierarchical *hp-d-* and the multi-level *hp*-approaches. In both cases, no solution-based refinement indicator is used. Instead, the refinement is steered geometrically towards the re-entrant corner by always refining the inner-most element.

The numerical approximation of the analytical heat flux magnitude $q = ||\nabla\phi||$ obtained with the *hp-d-* and multi-level *hp*-approaches are depicted in Figure 9. The comparison to the analytical solution in Figure 8 shows that the overall solution characteristics are captured correctly by both strategies. In the case of the *hp-d*-method, however, the elements near the singularity suffer from severe discontinuities in the flux, which is directly associated with the discretization error (see e.g. [57]). Increasing the number of recursive refinements reduces this defect, as the effects of the singularity are bound to the higher refinement levels. But even with ten recursive refinements, the numerical solution still shows large discontinuities in the flux field. In contrast, the results obtained by the multi-level *hp*-approach approximate the singularity significantly better with only minor discontinuities in the derivative field, which qualitatively shows the advantage of this new approach.

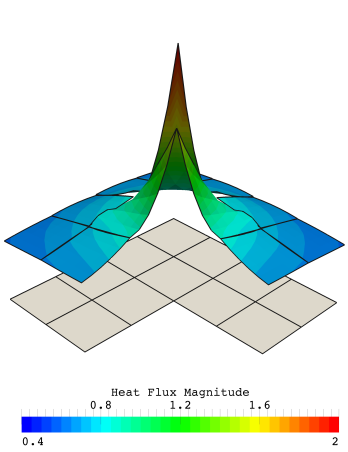
To quantify the quality of the obtained results in more detail, the convergence characteristics are analyzed. For this purpose, the function space is enlarged by increasing the polynomial order of the employed shape functions while keeping the element size unchanged. In the context of an *hp-d-*refinement, this only effects the order of the base mesh whereas the overlay meshes remain linear. In the case of multi-level *hp*, however, the *p*-refinement leads to an increase of the polynomial ansatz order on all overlay levels.

Figure 10 depicts the reduction of the relative discretizaton error in the energy norm

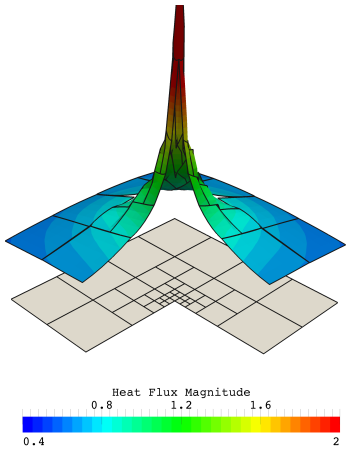
$$||e||_E = \sqrt{\frac{|\,\Pi_{ex} - \Pi_{fe}\,|}{\Pi_{ex}}} \cdot 100\%$$

versus the number of degrees of freedom (dofs), with Π_{ex} and Π_{fe} denoting the exact and the approximated energy, respectively. As discussed e.g. in [56; 58], *algebraic* convergence with a rate $\beta = \lambda = \frac{2}{3}$ is to be expected when *p*-refinement is applied to the singular problem under consideration. As shown in Figure 10a, these theoretical values are met when no refinement is employed (i.e. 0 refinements). If the discretization is refined following the *hp-d*-idea, the first two to three *p*-refinement

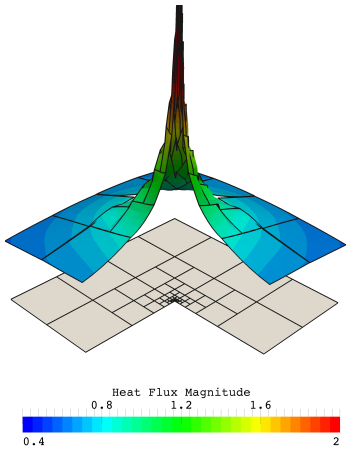
Journal publication: https://doi.org/10.1007/s00466-014-1118-x



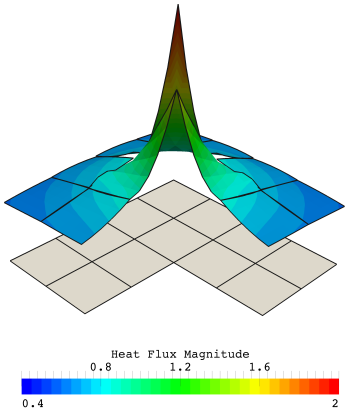
(a) Approximation of q using no hp-d-refinement



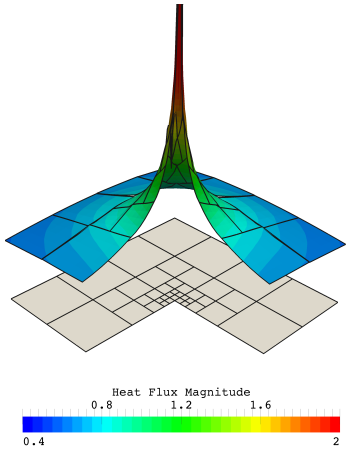
(b) Approximation of q using 3 hp-d-refinements



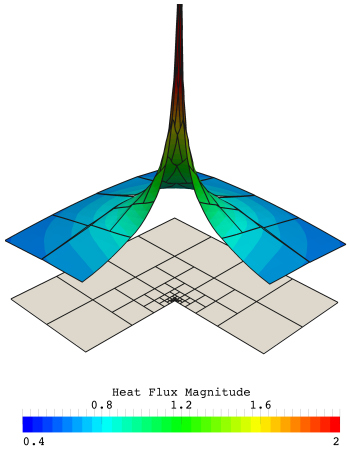
(c) Approximation of q using 10 hp-d-refinements



(d) Approximation of q using no multi-level hp-refinements



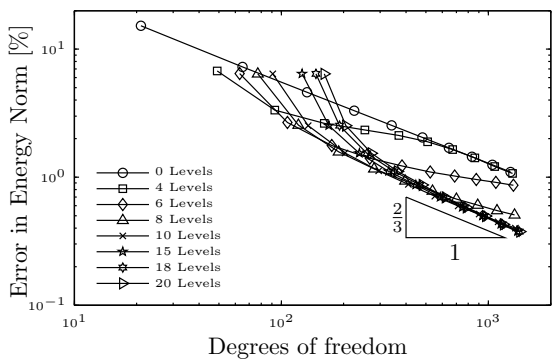
(e) Approximation of q using 3 multi-level hp-refinements



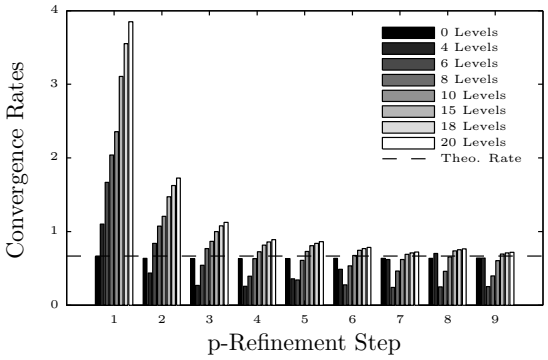
(f) Approximation of q using 10 multi-level hp-refinements

Figure 9: Numerical approximation of the flux q obtained with $p = 2$

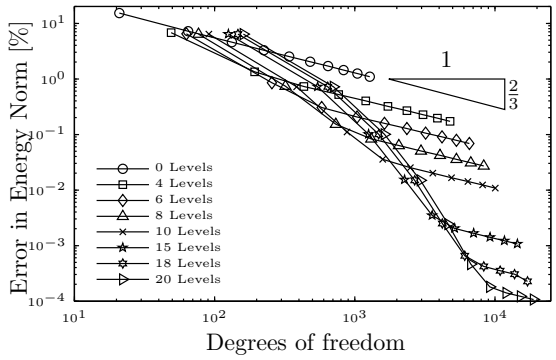
Journal publication: https://doi.org/10.1007/s00466-014-1118-x



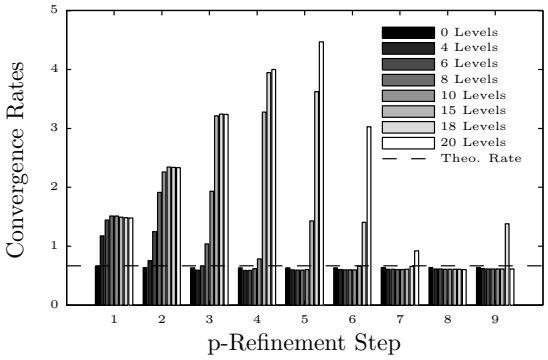
(a) Convergence using *hp-d*-refinements



(b) Convergence rates β using *hp-d*-refinements

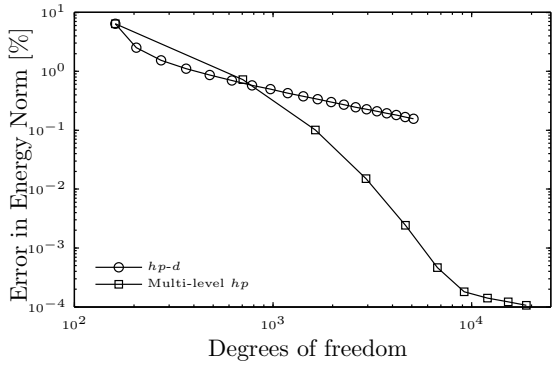


(c) Convergence using multi-level *hp*-refinements

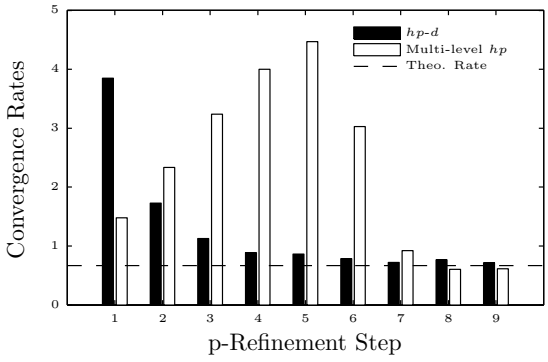


(d) Convergence rates β using multi-level *hp*

Figure 10: Convergence results of p-refinement using $p = 1, 2, \ldots, 10$



(a) Convergence curves



(b) Convergence rates

Figure 11: Direct comparison of convergence results obtained using *hp-d*- and multi-level *hp*-approach (20 refinements)

Journal publication: https://doi.org/10.1007/s00466-014-1118-x

steps yield a significant improvement of the numerical error. For the remaining steps, however, the convergence rates significantly reduce towards the theoretical value of $\beta = 2/3$ and asymptotically, an algebraic convergence is observed (see Figure 10b). In particular, neither the absolute error value nor the convergence characteristics are affected significantly by the employed number of mesh refinements. These results confirm the findings of e.g. [17].

The results presented in Figure 10c demonstrate that using multi-level *hp*-refinement changes the convergence characteristics substantially. Although the asymptotic convergence is still algebraic, the convergence rates continuously increase for the first refinement steps if the mesh grading is strong enough. The convergence can, therefore, be identified as exponential in the pre-asymptotic range. Only when the polynomial order of the shape functions exceeds the optimal value for the current mesh, the convergence rates decrease and tend towards the expected algebraic value of $\beta = 2/3$ in the asymptotic range. In contrast to the previously presented results, however, the comparison of different refinement levels demonstrates that this transition point can be shifted significantly to higher accuracy if more refinement levels are employed. These results thus show that pre-asymptotic exponential convergence for piecewise analytical functions cannot be achieved exclusively with geometrically graded meshes (see e.g. [56]). The direct comparison between the two methods, depicted in Figure 11, demonstrates that this new approach improves the approximation accuracy by several orders of magnitude in the energy norm.

As discussed in Section 3.4, the superposition approach results in a larger support of the shape functions than a conventional *hp*-approach. This leads to larger element matrices and a denser system of equations with an increased bandwidth. Therefore, this section concludes with a run-time analysis of the *hp-d-* and the multi-level *hp*-method.

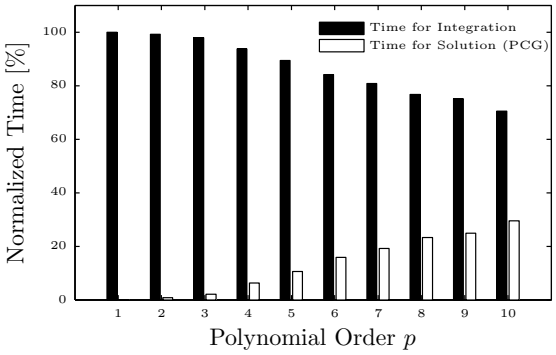


Figure 12: Comparison of integration and solution time of the multi-level *hp*-method (20 refinements)

It is well known that using higher-order shape functions, very high accuracy can be achieved with a comparable small number of unknowns. At the same time, however, the number of shape functions per element significantly increases. This raises the time required for integrating the element matrices, such that it typically dominates the time for solving the final system of equations. To evaluate this effect, the time required for integrating and the time required for solving the system of equations are compared in Figure 12. Thereby, a straightforward, non-specialized conjugate gradient solver with a simple diagonal pre-conditioning was used for solving the system of equations iteratively up to a residual error of 10^{-15} (see e.g. [59]).

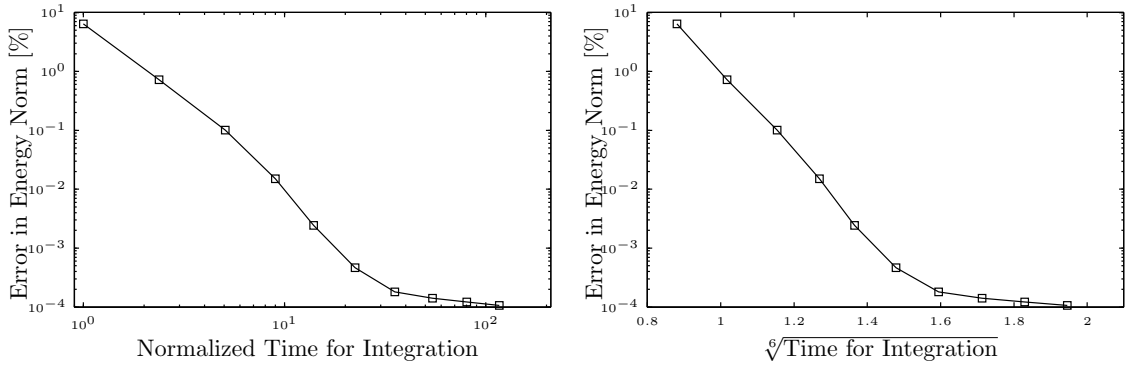
The results show that even when using such non-optimized solver, the time for solving the system of equations is not dominating. This applies in particular for the pre-asymptotic range ($p = 1 \dots 6$), in which the time for solving does not exceed 20%. As the increase in accuracy is largest in this range, it can, therefore, be deduced that the increase of the bandwidth and the reduction of the

Journal publication: https://doi.org/10.1007/s00466-014-1118-x

sparsity of the final equation system—although present—do not have a significant influence on the total computation time.

The more relevant question is thus whether the increase in integration time is in balance with the gain in accuracy. To answer this question, Figure 13a plots the discretization error in the energy norm versus the required time for integrating and assembling the element matrices in a double-logarithmic scale.

As in the previous studies, this plot shows that the pre-asymptotic and the asymptotic range can be clearly identified also in terms of the integration time. Within the pre-asymptotic range, the time for integration increases by a factor of approximately 35. In the same range, however, the error reduces by a factor of about 35,000. It is further interesting to note that by doubling the integration time, the relative error is reduced by a factor of ten, which undercuts the engineering accuracy of 1%.



(a) Double-logarithmic scaling

(b) Semi-logarithmic scaling

Figure 13: Convergence of the discretization error depending on the time required for integrating the element stiffness matrix (20 mulit-level *hp*-refinements, $p = 1 \dots 10$)

These numbers already indicate a significant performance gain achieved by the high-order overlay meshes. The results, however, do not allow to judge whether the increase of support described in Section 3.4 has a negative influence on the method's run-time complexity. For this purpose, the following simple complexity model shall be used: For a given discretization, the time t_i for integrating the element matrices depends on the number of integration points n_i and the time t_m required for computing the matrix product on the integration points. Using standard two-dimensional Gaussian quadrature, the number of integration points n_i depends on the polynomial order p of the shape functions as follows:

$$n_i = (p+1)^2.$$

Furthermore, the time t_m required for computing the matrix product on each integration point is

$$t_m = \alpha \cdot n_e^2,$$

with n_e denoting the number of unknowns per element and α being a positive constant. Furthermore, it is known that using the full tensor space (see e.g. [56; 58]), the number of unknowns can be expressed in terms of the polynomial degree as follows:

$$n_e = (p+1)^2,$$

Journal publication: https://doi.org/10.1007/s00466-014-1118-x

which equals the number of integration points n_i. As the matrix product has to be computed for every integration point, the total time required for integration can thus be estimated as

$$t_i = \alpha \cdot n_i \cdot n_e^2 = \alpha \cdot n_e^3,$$

thereby neglecting any overhead such as e.g. memory allocation. Inverting the above expression and, furthermore, assuming that the total number of unknowns n scales linearly with the number of unknowns per element n_e allows to express these as follows:

$$n \propto t_i^{\frac{1}{3}}.$$

The results in Figure 11 show that, in the pre-asymptotic range, the discretization error decays exponentially with the number of unknowns. Following e.g. [56; 58], the convergence of the error can thus be written as:

$$||e||_E \leq k \cdot e^{-\gamma \cdot n^{\theta}},$$

with k and γ being positive constants. According to e.g. [56; 58], it is assumed that $\theta = \frac{1}{2}$ for the two-dimensional problem under consideration. Using these observations, the decay of the error can be expressed depending on the integration time in the following way:

$$||e||_E \leq k \cdot e^{-\hat{\gamma} \cdot t_i^{\frac{1}{6}}},$$

with $\hat{\gamma}$ being a positive constant.

These results allow to judge on the complexity of the implemented multi-level *hp*-algorithm by depicting the discretization error logarithmically against the sixth root of the run-time. As shown in Figure 13b, a linear relation between these entities can be observed for the values of $p = 1 \ldots 6$.

This demonstrates that the multi-level *hp*-method yields the theoretical exponential convergence characteristics for non-smooth problems not only in terms of degrees of freedom but also in terms of the run-time. Together with the previously discussed results, this shows that the increase of span has no considerable deteriorating influence on the run-time complexity.

4.2 Muli-Level *hp*-Adaptivity for Moving Objects on Fixed Meshes

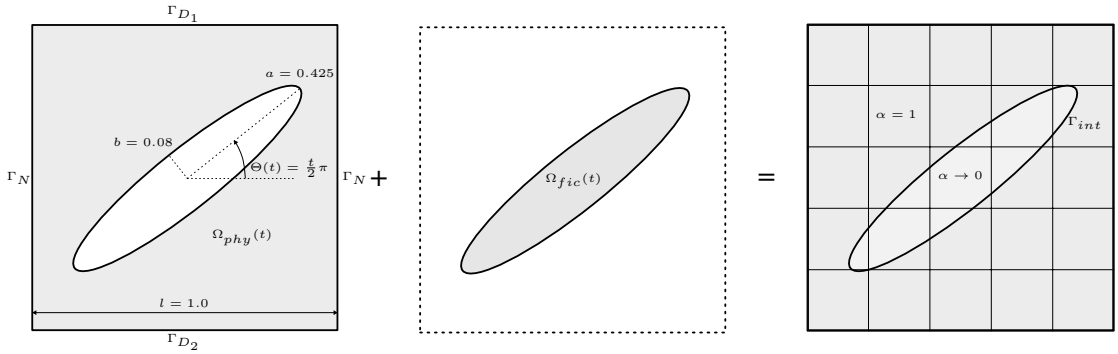


Figure 14: Setup of rotating ellipse example in the context of the Finite Cell Method

The aim of this second example is to combine the *hp-d-* and multi-level *hp*-refinement algorithms with the idea of high-order fictitious domain methods to capture the temperature distribution around

Journal publication: https://doi.org/10.1007/s00466-014-1118-x

moving objects on a fixed background mesh. For this purpose, the schemes are employed to simulate the heat flux around the rotating ellipse depicted in Figure 14 with the model problem defined as follows:

$$\begin{aligned} \frac{\partial \phi}{\partial t} &= \Delta \phi \qquad && \forall\, x \in \Omega \text{ and } t \in [0,2] \\ \phi &= 1 && \forall\, x \in \Gamma_{D_1} \text{ and } t \in [0,2] \\ \phi &= 0 && \forall\, x \in \Gamma_{D_2} \text{ and } t \in [0,2] \\ \nabla\phi \cdot n &= 0 && \forall\, x \in \Gamma_N \text{ and } t \in [0,2] \\ \phi &= 0 && \forall\, x \in \Omega \text{ and } t = 0. \end{aligned}$$

For the temporal discretization, a simple backward Euler approach with 32 equidistant time steps is used. To spatially discretize the domain, the Finite Cell Method (FCM) is used (see e.g. [60; 61]). The essential idea of this high-order fictitious domain approach is to embed the actual physical domain Ω_{phy} in a fictitious domain Ω_{fic}, such that their union can be discretized easily (see Figure 14). A scalar field

$$\alpha(x) = \begin{cases} 1 & \forall\, x \in \Omega_{phy} \\ \varepsilon & \forall\, x \notin \Omega_{phy} \end{cases}$$

is used to recover the original problem, with $\varepsilon \to 0$. This simple idea allows to circumvent the mesh generation step. Recent publications show that this approach can be applied successfully in the context of linear elasticity [60; 61; 62], topology optimization [63], geometrically nonlinear continuum mechanics [16; 17; 64], computational steering [65; 66], biomedical engineering [67], elastoplasticity [68; 69], wave propagation in heterogeneous materials [70], local enrichment for material interfaces [19], convection diffusion problems [71], thin-walled structures [72], design-through analysis and isogeometric analysis [15; 16; 18; 64; 73; 74; 75; 76], and multiphysics applications [77]. Recently, the convergence properties of the method have been analyzed in detail by [78]. A comprehensive review of the method and its recent extensions is given in [79].

In their recent works, Schillinger *et al.* showed that the Finite Cell Method can also be combined with *hp-d*-refinement schemes to significantly increase the approximation accuracy (see e.g. [15; 16; 17; 18]). The studies, however, concentrated on examples in which the geometry of the physical domain remained unchanged over time. There was, therefore, no need to update the discretization between the time steps. In the examples under consideration, however, the ellipse is rotating within the heat field. An fixed *a priori* adaption is therefore not applicable in this case.

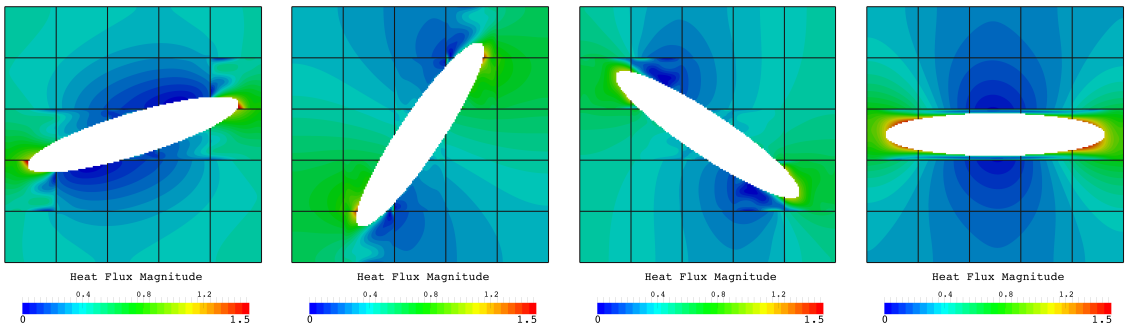
As shown in Figure 15, this reveals some challenges associated with the FCM. Although the physical behaviour is captured well qualitatively by the simulation, the numerical results suffer from oscillations. The reason for this problem is that the shape functions of elements incorporating the interface Γ_{int} cannot capture the solution characteristics correctly within one finite cell. To reduce the oscillations, the solution space has, therefore, to be increased around this internal interface.

The discussed *hp-d-* and multi-level *hp*-methods are well suited for this purpose as the cut cells can be refined adaptively as soon as Γ_{int} enters the element, and coarsend again when Γ_{int} leaves the element domain. As in the example in Section 4.1, the refinement is thus steered geometrically.

As shown in Figure 16, the use of the *hp-d*-refinement method significantly improves the results. But minor oscillations are still present when only two levels of refinements are used. The reason for this seems to be that the oscillations are "broken" at the element edges only and, therefore, restricted to the finest mesh level. As the discretization is still comparably coarse when applying only two refinements steps, oscillations can still be observed. Increasing the number of overlay meshes further improves the results (see Figure 17).

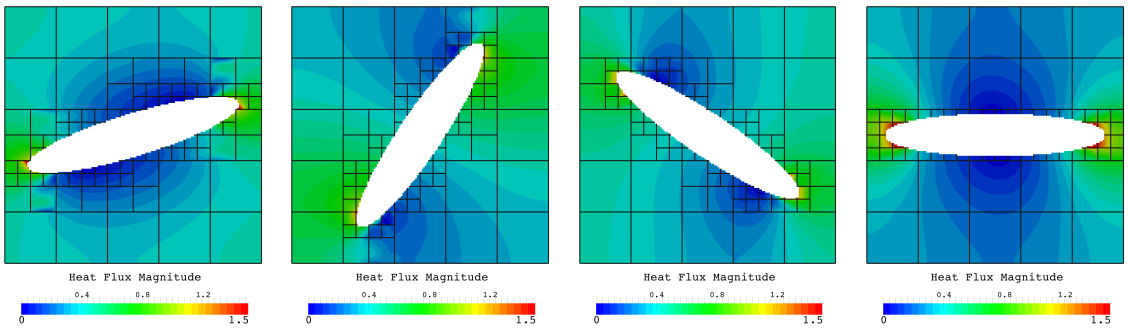
When using the new multi-level *hp*-method instead, only two levels of refinement suffice to circum-

Journal publication: https://doi.org/10.1007/s00466-014-1118-x



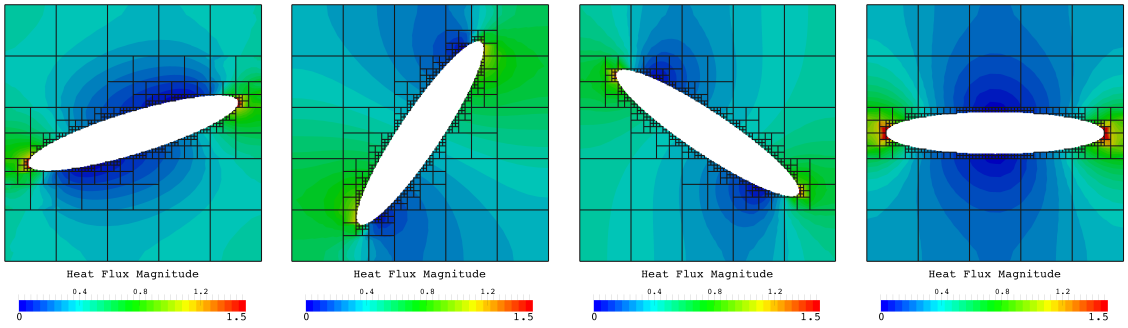
(a) $t = 10\%$ (b) $t = 30\%$ (c) $t = 80\%$ (d) $t = 100\%$

Figure 15: Heat flow around the rotating ellipse without using any refinement with $p = 4$



(a) $t = 10\%$ (b) $t = 30\%$ (c) $t = 80\%$ (d) $t = 100\%$

Figure 16: Heat flow around the rotating ellipse with $p = 4$ and 2 levels of *hp-d*-refinements

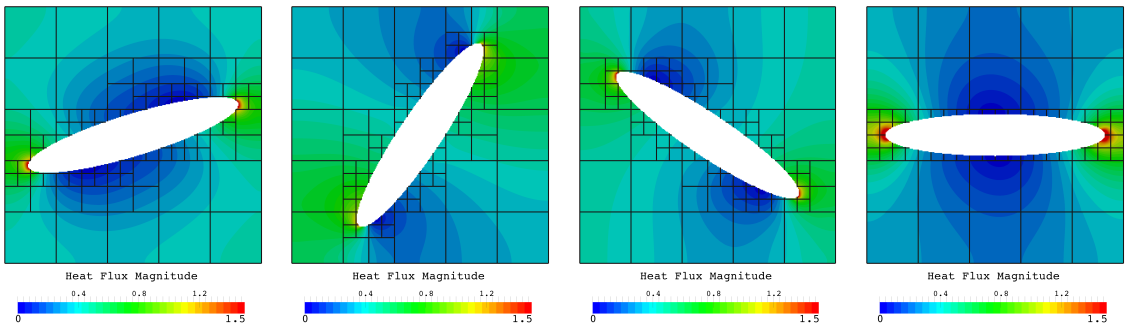


(a) $t = 10\%$ (b) $t = 30\%$ (c) $t = 80\%$ (d) $t = 100\%$

Figure 17: Heat flow around the rotating ellipse with $p = 4$ and 4 levels of *hp-d*-refinements

Journal publication: https://doi.org/10.1007/s00466-014-1118-x

vent artificial oscillations (see Figure 18). This can be explained by the high-order shape functions on the overlay meshes, which allow to capture the complex solution at the internal interface more accurately.



(a) $t = 10\%$ (b) $t = 30\%$ (c) $t = 80\%$ (d) $t = 100\%$

Figure 18: Heat flow around the rotating ellipse with $p = 4$ and 2 multi-level hp-refinements

This comparison shows that the use of the multi-level hp-method significantly improves the result quality when simulating the heat flow around moving objects on fixed background meshes.

4.3 Automatic Multi-Level hp-Refinement and Coarsening

This example demonstrates the applicability of the multi-level hp-refinement method to hyperbolic problems. In particular, the ability of the method to robustly refine and coarsen the discretization *automatically* is of interest. For this purpose, the refinement strategy is combined with a simple energy-based refinement indicator.

4.3.1 Problem definition

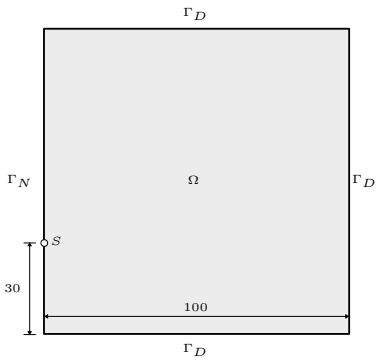


Figure 19: Setup of compression wave example

Journal publication: https://doi.org/10.1007/s00466-014-1118-x

To simulate the evolution of an acoustic wave, the well-known wave equation (see e.g. [80]) is used:

$$\begin{aligned}
\frac{\partial^2 \phi}{\partial t^2} &= \Delta\phi + s(x,t) && \forall\, x \in \Omega \text{ and } t \in [0, 100] \\
\phi &= 0 && \forall\, x \in \Gamma_D \text{ and } t \in [0, 100] \\
\nabla\phi \cdot n &= 0 && \forall\, x \in \Gamma_N \text{ and } t \in [0, 100] \\
\phi &= 0 && \forall\, x \in \Omega \text{ and } t = 0 \\
\frac{\partial \phi}{\partial t} &= 0 && \forall\, x \in \Omega \text{ and } t = 0,
\end{aligned}$$

with Ω being depicted in Figure 19. Centered on the point S, the following wavelet-shaped source term is applied to emit a wave front:

$$\begin{aligned}
s(x,t) &= a_t(t) \cdot a_x(x) \\
\text{with } a_x(x) &= A_x \cdot e^{-\frac{(x-S)^2}{2\sigma_x^2}} \\
a_t(t) &= -A_t \cdot (t-\mu) e^{-\frac{(t-\mu)^2}{2\sigma_t^2}} \\
\text{and } A_x = 100,\ A_t &= 1000,\ \sigma_x = 6,\ \sigma_t = 1,\ \mu = 10
\end{aligned}$$

4.3.2 Numerical Results

The time range is discretized in 200 equidistant time steps using the well-known Newmark time stepping scheme (see e.g. [24; 23]). The spatial domain is discretized with 10×10 elements on which shape functions with $p = 4$ are defined. During the simulation, this base mesh is superposed with a maximum of three levels of overlay meshes with $p = 4$. The decision on where to employ the refinement is based on the relative energy contribution of the respective element i in comparison to the total energy of the current solution:

$$\frac{||u||_E^i}{||u||_E^{tot}} > tol,$$

with *tol* being a user defined tolerance value.

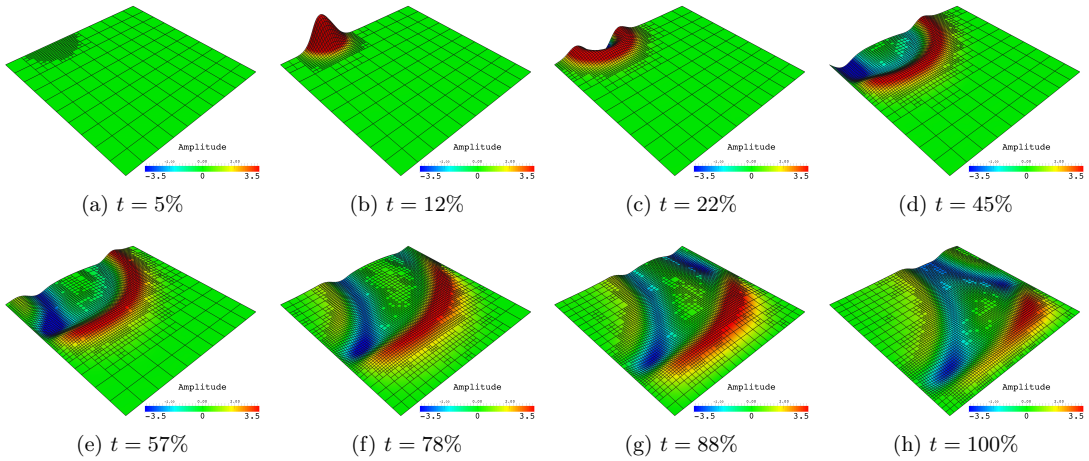
Figure 20 depicts the results of this simulation, with the mesh being warped in the z-direction according to the current wave amplitude. The time series clearly shows how the discretization accuracy is following the wave front robustly. In particular, two distinct regions of refinement can be seen after the wave is reflected from the lower boundary. In-between these compression and expansion waves, the mesh is coarsened again as here the gradient is significantly smaller.

These results demonstrate that even with this simple solution-based refinement criterion, the multi-level *hp*-refinement can be controlled automatically, without any difficulties of hanging nodes or mesh irregularities.

5 Conclusion and Outlook

In the present work, a novel multi-level *hp*-refinement scheme was introduced, which allows to alleviate the challenges associated with the implementation of the compatibility enforcement for arbitrary hanging nodes in classical *hp*-schemes. Extending the idea of *hp-d*-methods, this new approach refines a coarse base mesh by *hierarchically* superposing it with finer, *high-order* overlay meshes on which small-scale solution characteristics can be captured with high accuracy. This multi-level *hp*-formulation of the Finite Element Method avoids the difficulties associated with hanging nodes as

Journal publication: https://doi.org/10.1007/s00466-014-1118-x



(a) $t = 5\%$ (b) $t = 12\%$ (c) $t = 22\%$ (d) $t = 45\%$

(e) $t = 57\%$ (f) $t = 78\%$ (g) $t = 88\%$ (h) $t = 100\%$

Figure 20: Propagation of wave front and discretization over time (max. three overlay meshes, $p = 4$)

the compatibility requirements are ensured by construction while yielding the same capabilities as conventional *hp*-approaches.

The different examples discussed in this work demonstrated that this new approach is well suited for problems with non-smooth solution characteristics and can be combined easily with the idea of fictitious domain methods and automatically controlled refinement procedures. In particular, it could be shown that the introduced method achieves comparable approximation properties as conventional *hp*-schemes and works robustly in the context of elliptic, parabolic, and hyperbolic applications.

In the context of elliptic problems, this could be demonstrated by applying the multi-level *hp*-approach to approximate the singular solution of the well-known L-shaped domain benchmark. The presented results showed that using high-order overlay meshes for refinement significantly improves the simulation accuracy compared to linear overlay meshes in the *hp-d*-approach. In particular, the obtained convergence characteristics are exponential in the pre-asymptotic range.

The second example demonstrated that the agile data structure of the method allows to robustly refine and coarsen the discretization in every time step of a parabolic simulation. In particular, it could be shown that, in combination with the idea of fictitious domain approaches (Finite Cell Method), the multi-level *hp*-algorithm allows to capture the heat flow around a moving object on a fixed background mesh.

In the third example, the multi-level *hp*-scheme was applied in the context of hyperbolic problems to capture a propagating wave front more accurately. The presented results show that the refinement approach can also be combined with a solution-based refinement criterion which allows for *automatic* mesh adaptivity.

These three different examples demonstrate the potential of the multi-level *hp*-approach introduced in this work. However, further research is necessary to analyze the method in more detail and to extend its application range.

In a first step, the authors plan to extend the multi-level *hp*-method to three dimensions. In the context of the increasing problem size, it will be interesting to investigate whether the hierarchical structure of the shape functions can be utilized to improve the efficiency of the equation solver. A first idea is to apply a multi-grid approach by solving for the respective error components on the different mesh levels. Similar approaches have been applied successfully in the context of *hp*-FEM by e.g. [2; 81; 82]. An alternative idea is to exploit the local nature of the overlay shape functions. As these modes are not coupled directly to adjacent coarse elements, the efficiency of the solution

Journal publication: https://doi.org/10.1007/s00466-014-1118-x

procedure can be improved by a suitable condensation of their associated degrees of freedom (see e.g. [83; 2]).

A second interesting research direction is to further improve the method's convergence speed. Numerical studies show that high-order modes do not contribute significantly to the accuracy in the direct vicinity of a singularity. This observation is utilized in *graded mesh* approaches by decreasing the polynomial degree of the shape functions towards the singularity (see e.g. [84; 56]). The same idea can also be applied in the context of multi-level *hp*-refinement by decreasing the polynomial degree of the shape functions gradually on higher mesh levels (see Figure 21). As a similar idea proposed in [85] has shown good results for one-dimensional applications, the authors expect that this will further improve the rates of convergence.

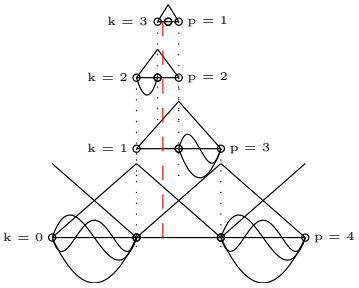


Figure 21: Conceptional idea of graded multi-level *hp*-FEM ($p = 4$, k: refinement level, dashed line: singular point)

A third possible research direction is the formulation of an improved error-based refinement indicator. Additionally, the aspect of smoothness estimation has to be investigated to decide on whether an *h*- or *p*-refinement has a greater effect on the simulation accuracy. The authors conjecture that this will then allow to also achieve exponential convergence for singular problems in the asymptotic sense (see e.g. [1; 2; 56; 58; 4]).

Acknowledgements

The first and the last author gratefully acknowledge the financial support of the German Research Foundation (DFG) under Grants RA 624/19-2 and RA 624/22-1.

References

[1] L. Demkowicz, *Computing with hp-adaptive finite elements, Vol. 1: One and Two Dimensional Elliptic and Maxwell Problems*. Applied mathematics and nonlinear science series, Boca Raton: Chapman & Hall/CRC, 2007.

[2] P. Solin, *Higher-order finite element methods*. Studies in advanced mathematics, Boca Raton, FL: Chapman & Hall/CRC, 2004.

[3] L. Demkowicz, J. T. Oden, W. Rachowicz, and O. Hardy, "Toward a universal h-p adaptive finite element strategy, part 1. constrained approximation and data structure," *Computer Methods in Applied Mechanics and Engineering*, vol. 77, no. 1–2, pp. 79–112, 1989.

Journal publication: https://doi.org/10.1007/s00466-014-1118-x

[4] W. Rachowicz, J. T. Oden, and L. Demkowicz, "Toward a universal h-p adaptive finite element strategy part 3. design of h-p meshes," *Computer Methods in Applied Mechanics and Engineering*, vol. 77, no. 1–2, pp. 181–212, 1989.

[5] L. Demkowicz, K. Gerdes, C. Schwab, A. Bajer, and T. Walsh, "HP90: A general and flexible fortran 90 hp-FE code," *Computing and Visualization in Science*, vol. 1, no. 3, pp. 145–163, 1998.

[6] L. Demkowicz, A. Bajer, W. Rachowicz, and K. Gerdes, "3d hp-adaptive finite element package fortran 90 implementation (3dhp90)," TICAM Report 99-29, The University of Texas at Austin, Texas Institute for Computational and Applied Mathematics, 1999.

[7] W. Rachowicz and L. Demkowicz, "An hp-adaptive finite element method for electromagnetics: Part 1: Data structure and constrained approximation," *Computer Methods in Applied Mechanics and Engineering*, vol. 187, no. 1–2, pp. 307–335, 2000.

[8] W. Rachowicz and L. Demkowicz, "An hp-adaptive finite element method for electromagnetics—part II: A 3d implementation," *International journal for numerical methods in engineering*, vol. 53, no. 1, pp. 147–180, 2002.

[9] M. Paszyński and L. Demkowicz, "Parallel, fully automatic hp-adaptive 3d finite element package," *Engineering with Computers*, vol. 22, no. 3-4, pp. 255–276, 2006.

[10] P. Solin and J. Cerveny, "Automatic hp-adaptivity with arbitrary-level hanging nodes," Tech. Rep. Research Report No. 2006-07, The University of Texas at El Paso, Department of Mathematical Sciences, 2006.

[11] P. Solin, J. Cerveny, and I. Dolezel, "Arbitrary-level hanging nodes and automatic adaptivity in the hp-FEM," *Mathematics and Computers in Simulation*, vol. 77, no. 1, pp. 117–132, 2008.

[12] P. Kus, *Automatic hp-Adaptivity on Meshes with Arbitrary-Level Hanging Nodes in 3D*. Phd thesis, Charles University, Institute of Mathematics, Prague, 2011.

[13] A. Schröder, "Constrained approximation in hp-FEM: Unsymmetric subdivisions and multi-level hanging nodes," in *Spectral and High Order Methods for Partial Differential Equations* (J. S. Hesthaven and E. M. Ronquist, eds.), no. 76 in Lecture Notes in Computational Science and Engineering, pp. 317–325, Springer Berlin Heidelberg, 2011.

[14] E. Rank, "Adaptive remeshing and h-p domain decomposition," *Computer Methods in Applied Mechanics and Engineering*, vol. 101, no. 1–3, pp. 299–313, 1992.

[15] 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*, vol. 200, no. 47-48, pp. 3358–3380, 2011.

[16] 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*. Doctoral thesis, Technische Universität München, Chair for Computation in Engineering, 2012.

[17] 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*, vol. 89, no. 9, pp. 1171–1202, 2012.

Journal publication: https://doi.org/10.1007/s00466-014-1118-x

[18] D. Schillinger, L. Dedè, M. A. Scott, J. A. Evans, M. J. Borden, E. Rank, and T. J. 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*, vol. 249-252, pp. 116–150, 2012.

[19] M. Joulaian and A. Düster, "Local enrichment of the finite cell method for problems with material interfaces," *Computational Mechanics*, vol. 52, no. 4, pp. 741–762, 2013.

[20] M. Ainsworth and B. Senior, "Aspects of an adaptive hp-finite element method: Adaptive strategy, conforming approximation and efficient solvers," *Computer Methods in Applied Mechanics and Engineering*, vol. 150, no. 1–4, pp. 65–87, 1997.

[21] O. Zienkiewicz, R. Taylor, and J. Zhu, *The Finite Element Method: Its Basis and Fundamentals*. Butterworth-Heinemann, 6 ed., 2005.

[22] D. C. Franke, A. Düster, V. Nübel, and E. Rank, "A comparison of the h-, p-, hp-, and rp-version of the FEM for the solution of the 2d hertzian contact problem," *Computational Mechanics*, vol. 45, no. 5, pp. 513–522, 2010.

[23] T. J. R. Hughes, *The finite element method: linear static and dynamic finite element analysis*. Mineola, NY: Dover Publications, 2000.

[24] K. J. Bathe, *Finite element procedures*. Prentice Hall, 2007.

[25] G. Strang, *An analysis of the finite element method*. Englewood Cliffs, N.J: Prentice-Hall, 1973.

[26] C. A. Felippa, "Introduction to finite element methods." 2013.

[27] M. Rivara, "Mesh refinement processes based on the generalized bisection of simplices," *SIAM Journal on Numerical Analysis*, vol. 21, no. 3, pp. 604–613, 1984.

[28] W. F. Mitchell, "A comparison of adaptive refinement techniques for elliptic problems," *ACM Trans. Math. Softw.*, vol. 15, no. 4, pp. 326–347, 1989.

[29] I. Babuška and A. Aziz, "On the angle condition in the finite element method," *SIAM Journal on Numerical Analysis*, vol. 13, no. 2, pp. 214–226, 1976.

[30] P. Solin, L. Dubcova, and I. Dolezel, "Adaptive hp-FEM with arbitrary-level hanging nodes for maxwell's equations," *Adv. Appl. Math. Mech*, vol. 2, no. 4, pp. 518–532, 2010.

[31] R. Schneiders, "Algorithms for quadrilateral and hexahedral mesh generation," *Proceedings of the VKI Lecture Series on Computational Fluid Dynamic, VKI-LS*, vol. 4, 2000.

[32] R. Niekamp and E. Stein, "An object-oriented approach for parallel two- and three-dimensional adaptive finite element computations," *Computers & Structures*, vol. 80, no. 3–4, pp. 317–328, 2002.

[33] T.-P. Fries, A. Byfut, A. Alizada, K. W. Cheng, and A. Schröder, "Hanging nodes and XFEM," *International Journal for Numerical Methods in Engineering*, vol. 86, no. 4-5, pp. 404–430, 2011.

[34] K.-W. Cheng and T.-P. Fries, "XFEM with hanging nodes for two-phase incompressible flow," *Computer Methods in Applied Mechanics and Engineering*, vol. 245-246, pp. 290–312, 2012.

[35] A. Szymczak, A. Paszyńska, M. Paszyński, and D. Pardo, "Preventing deadlock during anisotropic 2d mesh adaptation in hp-adaptive FEM," *Journal of Computational Science*, vol. 4, no. 3, pp. 170–179, 2013.

Journal publication: https://doi.org/10.1007/s00466-014-1118-x

[36] C. D. Mote, "Global-local finite element," *International Journal for Numerical Methods in Engineering*, vol. 3, no. 4, pp. 565–574, 1971.

[37] A. K. Noor, "Global-local methodologies and their application to nonlinear analysis," *Finite Elements in Analysis and Design*, vol. 2, no. 4, pp. 333–346, 1986.

[38] O. C. Zienkiewicz and A. Craig, "Adaptive refinement, error estimates, multigrid solution and hierarchic finite element method concepts," in *Accuracy Estimates and Adaptive Refinements in Finite Element Calculations* (I. Babuska, O. C. Zienkiewicz, J. Gago, and E. R. de Oliviera, eds.), pp. 25–55, New York: Wiley, 1986.

[39] T. Belytschko, J. Fish, and B. E. Engelmann, "A finite element with embedded localization zones," *Computer Methods in Applied Mechanics and Engineering*, vol. 70, no. 1, pp. 59–89, 1988.

[40] J. Fish and T. Belytschko, "Elements with embedded localization zones for large deformation problems," *Computers & Structures*, vol. 30, no. 1–2, pp. 247–256, 1988.

[41] J. Fish and T. Belytschko, "A finite element with a unidirectionally enriched strain field for localization analysis," *Computer Methods in Applied Mechanics and Engineering*, vol. 78, no. 2, pp. 181–200, 1990.

[42] T. Belytschko, J. Fish, and A. Bayliss, "The spectral overlay on finite elements for problems with high gradients," *Computer Methods in Applied Mechanics and Engineering*, vol. 81, no. 1, pp. 71–89, 1990.

[43] E. Rank and R. Krause, "A multiscale finite-element method," *Computers & structures*, vol. 64, no. 1, pp. 139–144, 1997.

[44] J. Fish, "The s-version of the finite element method," *Computers & Structures*, vol. 43, no. 3, pp. 539–547, 1992.

[45] Y. H. Kim, I. Levit, and G. Stanley, "A finite element adaptive mesh refinement technique that avoids multipoint constraints and transition zones," in *Iterative Equation Solvers for Structural Mechanics Problems* (I. Parsons and B. Nour-Omid, eds.), vol. CED-Vol. 4, pp. 27–35, New York: ASME, 1991.

[46] J. Fish, "Hierarchical modelling of discontinuous fields," *Communications in Applied Numerical Methods*, vol. 8, no. 7, pp. 443–453, 1992.

[47] J. Fish and S. Markolefas, "Adaptive s-method for linear elastostatics," *Computer Methods in Applied Mechanics and Engineering*, vol. 104, no. 3, pp. 363–396, 1993.

[48] J. Fish, S. Markolefas, R. Guttal, and P. Nayak, "On adaptive multilevel superposition of finite element meshes for linear elastostatics," *Applied Numerical Mathematics*, vol. 14, no. 1–3, pp. 135–164, 1994.

[49] P. K. Moore and J. E. Flaherty, "Adaptive local overlapping grid methods for parabolic systems in two space dimensions," *Journal of Computational Physics*, vol. 98, no. 1, pp. 54–63, 1992.

[50] I. Babuška and J. M. Melenk, "The partition of unity method," *International Journal for Numerical Methods in Engineering*, vol. 40, no. 4, pp. 727–758, 1997.

[51] T. Strouboulis, I. Babuška, and K. Copps, "The design and analysis of the generalized finite element method," *Computer Methods in Applied Mechanics and Engineering*, vol. 181, no. 1–3, pp. 43–69, 2000.

Journal publication: https://doi.org/10.1007/s00466-014-1118-x

[52] T.-P. Fries and T. Belytschko, "The extended/generalized finite element method: An overview of the method and its applications," *International Journal for Numerical Methods in Engineering*, vol. 84, no. 3, pp. 253–304, 2010.

[53] R. Krause and E. Rank, "Multiscale computations with a combination of the h- and p-versions of the finite-element method," *Computer Methods in Applied Mechanics and Engineering*, vol. 192, no. 35-36, pp. 3959–3983, 2003.

[54] A. Düster, A. Niggl, and E. Rank, "Applying the hp–d version of the FEM to locally enhance dimensionally reduced models," *Computer Methods in Applied Mechanics and Engineering*, vol. 196, no. 37-40, pp. 3524–3533, 2007.

[55] D. Schillinger, J. A. Evans, A. Reali, M. A. Scott, and T. J. R. Hughes, "Isogeometric collocation: Cost comparison with galerkin methods and extension to adaptive hierarchical NURBS discretizations," *Computer Methods in Applied Mechanics and Engineering*, vol. 267, pp. 170–232, 2013.

[56] B. A. Szabó and I. Babuska, *Finite element analysis*. New York: John Wiley & Sons, 1991.

[57] E. Rank and O. C. Zienkiewicz, "A simple error estimator in the finite element method," *Communications in Applied Numerical Methods*, vol. 3, no. 3, pp. 243–249, 1987.

[58] B. A. Szabó, A. Düster, and E. Rank, "The p-version of the finite element method," in *Encyclopedia of Computational mechanics* (E. Stein, ed.), Chichester, West Sussex: John Wiley & Sons, Ltd, 2004.

[59] J. R. Shewchuk, "An introduction to the conjugate gradient method without the agonizing pain," tech. rep., 1994.

[60] J. Parvizian, A. Düster, and E. Rank, "Finite cell method," *Computational Mechanics*, vol. 41, no. 1, pp. 121–133, 2007.

[61] 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*, vol. 197, no. 45–48, pp. 3768–3782, 2008.

[62] N. Zander, T. Bog, M. Elhaddad, R. Espinoza, H. Hu, A. 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*, vol. 74, pp. 49–63, 2014.

[63] J. Parvizian, A. Düster, and E. Rank, "Topology optimization using the finite cell method," *Optimization and Engineering*, vol. 13, no. 1, pp. 57–78, 2011.

[64] 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*, vol. 50, no. 4, pp. 445–478, 2012.

[65] 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*, vol. 91, no. 5, pp. 457–471, 2012.

[66] Z. Yang, S. Kollmannsberger, A. Düster, M. Ruess, E. G. Garcia, R. Burgkart, and E. Rank, "Non-standard bone simulation: interactive numerical analysis by computational steering," *Computing and Visualization in Science*, vol. 14, no. 5, pp. 207–216, 2012.

Journal publication: https://doi.org/10.1007/s00466-014-1118-x

[67] 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*, vol. 11, no. 3-4, pp. 425–37, 2012.

[68] A. Abedian, J. Parvizian, A. Düster, and E. Rank, "The finite cell method for the j2 flow theory of plasticity," *Finite Elements in Analysis and Design*, vol. 69, pp. 37–47, 2013.

[69] A. Abedian, J. Parvizian, A. Düster, and E. Rank, "Finite cell method compared to h-version finite element method for elasto-plastic problems," *Applied Mathematics and Mechanics*, vol. 35, no. 10, pp. 1239–1248, 2014.

[70] S. Duczek, M. Joulaian, A. Düster, and U. Gabbert, "Simulation of lamb waves using the spectral cell method," pp. 86951U–86951U–11, 2013.

[71] Q. Cai, *Finite Cell Method for Transport Problems in Porous Media*. Doctoral thesis, Technische Universität München, Munich, 2013.

[72] E. Rank, S. Kollmannsberger, C. Sorger, and A. Düster, "Shell finite cell method: A high order fictitious domain approach for thin-walled structures," *Computer Methods in Applied Mechanics and Engineering*, vol. 200, no. 45-46, pp. 3200–3209, 2011.

[73] E. Rank, M. Ruess, S. Kollmannsberger, D. Schillinger, and A. Düster, "Geometric modeling, isogeometric analysis and the finite cell method," *Computer Methods in Applied Mechanics and Engineering*, vol. 249-252, pp. 104–115, 2012.

[74] M. Ruess, Y. Bazilevs, D. Schillinger, N. Zander, and E. Rank, "Weakly enforced boundary conditions for the NURBS-based finite cell method," in *European Congress on Computational Methods in Applied Sciences and Engineering (ECCOMAS)*, (Vienna, Austria), 2012.

[75] 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*, vol. 95, no. 10, pp. 811–846, 2013.

[76] 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*, vol. 269, pp. 46–71, 2014.

[77] N. Zander, S. Kollmannsberger, M. Ruess, Z. Yosibash, and E. Rank, "The finite cell method for linear thermoelasticity," *Computers & Mathematics with Applications*, vol. 64, no. 11, pp. 3527–3541, 2012.

[78] M. Dauge, A. Düster, and E. Rank, "Theoretical and numerical investigation of the finite cell method," Tech. Rep. hal-00850602, version 2, Université de Rennes, 2014.

[79] 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*, pp. 1–65, 2014.

[80] M. H. Sadd, *Elasticity theory, applications and numerics*. Burlington: Elsevier Butterworth-Heinemann, 2009.

[81] W. F. Mitchell, "The hp-multigrid method applied to hp-adaptive refinement of triangular grids," *Numerical Linear Algebra with Applications*, vol. 17, no. 2-3, pp. 211–228, 2010.

Journal publication: https://doi.org/10.1007/s00466-014-1118-x

[82] N. Hu, X.-Z. Guo, and I. N. Katz, "Multi-p preconditioners," *SIAM J. Sci. Comput.*, vol. 18, no. 6, pp. 1676–1697, 1997.

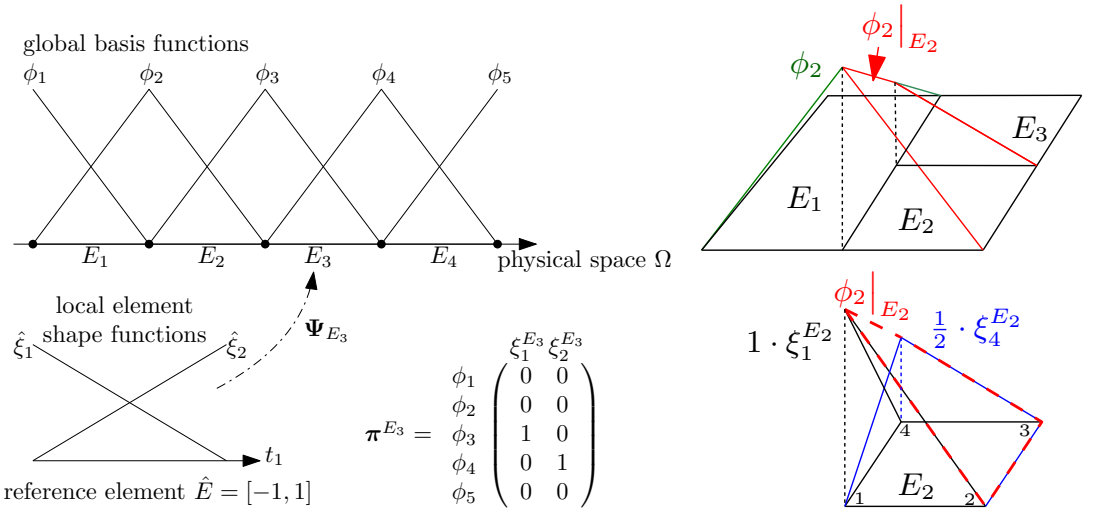
[83] E. L. Wilson, "The static condensation algorithm," *International Journal for Numerical Methods in Engineering*, vol. 8, no. 1, pp. 198–203, 1974.

[84] B. A. Szabó, "Estimation and control of error based on p convergence," in *Accuracy Estimates and Adaptive Refinements in Finite Element Calculations* (I. Babuska, O. C. Zienkiewicz, J. Gago, and E. R. de Oliviera, eds.), pp. 25–55, New York: Wiley, 1986.

[85] R. Niekamp and E. Stein, "The hierarchically graded multilevel finite element method," *Computational Mechanics*, vol. 27, no. 4, pp. 302–304, 2001.

Journal publication: https://doi.org/10.1007/s00466-014-1118-x

A unified concept for hierarchic enrichments of *hp*-type Starting point to this line of research was the pure curiosity of trying to understand whether the multi-level *hp* concept constructs the same space of functions as the classical refine-by-replacement *hp* methodology. If this were the case, clearly all proofs of classical *hp*-methods w.r.t. their convergence properties would directly apply to the multi-level *hp*-method. This idea was brought to attention by Schröder at the HOFEIM workshop in 2014. The approach was to cast the multi-level *hp*-method into the concept of a generalized assembly procedure using connectivity matrices. The concept is depicted in fig. 20.



(a) One-dimensional example of generic assembly

(b) Constrained approximation of global shape function

Figure 20: Connection between global basis functions and local element shape functions in one and two dimensions [65]

It is common practice to compose the global shape functions ϕ_i of an element E from local shape functions $\hat{\xi}_j$ defined on a reference element $\hat{E}$ which is transformed into the physical space by a geometric mapping $\boldsymbol{\Psi}_E : \hat{E} \to E$. The local element shape functions on an element E are thus given by:

$$\xi_j^E = \hat{\xi}_j \circ \boldsymbol{\Psi}_E^{-1}. \tag{3}$$

The idea of the generalized assembly is now to define the mapping between the global and local shape functions by a connectivity *matrix* $\boldsymbol{\pi}^E$ of an element E. In this matrix, every row corresponds to one global shape function ϕ_i and every column to one local shape function ξ_j^E of the respective element. This allows to express each global basis function as a linear combination of local shape functions:

$$\phi_i|_E = \sum_j \pi_{ij}^E \xi_j^E \tag{4}$$

whereby $|_E$ symbolizes the restriction of the global shape functions to that element E. This allows to define the generalized assembly procedure of the global load vector $\boldsymbol{F}$ as a sum of the corresponding matrix vector multiplication of all local element force vectors $\boldsymbol{F}^E$ in the

finite element mesh $\mathcal{E}$:

$$\boldsymbol{F} = \sum_{E \in \mathcal{E}} \boldsymbol{\pi}^E \boldsymbol{F}^E \tag{5}$$

Likewise, the global stiffness matrix $\boldsymbol{K}$ can be assembled as a sum of the local element stiffness matrices $\boldsymbol{K}^E$, which are pre- and post-multiplied by the corresponding connectivity matrix $\boldsymbol{\pi}^E$:

$$\boldsymbol{K} = \sum_{E \in \mathcal{E}} \boldsymbol{\pi}^E \boldsymbol{K}^E \left(\boldsymbol{\pi}^E\right)^\top \tag{6}$$

This assembly is more general than the standard location matrix procedures because it is possible to associate to every global shape function – not only one local shape function but fractions of local shape functions. Consider, for example, the global shape function ϕ_2 in fig. 20b. It can be constructed by combining the first and fourth element shape functions of the second element with a weight of 1 and 1/2, respectively. Accordingly, the second row of the connectivity matrix of element E_2 is:

$$\left(\pi_{2j}^{E_2}\right)_{j=1,\dots,4} = \begin{pmatrix} 1 & 0 & 0 & \frac{1}{2} \end{pmatrix} \tag{7}$$

It is now possible to define the multi-level *hp*-method in the framework of the generalized assembly procedure. To this end, it is first necessary to define the supports of the resulting global basis functions ϕ_i. Next, a rule-set needs to be defined, leading to the corresponding connectivity matrix. This rule set was published in [65], and it turned out to be simpler than the one needed for classical *hp*-methods. Unfortunately, in the course of this work, we found that the function space spanned by the multi-level *hp*-method is similar but not the same as in the classical *hp*-method. However, at least the spaces intersect.

It is interesting to note that by recasting the multi-level *hp*-method into the concept of the generalized assembly, in essence, a refine-by-replacement procedure is defined for the multi-level *hp*-method. This is due to the fact that the generalized assembly procedure defines an assembly of the refined basis directly from a reference element – without the recursive mapping and evaluation procedures, both of which are necessary in the multi-level *hp*-method. Of course, this is to the expense of having to set up the corresponding connectivity matrices, which is a non-trivial task.

The concepts of refine-by-replacement and refine-by-overlay can be considered as two sides of the same medal, each with algorithmic advantages and disadvantages. This is true at least for the case in which the overlays are those given by the hierarchic enrichments of multi-level *hp* type.

Enrichment by replacement The advantage of composing a global discretization directly from a reference element is especially desirable if other types of enrichments are to be included into an already existing code. The corresponding concept to connectivity matrices in isogeometric analysis is Bézier extraction. the one-dimensional element mesh with the associated hierarchical B-spline basis denoted as $\mathcal{HB}$, as depicted in fig. 21.

It is composed of the B-spline functions defined in $\mathcal{B}^0$ and its two overlays $\mathcal{B}^1$ as well as $\mathcal{B}^2$. The superscript denotes the concerned level of refinement l. Both of the overlays are

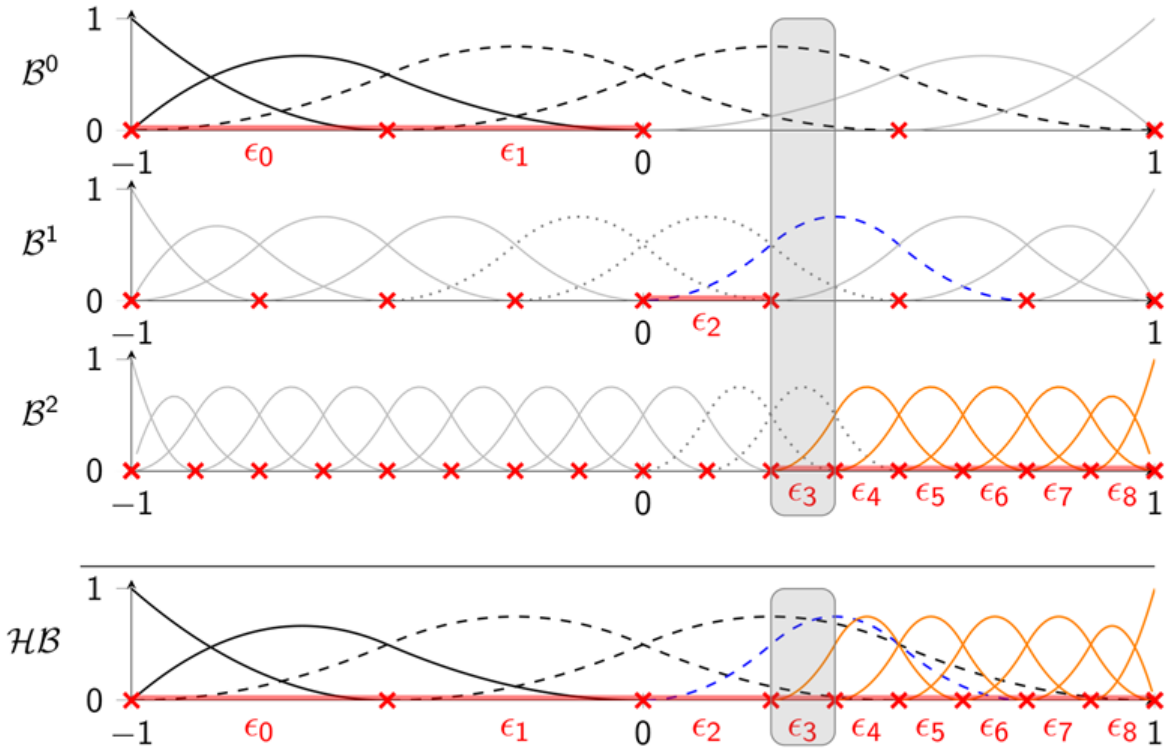


Figure 21: Hierachical B-spline basis as presented in [66]

generated by knot insertion of the knots b_i^l. Elements of the mesh ϵ_j may then be defined as non-overlapping knot-spans. To facilitate the refinement, they are located on different levels in $\mathcal{B}^l$. An example are the elements marked by the thick red lines on the horizontal axes in fig. 21. Their union forms the finite element mesh[5].

Next, a linearly independent basis needs to be constructed. For this purpose, the set of active functions $\mathcal{B}_a^l$ on each element is defined as the set of all functions with support on the considered element. From these, a subset of linearly independent functions is chosen. To this end, the functions in $\mathcal{B}_a^l$ are classified into those which overlap coarser elements ($\mathcal{B}_-^l$), those which overlap finer elements but not coarser ones ($\mathcal{B}_+^l$), and those which only overlap with elements of the level corresponding with that chosen element ($\mathcal{B}_=^l$). Figure 21 depicts the situation for the specific element ϵ_3. $\mathcal{B}_-^l$ is composed of the dotted functions, and $\mathcal{B}_+^l$ is composed of the dashed functions. $\mathcal{B}_=^l$ is composed of the solid non-gray functions. All other functions are set inactive, which is symbolized by their solid gray color. The hierarchical B-spline basis $\mathcal{HB}$ is then defined by the union of the B-splines of each level l whose support covers only elements of level $l* \geq l$ and at least one element of level l. This choice of functions was shown to deliver a linear independent basis in [67].

The above line of argument still takes the point of view in which the construction of refinements is carried out by hierarchical overlays. However, the refine-by-replacement point of view, depicted in fig. 22, is possible as well. To this end, the concept of Bézier extraction is used. It allows to construct standard B-splines directly from Bernstein polynomials defined

[5] which translates to a finite element grid by means of a tensor product to produce discretizations in more than one dimension

on a reference element. Further, as published in [66], it is possible to compute a multi-level extraction operator that composes the multi-level basis for the considered element directly from the standard B-spline basis. These two operators may be combined to form a multi-level Bézier extraction operator. This operator is local to the element, and it serves to directly construct the multi-level basis from a standard reference element upon which Bernstein polynomials are defined.

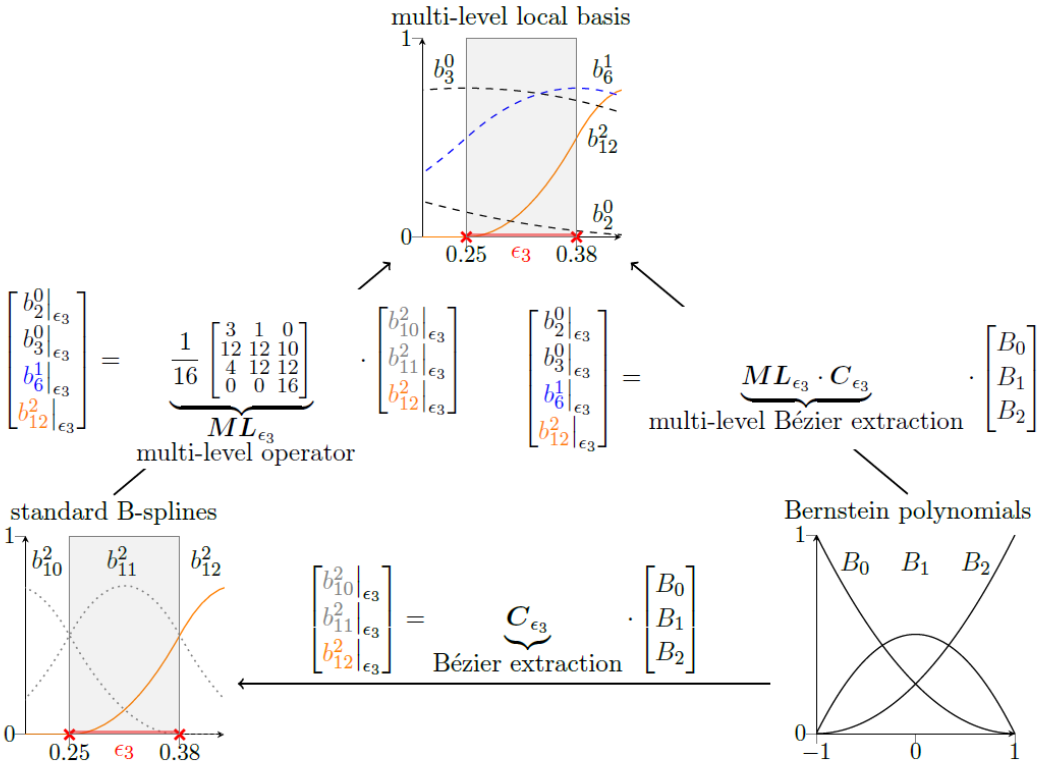


Figure 22: Basis functions of $\mathcal{HB}$ local to the element $\epsilon_3 = (0.25, 0.375)$ of the example in fig. 21, expressed as linear combination of standard functions in $\mathcal{B}^2$ with support on ϵ_3 or of Bernstein polynomials on the reference element as in [66].

This multi-level Bézier extraction operator can also be viewed as a type of a connectivity matrix $\boldsymbol{\pi}^E$, as introduced in the previous paragraph. It carries out the generalized assembly of basis functions of a local reference element to form a global basis and ensures the correct continuity and linear independence of the global discretization.

3.4.3 Refinement indicators and error estimation

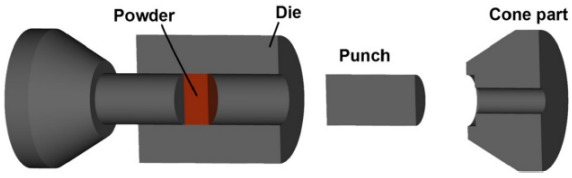
The development of the refinement schemes discussed in section 3.4.2 was driven by the need to resolve small-scale features in the context of the FCM. The next step is to answer the question of how and where to refine in multi-level *hp*. In the context of the FCM, refinement indicators have successfully been used to decide where to refine. For example, an indicator could simply be to check the second derivative of an approximate solution. High values of the second derivatives then indicate refinement, medium ones indicate that there is nothing

to do, and smaller ones indicate coarsening. There are even simpler indicators, such as pre-refinement at re-entrant corners, changes in the boundary conditions, or rapid jumps in the material coefficients of the domain. Unfortunately, error indicators do not estimate the error and, thus, do not provide upper bounds i.e. it is not possible to ensure that the error will be under a certain threshold. To this end, it is common to employ error estimators that provide upper bounds [6]. Numerous types and variants exist, along with a vast body of literature, see e.g. [68] for a comprehensive introduction. Among these, a-posteriori error estimators are convenient because they derive the estimated error from an associated residual which can be computed directly from the numeric approximation to the already computed solution.

The main contribution w.r.t. the multi-level *hp*-method was published in [69], where it was demonstrated that a-posteriori error estimators can also be used in the multi-level *hp*-method. The key issue is to correctly interpret what an element actually is in the refine-by-overly. A correct definition of the extension of an element is crucial because a-posteriori estimators require an evaluation of residuals on element boundaries. As a remedy, the concept of elements was extended to integration domains. The integration domains are sub-domains that are defined by the imprint of all leaf elements upon the base element. The error estimator was then associated to these integration domains instead of the element. Indeed, this definition is consistent with the definition of the partition of computational domains into elements of a classic finite element discretization. Therein, for second order problems, the shape function must represent exactly all polynomials of order up to one (completeness), be C^1-continuous within each element (smoothness), and be C^0-continuous across element boundaries (continuity). The error estimator can then be enhanced by a standard smoothness indicator to drive an adaptive scheme using multi-level *hp* adaptivity. An example is given in fig. 23. This example stems from a project in field assisted sintering where a copper powder is heated by a magnetic field and compacted in a die with a punch in order to form a solid artifact.

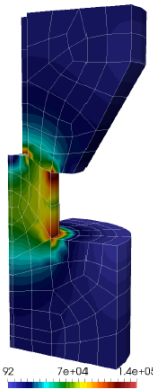
The improvement of the solution by adapted enrichments still has a lot of potential, especially in combination with the Finite Cell Method. There are numerous questions that must yet be addressed theoretically, such as the estimation of errors on badly cut elements for which the multiplicative constant grows out of bounds. Additionally, it remains an open question how to construct an efficient error estimator that is able to treat small voids which are situated inside one finite cell.

[6] at least up to a multiplicative constant

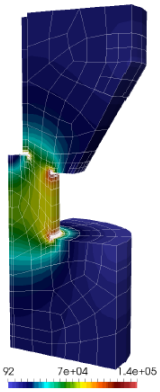


(a) Schematic sketch of the FAST process from [29].

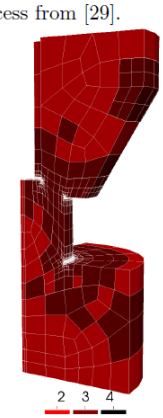
(b) Geometry description in $[mm]$.



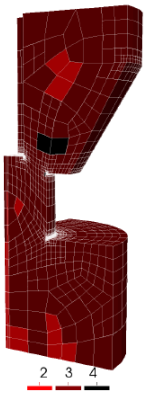
(c) Solution gradient magnitude on initial mesh.



(d) Solution gradient magnitude on refined mesh.



(e) Ansatz order after 30 steps.



(f) Ansatz order after 57 steps.

Figure 23: A geometrically complex example for adaptivity using the multi-level *hp*-method as presented in [69].

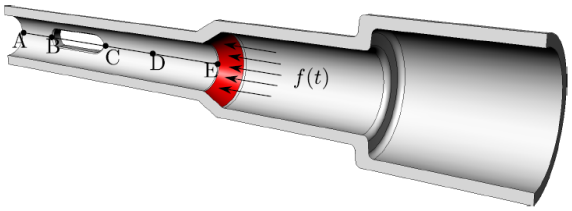
4 Towards engineering applications of the Finite Cell Method

The previous chapters discussed extensions of the FCM, such as its extension to various geometric models in section 2 as well as enhancements in the discretization, see section 3. The underlying motivation for these enhancements are engineering applications. Some of them were already presented in the methodological sections – including the application for biomechanics, see fig. 10, the computation of stresses in engineering parts constructed by CSG, as presented in fig. 6, as well as the approach to handle dirty geometries (see e.g. section 2.3). By contrast, this section discusses engineering applications that benefit from flexibility of the developed method, using it directly to answer questions of engineering relevance. This includes the application of the FCM to structural dynamics section 4.1 and large scale simulations on CT scans section 4.2. Special attention is given to biomedical applications in section 4.3 by including the corresponding journal publication. This treatise then concludes with a process model for Additive Manufacturing in section 4.4. The process of additive manufacturing incorporates numerous physical phenomena and scales, and it poses many challenges such as strong traveling gradients and associated phase changes. These phenomena occur on growing domains. Section 4.4 demonstrates the rather tedious but important cycle of selecting a physical model, constructing a discretization method for it, verifying its implementation, as well as its validation back to an update of the physical model to better represent the observations. To illustrate the details of this scientific work, three publications are included. One in section 4.4.1 on the treatment of thermal processes including phase changes, another on the coupled thermo-mechanical aspects in section 4.4.2, and a final publication on validation in section 4.4.3.

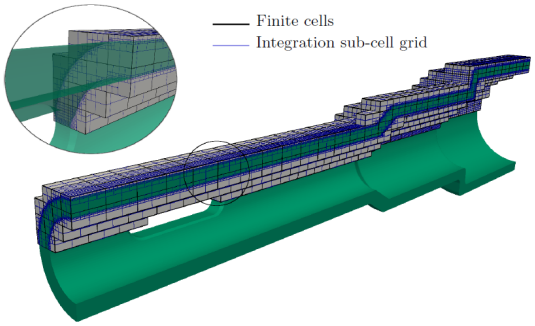
4.1 Structural dynamics

A straightforward extension of the FCM is that towards structural dynamics, with an example of application in fig. 24. It depicts a chuck of a hammer-action drill as used in a product of Hilti. The chuck is subject to an impact load. For this class of industrial applications, there are competitive standard software products that rely on explicit time integration methods – such as LS-DYNA^{TM}, for example. Thus, the most eminent question of interest is whether the same results can be obtained for the FCM computation using the implicit version of the Newmark method as compared to the commercial program LS-DYNA^{TM}using explicit time integration.

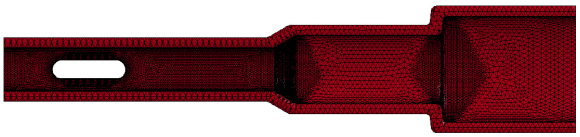
This comparison might seem a little awkward at first, but it does have practical relevance. Stress concentrations develop in the smaller radii of the elliptic holes, e.g. at Point B in fig. 24, for which fine tetrahedral or hexahedral meshes are needed. These, in turn, require very small time steps in an explicit time simulation within LS-DYNA^{TM}. In addition, the maximum of these stress concentrations occurs at a point in time much later than the load impact on the red surface in fig. 24a. Therefore, the total number of time steps necessary for an explicit simulation is very large. This is an ideal setting for an implicit simulation that allows for much larger time steps. However, the high number of degrees of freedom necessary to resolve the stresses accurately enough caused the memory to grow out of bounds. One possible approach to keep the number of degrees of freedom at bay is to seek higher convergence rates, which are provided by the p-version of the finite element method. However, it turned out to be non-trivial to impossible to create a boundary-conforming p-finite element mesh. This is why



(a) geometry and loding



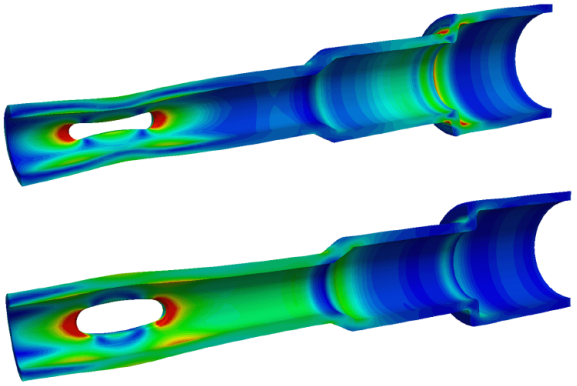
(b) FCM discretization



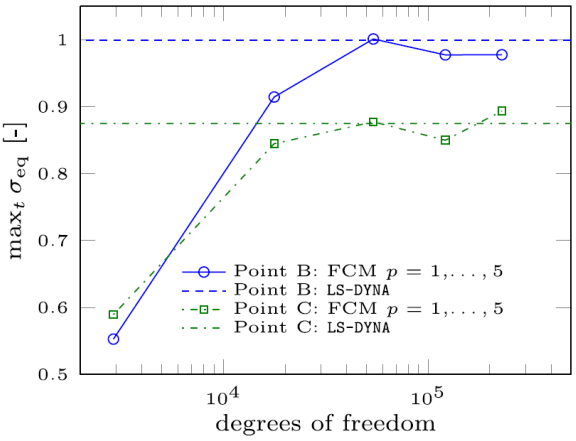
(c) FEM discretization

Figure 24: Chuck of a hammer-action drill

FCM provides a possible solution. Due to intellectual property rights concerning the chuck and its specific design, only a case study was investigated. Comparable accuracy was reached with the discretizations depicted in figs. 24b and 24c. The discretization in LS-DYNA^{TM}used 366,097 tetrahedral finite elements with 230,358 degrees of freedom. The explicit dynamics time integration called for 45,526 time steps. Comparable accuracy is reached with only 533 finite cells at $p = 3$, which delivers only 54,021 degrees of freedom solved in 400 time steps. The fact that the FCM achieved comparable results for an example of industrial interest was presented at conferences [70] and a journal publication [71] in which all results needed to be normalized, see fig. 25 for an excerpt.



(a) Von Mises stresses at different time instances

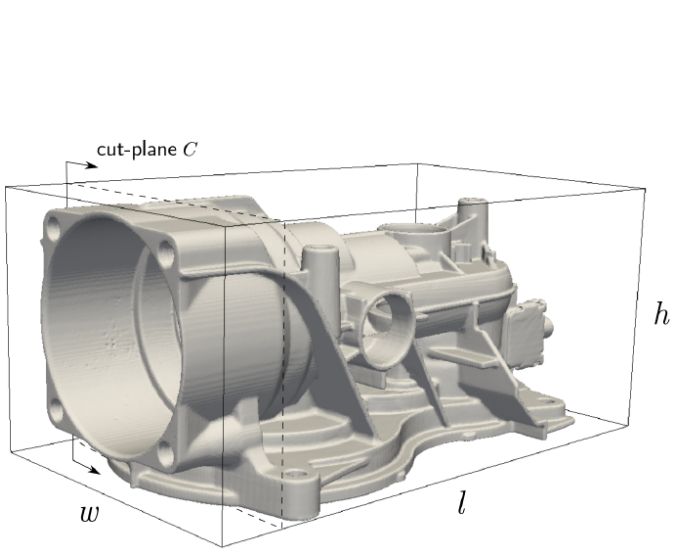


(b) convergence of Von Mises stresses

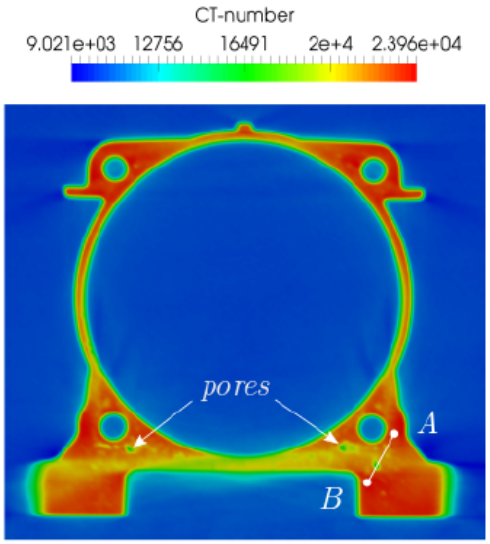
Figure 25: Chuck VonMisesStresses

4.2 Large scale structural snalysis of CT scans

Another engineering application of current interest is the computation of stress concentrations around pores in CT-scanned die casts [72] – for which it is generally possible to employ the FCM as originally developed for CT-scan based simulations in bio-mechanics in a straightforward manner [12], [73, 35]. However, straightforward applications like this are not sufficient to compute locally detailed stress concentrations at pores in these industrial parts. These pores are inherent to the process of die casting, and they are of industrial interest because they can reduce the life time of such parts. A detailed analysis of such situations, however, is possible using the developed hierarchic refinement methods of *hp*-type described in section 3.4. For example, let us consider a gearbox housing and its CT-scan, which is depicted in fig. 26a. Figure 26b provides the corresponding Houndsfield units in the cut in plane C and points out the pores in the lower corners of the artifact.



(a) geometry and plane C



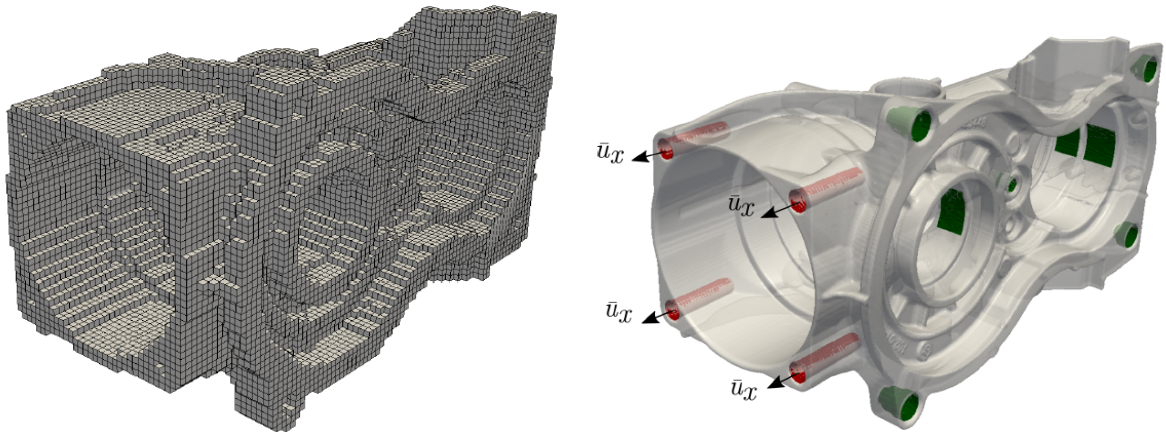
(b) Houndsfield units at plane C along with line A-B

Figure 26: CT-scan of a gearbox housing with pores resulting from a die cast process

The computational analysis is carried out by submerging the gearbox into 34,230 finite cells, as depicted in fig. 27a. They are of order $p = 4$ and deliver approx. 1.66 million degrees of freedom. Homogeneous Dirichlet type boundary conditions are then applied at the dark green surfaces, and a displacement u_x is applied on the red cylindrical surfaces as given in fig. 27b.

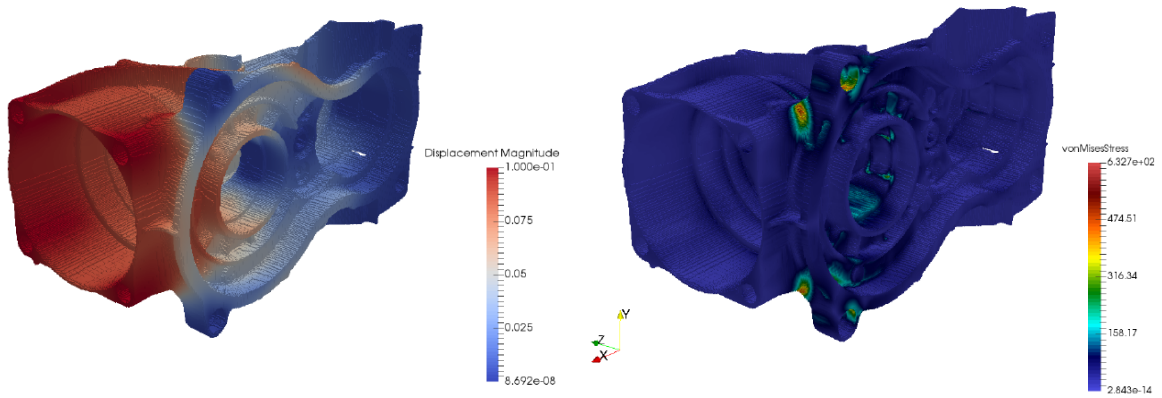
Figure 28 depicts the resulting maximum displacements and the Von Mises stress distribution of the computation. Note how the stresses concentrate around the regions where bolts are placed in the application. This, unfortunately, coincides with the location of the pores where a further increase of the stresses is expected.

The discretization is then refined with 4 multi-level *hp*-refinements towards these pores. This increases the number of degrees of freedom to 1.8 million. The comparison of the results is given in fig. 29. The stress concentrations are clearly under-resolved on the coarse mesh.



(a) Finite Cell discretization **(b)** boundary conditions

Figure 27: Finite Cell discretization and boundary conditions



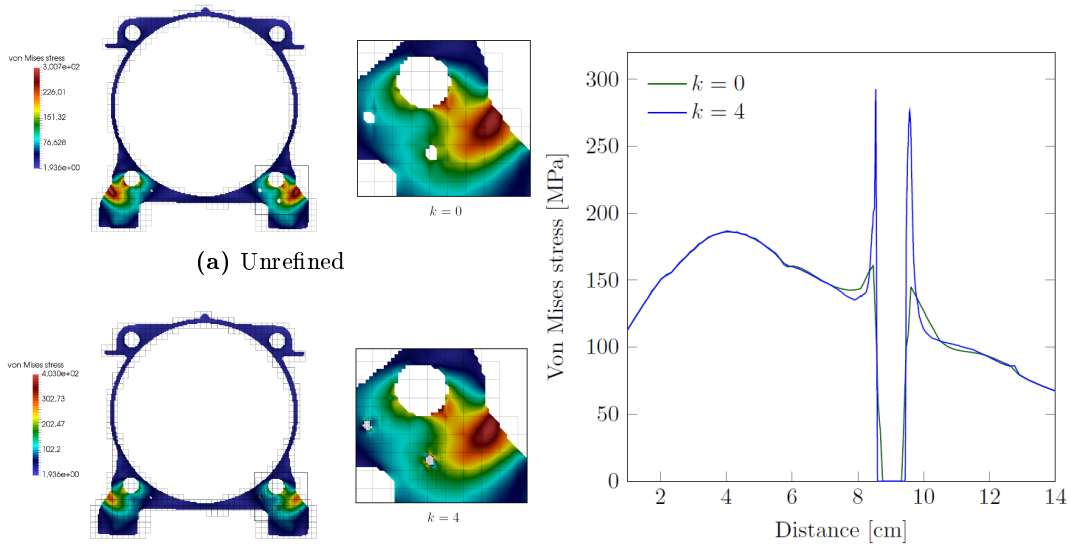
(a) dislpacement magnitude **(b)** Von Mises stresses

Figure 28: Finite Cell global results for dislpacement magnitudes and Von Mises stresses

Only the local refinements provide an insight into the local stresses. The underresolution is even more apparent in fig. 29c, which depicts the Von Mises stresses along the line A-B given in fig. 26b. At the boundary of a cavity, these concentrations reach approximately twice the value of the far field – a result well in accordance with engineering intuition. The possibility to compute stress concentrations locally now opens the door to compute fracture or fatigue in examples of this scale.

In the more detailed analysis, the number of degrees of freedom increases only marginally – which is due to the locality of the more detailed discretization resolving the local stress concentrations.

However, the recent generation of CT scans is able to deliver more detailed (yet much larger) voxel models. Their direct use in mechanical analysis will result in systems with tens of millions of degrees of freedoms, requiring memory capabilities beyond the available single-node version of the code if direct solvers are employed. In these situations, iterative solvers are a valid option. These, however, require proper preconditioning to work efficiently. Precondition-



(a) Unrefined

(b) Refined

(c) stresses along cutline A-B given in fig. 26b

Figure 29: Finite Cell discretization and local Von Mises stresses in the plane C and along the line A-B

ers for system matrices generated by large-scale FCM analysis combined with a parallelization of the multi-level *hp* version of the Finite Cell Method are published in [74, 75].

4.3 Biomedical applications

In the previous section, it was demonstrated by means of examples that the Finite Cell Method can be favorably applied to voxel-based computations. Voxel-based computations are also of use in medical applications, where CT scans are frequently consulted to diagnose defects in bones. However, beyond a geometric assessment of the situation, they may also form a starting point in computational mechanobiology to assess the spatial distribution of stresses in bones.

In this field, the Finite Cell Method delivers accurate results for proximal femurs [76]. Several steps were necessary in this achievement, including a) the use of a material model to map Houndsfield units obtained by CT scans to local Youngs' Moduli, and b) to demonstrate that this material model leads to locally correct strains. Concurrently, an efficient pre-integration technique was developed [77] which is able to compute stiffness matrixes and local updates very rapidly. These techniques allow for an interactive analysis of bones by computational steering, as presented in [78]. Therein, it is demonstrated how principle stress lines in a bone can be computed and visualized in quasi real time. This can be used to visualize the effects of a total hip replacement on the stress field in a femur. In this type of orthopedic operation, the head of the femur is removed and replaced by an implant which is inserted into the shaft of the femur. The geometric aspect of implants in bones was already discussed in section 3.1, where it was pointed out how multiple geometric models (here the voxel model for the bone and the STL model for the implant) may naturally be taken into account within one finite cell computation (see also fig. 10). However, in order to obtain results which are locally accurate one must also take the following discretizational aspects into account.

Firstly, the Young's modulus of an implant is much higher than that of bone. Therefore, implants inserted into bone are to be seen as material interfaces. In these cases, the Finite Cell Method delivers inaccurate results if only the techniques development in [76, 77, 78] are applied. They must be enhanced with any of the options described in section 3.4.1 to deliver locally accurate results. Given the geometric complexity of these internal interfaces, the method of choice is trim-and-glue combined with a simple penalty method. It is demonstrated in [79] that a reasonable choice of the penalty parameter delivers very accurate results for material interface problems.

Secondly, the geometrically complex nature of the material interfaces leads to high local gradients of the strains in the vicinity of the material interfaces. These may be resolved by the any of the local enrichments discussed in section 3.4.2.

The following publication serves to demonstrate that a combination of the techniques discussed above leads to very efficient discretizations for implants – with unprecedented accuracy.

Journal Publication

title:	Multi-level *hp*-finite cell method for embedded interface problems with application in biomechanics
authors:	M. Elhaddad, N. Zander, T. Bog, L. Kudela, S. Kollmannsberger, J. Kirschke, T. Baum, M. Ruess, E. Rank
published at:	*International Journal for Numerical Methods in Biomedical Engineering*
publisher:	Wiley
year:	2018
volume:	34
doi:	https://doi.org/10.1002/cnm.2951

Multi-level hp-finite cell method for embedded interface problems with application in biomechanics

Mohamed Elhaddad*[1], Nils Zander[1], Tino Bog[1], László Kudela[1], Stefan Kollmannsberger[1], Jan S. Kirschke[2], Thomas Baum[2], Martin Ruess[3] and Ernst Rank[1,4]

[1]Chair for Computation in Engineering, Technische Universität München, Arcisstr. 21, 80333 München, Germany
[2]Abteilung für Neuroradiologie, Klinikum rechts der Isar der Technischen Universität München, Ismaninger Str. 22, 81675 Munich, Germany
[3]School of Engineering, University of Glasgow, Rankine Building, Oakfield Ave. G12 8LT, United Kingdom
[4]Institute for Advanced Study, Technische Universität München, Lichtenbergstr. 2a, 85748 Garching, Germany

Abstract

This work presents a numerical discretization technique for solving three-dimensional material interface problems involving complex geometry without conforming mesh generation. The finite cell method (FCM), which is a high-order fictitious domain approach, is used for the numerical approximation of the solution without a boundary-conforming mesh. Weak discontinuities at material interfaces are resolved by using separate FCM meshes for each material sub-domain, and weakly enforcing the interface conditions between the different meshes. Additionally, a recently developed hierarchical hp-refinement scheme is employed to locally refine the FCM meshes in order to resolve singularities and local solution features at the interfaces. Thereby, higher convergence rates are achievable for non-smooth problems. A series of numerical experiments with two- and three-dimensional benchmark problems is presented, showing that the proposed hp-refinement scheme in conjunction with the weak enforcement of the interface conditions leads to a significant improvement of the convergence rates, even in the presence of singularities. Finally, the proposed technique is applied to simulate a vertebra-implant model. The application showcases the method's potential as an accurate simulation tool for biomechanical problems involving complex geometry, and it demonstrates its flexibility in dealing with different types of geometric description.

Keywords: high-order finite elements, hp-adaptivity, finite cell method, domain coupling, embedded interface problems, vertebra-implant model

*mohamed.elhaddad@tum.de, Corresponding author

Preprint submitted to *Int. J. Numer. Meth. Biomed. Engng.* December 20, 2017
The final publication is available at http://onlinelibrary.wiley.com/doi/10.1002/cnm.2951/full

Journal publication: https://doi.org/10.1002/cnm.2951

Contents

Journal publication: https://doi.org/10.1002/cnm.2951

1 Introduction

In engineering problems, an accurate resolution of domains with discontinuous material properties is often required to ensure a reliable design. Possible applications include the simulation of multi-phase materials, composite structures and biomechanical problems, where different tissues—often in combination with implants—are considered.

Problems of linear elasticity show a reduced regularity at the material interfaces, where the displacement field is in general only C^0-continuous. This fits perfectly to the classical finite element method (FEM) using a C^0-continuous approximation space [1, 2], as long as the material interface is matched exactly by edges (2D) or faces (3D) of the finite element mesh. However, this requires that the mesh generator is able to follow the material interfaces exactly, which can be cumbersome in cases of complex geometry. Moreover, high curvatures and kinks along the interfaces induce high solution gradients and singularities. This requires local mesh refinement for an accurate solution, which makes the task of mesh generation even more challenging.

Embedded or fictitious domain methods have emerged as an alternative to the boundary conforming FEM to avoid complex mesh generation. One example of this class of methods is the finite cell method (FCM) introduced in [3], which combines the embedded domain concept with high-order basis functions. The main idea is to extend the originally complex geometry by a fictitious domain to a much simpler shape, which can be easily meshed using a Cartesian grid. The use of high-order basis functions, which smoothly extend into the fictitious domain, allows the FCM to achieve exponential rates of convergence for smooth problems. The FCM has been investigated and successfully applied in various problems, including three-dimensional solid mechanics [4], linear thermoelasticity [5], geometrical non-linearities [6], implicit and explicit elastodynamics [7, 8], and biomechanics [9, 10]. An open-source MATLAB® toolbox for the FCM has been presented in [11] offering an easy entry point into the topic.

Unfortunately, the standard FCM loses some of its attractive approximation properties for domains with multiple materials [12, 13]. There are two main challenges facing the FCM for embedded interface problems:

(1) The approximate solution within a finite cell is a polynomial (C^∞-continuous) whereas the exact solution is only C^0-continuous. The inability of the smooth polynomials to represent the discontinuity across the material interface yields an oscillatory solution and a reduction of the convergence rate.

(2) The use of a uniform Cartesian grid is not suitable for problems with locally high gradients or singularities, which are expected for material interfaces with complex geometry, as such problems require local mesh refinement.

To handle the first challenge, the numerical approximation needs to be able to reproduce a C^0-continuous solution. Several finite element approaches emerged that are suitable to handle weak discontinuities, *without* generating an interface-conforming mesh. Two major approaches are the partition of unity enrichment and the weak coupling of non-matching discretizations.

Partition of unity methods (PUM) [14], such as the generalized and extended finite element methods (GFEM, XFEM) [15, 16] enhance the numerical solution by enriching the basis with specially constructed functions to approximate the weakly discontinuous solutions [17, 18]. Recent developments in XFEM and GFEM for material interface problems include the use of adaptive mesh refinement in conjunction with the level-set method [19, 20], and the interface-enriched GFEM [21–25]. The local enrichment approach was combined with the FCM as presented by Joulaian and Düster for two-dimensional problems [13]. This combination leads to a significant improvement in the convergence rates of the FCM. However, the efficient and robust application of PUM enrichments to complex three-dimensional geometries remains a challenge.

The weak coupling approach allows for non-matching discretizations of the sub-domains, and it enforces the interface conditions in a weak sense. To that end, it is possible to apply the penalty method, Lagrange multipliers or Nitsche's method [26]. Nitsche's method is commonly used to weakly enforce interface conditions, as presented by Hansbo and Hansbo [27], and further developed by Dolbow and Harari [28–31]. The weak coupling approach has also been applied in

Journal publication: https://doi.org/10.1002/cnm.2951

the context of the FCM, using Nitsche's method [32, 33], or based on a parameter-free coupling method [34]. There, it has been shown that a weak enforcement of the interface conditions enables the FCM to recovers its favorable convergence properties for piecewise smooth problems.

To address the second challenge, the finite cell mesh needs to be locally refined. Alternative versions of FCM make use of unstructured meshes, and apply a local mesh refinement to resolve local solution features [35–37]. However, these mesh-based refinement approaches sacrifice the uniform grid structure of FCM, which is particularly useful for image-based geometries that are common in biomechanical applications.

An attractive alternative is the application of the *hp*-version of the FEM for the non-uniform refinement of the underlying FCM meshes. The classical *hp*-FEM [38–40] is a powerful method which uses a combination of *h*- and *p*-refinement to efficiently approximate non-smooth solutions. The main idea is to sub-divide the elements close to the singularities in a recursive manner, in order to achieve a geometrical progression of element size. The resulting mesh has low order elements of small size close to the singularities, and high order elements of larger size further away from the singularities to approximate the smooth solution. While numerically efficient, *hp*-FEM for two- and three-dimensional problems requires to constrain the hanging nodes, which demands for a sophisticated discretization kernel [41, 42].

A simpler variant, which we employ in this work, is the recently introduced multi-level *hp*-version of the FEM [43, 44]. Instead of the classical refine-by-replacement concept, where an element is replaced by a set of smaller elements, the multi-level *hp*-FEM superposes the refined element with a hierarchy of overlay meshes. Using a simple formulation, in which homogeneous Dirichlet boundary conditions are enforced on the overlay meshes, the multi-level *hp*-FEM avoids the need to constrain hanging nodes. Thereby, it offers a simpler implementation for arbitrary hanging nodes, making it readily applicable to three-dimensional problems [45]. The multi-level *hp*-FEM has also been applied to cohesive fracture modeling [46], demonstrating its efficiency for propagating phenomena.

In this contribution, we apply the FCM in conjunction a weak enforcement of interface conditions and the multi-level *hp*-refinement scheme to solve material interface problems. This combination of these numerical techniques enables us to handle complex three-dimensional geometries, without the need for mesh generation. We demonstrate that the multi-level *hp*-scheme is very well-suited for the resolution of geometry induced stress concentrations, and presents a natural compliment to the FCM's strategy to avoid mesh generation. In particular, we demonstrate in this manuscript the use of the proposed refinement scheme in the context of biomechanical simulations, where a combination of heterogeneous tissue material properties and sharp material interfaces between tissue and osteosynthesis implants often requires a robust and flexible modeling approach to ensure reliable and accurate analysis results.

This paper is organized as follows: Section 2 offers a brief review of the finite cell method for embedded interface problems. In Section 3, we outline the multi-level *hp*-refinement scheme, and elaborate on the refinement strategy applied for material interface problems. In Section 4, we present a series of numerical experiments of embedded interface problems with geometry-induced stress concentrations. The results demonstrate the improved approximation accuracy of coupled FCM with multi-level *hp*-refinement. Finally, an application in biomechanics is presented in Section 5, where a thoracic vertebra with pedicle screws is simulated. The paper closes with a concluding outlook in Section 6.

2 The finite cell method for embedded interface problems

In this section, we present a brief review of the finite cell method for material interface problems, outlining the main concept of the method—including a coupling formulation that allows to weakly impose the interface conditions among sub-domains.

2.1 Model problem

We consider an open and bounded domain, denoted as the physical domain $\Omega_{\text{phy}} \subset \mathbb{R}^d, d \in \{2, 3\}$, with the boundary $\partial\Omega_{\text{phy}} = \Gamma_D \cup \Gamma_N$, and $\Gamma_D \cap \Gamma_N = \emptyset$ [32]. The physical domain Ω_{phy} is subdi-

Journal publication: https://doi.org/10.1002/cnm.2951

vided into n disjoint, open, and bounded subsets $\Omega^{(k)}$

$$\Omega_{\text{phy}} = \bigcup_{k=1}^{n} \Omega^{(k)}, \quad \Omega^{(i)} \cap \Omega^{(j)} = \emptyset \text{ for } i \neq j. \tag{1}$$

The sub-domains $\Omega^{(k)}$ are separated by internal boundaries Γ_{ij}, denoting the material interfaces between $\Omega^{(i)}$ and $\Omega^{(j)}$. The model is governed by the partial differential equations of linear elastostatics, given in the strong form of the boundary value problem by:

$$\nabla \cdot \boldsymbol{\sigma}^{(i)} + \boldsymbol{b}^{(i)} = \mathbf{0} \qquad \forall \boldsymbol{x} \in \Omega^{(i)}, \quad i = 1, \dots, n \tag{2a}$$

$$\boldsymbol{\sigma}^{(i)} = \mathbb{C}^{(i)} : \boldsymbol{\varepsilon}(\boldsymbol{u}^{(i)}) \qquad \forall \boldsymbol{x} \in \Omega^{(i)}, \quad i = 1, \dots, n \tag{2b}$$

$$\boldsymbol{\varepsilon}(\boldsymbol{u}^{(i)}) = \frac{1}{2}\left(\nabla \boldsymbol{u}^{(i)} + \nabla \boldsymbol{u}^{(i)^{\top}}\right) \qquad \forall \boldsymbol{x} \in \Omega^{(i)}, \quad i = 1, \dots, n, \tag{2c}$$

where $\boldsymbol{\sigma}^{(i)}$ is the stress tensor, $\boldsymbol{b}^{(i)}$ is the body load, $\mathbb{C}^{(i)}$ is the elastic material tensor, $\boldsymbol{\varepsilon}$ is the strain tensor, and $\boldsymbol{u}^{(i)}$ is the displacement vector of the sub-domain $\Omega^{(i)}$. The system is subject to the boundary conditions:

$$\boldsymbol{u}^{(i)} = \hat{\boldsymbol{u}}^{(i)} \qquad \forall \boldsymbol{x} \in \Gamma_D^{(i)}, \quad i = 1, \dots, n \tag{3a}$$

$$\boldsymbol{\sigma}^{(i)} \cdot \boldsymbol{n}^{(i)} = \hat{\boldsymbol{t}}^{(i)} \qquad \forall \boldsymbol{x} \in \Gamma_N^{(i)}, \quad i = 1, \dots, n, \tag{3b}$$

where $\hat{\boldsymbol{u}}^{(i)}$ are the prescribed displacements on the Dirichlet boundary, $\hat{\boldsymbol{t}}^{(i)}$ are the prescribed tractions on the Neumann boundary, and $\boldsymbol{n}^{(i)}$ denotes the outward facing normal vector on the sub-domain's boundary $\partial\Omega^{(i)}$ (see Figure 1), and

$$\Gamma_D^{(i)} = \Gamma_D \cap \partial\Omega^{(i)} \tag{4a}$$

$$\Gamma_N^{(i)} = \Gamma_N \cap \partial\Omega^{(i)}. \tag{4b}$$

Additionally, the following interface conditions apply along Γ_{ij}:

$$\boldsymbol{u}^{(i)} = \boldsymbol{u}^{(j)} \qquad \forall \boldsymbol{x} \in \Gamma_{ij}, \tag{5a}$$

$$\boldsymbol{\sigma}^{(i)} \cdot \boldsymbol{n}^{(i)} = -\boldsymbol{\sigma}^{(j)} \cdot \boldsymbol{n}^{(j)} \qquad \forall \boldsymbol{x} \in \Gamma_{ij}. \tag{5b}$$

The first interface condition ensures the kinematic compatibility between the sub-domains, whereas the second condition prescribes the equilibrium of normal traction across the interfaces.

2.2 Aspects of the finite cell method

The finite cell method is motivated by the superior convergence properties achievable by the p-version of FEM. The geometry-conforming p-FEM requires special attention to mesh generation, which can become prohibitive for problems involving complex geometries. The FCM avoids the need for complex mesh generation procedures by combining the p-FEM with an embedded domain approach.

The principal idea of the FCM is to augment the physical domain Ω_{phy} by a fictitious domain Ω_{fict} to form a simple-shaped domain $\hat{\Omega} = \Omega_{\text{phy}} \cup \Omega_{\text{fict}}$. In order to recover the original problem, the influence of the fictitious domain is attenuated by penalization of the material parameters in the fictitious domain:

$$\mathbb{C}_{\text{fict}} = \epsilon \cdot \mathbb{C}_{\text{phy}}, \quad \epsilon \ll 1 \tag{6}$$

The modified problem is solved over the extended domain, which can be easily meshed with a Cartesian grid of finite cells. The FCM makes use of high-order basis functions which smoothly extend over the embedding domain to approximate the physical solution field. Typical choices of basis functions are the integrated Legendre polynomials, commonly used in the p-FEM [3],

Journal publication: https://doi.org/10.1002/cnm.2951

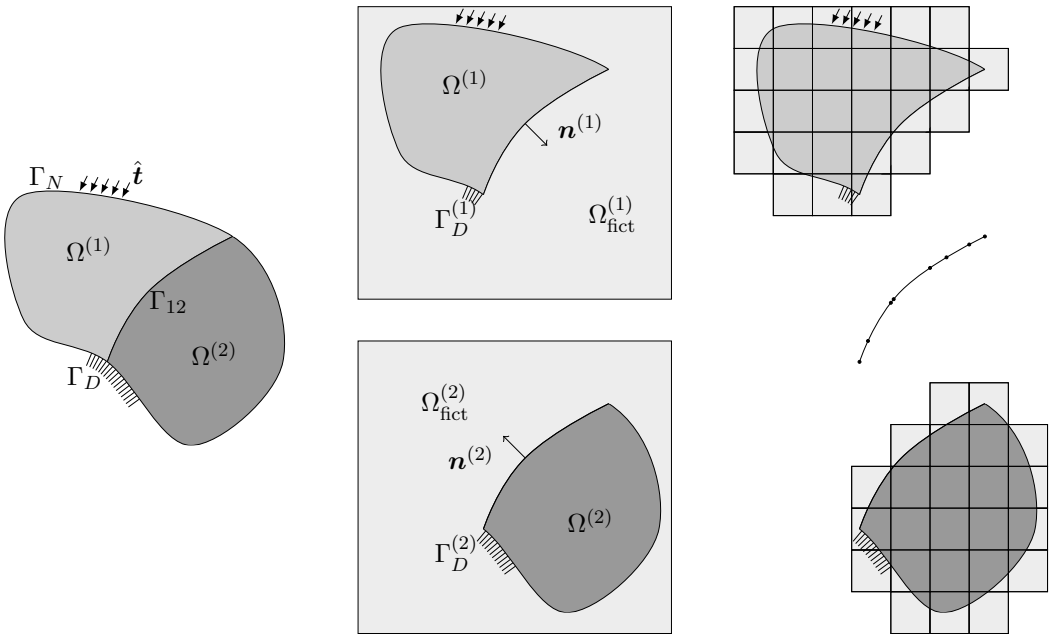


Figure 1: Coupling of FCM meshes, following [32].

and B-Splines [6]. For problems with smooth solutions, the FCM achieves exponential rates of convergence, owing to the underlying high-order basis functions, without the need for a boundary-conforming mesh [3, 4, 6].

The FCM has two salient differences to the conventional finite element method. Firstly, the FCM needs a specialized approach for the numerical integration of the weak form. Due to the discontinuity introduced by the penalization of the fictitious domain, conventional Gauss-Legendre quadrature performs poorly for FCM. Several approaches have been developed to ensure an accurate and efficient evaluation of the discontinuous domain integrals. One possibility is to use a fine grid of integration sub-cells to partition each finite cell, with the Gauss-Legendre quadrature applied for each sub-cell [3, 47]. An alternative approach applies recursive sub-division of the finite cells cut by the boundary, leading to a quadtree of integration sub-cells (octree for the three-dimensional case) [6]. The quadtree approach is robust and easy to implement, but generates a high number of quadrature points. More sophisticated integration schemes have been developed to efficiently evaluate the discontinuous domain integrals for the FCM. These schemes include the blended partitioning using the smart-octree [48, 49], moment-fitting [50], adaptively weighted quadratures [51, 52], and quadratic re-parametrization for the tetrahedral finite cell method. [33]

The other principal difference to conventional FEM is the enforcement of Dirichlet boundary conditions. As the boundary is no longer directly resolved by the finite elements, the Dirichlet boundary conditions cannot be enforced in the classical manner by manipulating the corresponding entries of the stiffness matrix and the load vector [53]. The are several possibilities to impose embedded Dirichlet boundary conditions in the FCM—including the penalty method, and Nitsche's method [5, 54, 34].

2.3 Weak enforcement of interface conditions

For material interface problems, the standard FCM—where a single mesh is used to discretize the entire extended domain—requires additional means to maintain its attractive approximation properties. The solution exhibits a weak discontinuity across the interface (a kink in the displacement field), and this discontinuity cannot be well represented by the high-order polynomial basis functions used in the FCM. This leads to an oscillatory solution in the cells cut by the material

Journal publication: https://doi.org/10.1002/cnm.2951

interface, and the convergence is reduced to a slow algebraic rate.

To overcome this challenge, we use a separate FCM mesh for each extended sub-domain:

$$\hat{\Omega}^{(i)} = \Omega^{(i)} \cup \Omega^{(i)}_{\text{fict}}, \tag{7}$$

and couple the FCM meshes to weakly impose the interface conditions, as presented in [32]. Figure 1 illustrates this approach for two sub-domains. The use of a separate set of basis functions for each sub-domain allows the discretization to approximate the strain fields with different gradients on each sub-domain. Additionally, the coupling of the FCM meshes recovers the weak discontinuity (jump in the strain field). Thereby, FCM is able to recover its exponential convergence rates for piecewise-smooth problems [32, 34]. However, this multiple-mesh approach requires an additional discretization of the material interfaces Γ_{ij} for the integration of the coupling terms along the interfaces. Fortunately, the discretization of the interface for integration is considerably less demanding than FEM mesh-generation for multiple sub-domains.

In the present work, we use a pure penalty approach for the coupling of FCM meshes, and for the weak enforcement of Dirichlet boundary conditions. The penalty method is variationally inconsistent, and the accuracy depends on the choice of the penalty parameters. However, it is easy to implement—and it delivers sufficiently accurate solutions (for engineering purposes), as demonstrated by the numerical experiments presented in Section 4.

Finally, the weak form of the coupled FCM problem reads:

Find $\boldsymbol{u}^{(i)} \in \mathcal{S}(\hat{\Omega}^{(i)}), \quad i = 1, \dots, n$ such that:

$$\hat{\mathcal{B}}(\boldsymbol{u}, \boldsymbol{v}) = \hat{\mathcal{F}}(\boldsymbol{v}) \qquad \forall \boldsymbol{v}^{(i)} \in \mathcal{V}(\hat{\Omega}^{(i)}) \tag{8}$$

with:

$$\begin{aligned}
\hat{\mathcal{B}}(\boldsymbol{u}, \boldsymbol{v}) = &\sum_i \int_{\hat{\Omega}^{(i)}} \boldsymbol{\varepsilon}(\boldsymbol{v}^{(i)}) : \alpha^{(i)}(\boldsymbol{x})\ \mathbb{C}^{(i)} : \boldsymbol{\varepsilon}(\boldsymbol{u}^{(i)})\, \mathrm{d}\Omega \\
&+ \sum_{\substack{ij \\ i<j}} \beta_{ij} \int_{\Gamma_{ij}} [\![\boldsymbol{v}]\!]_{ij} \cdot [\![\boldsymbol{u}]\!]_{ij}\, \mathrm{d}\Gamma + \sum_i \beta_D^{(i)} \int_{\Gamma_D^{(i)}} \boldsymbol{v}^{(i)} \cdot \boldsymbol{u}^{(i)}\, \mathrm{d}\Gamma
\end{aligned} \tag{9a}$$

$$\begin{aligned}
\hat{\mathcal{F}}(\boldsymbol{v}) = &\sum_i \int_{\hat{\Omega}^{(i)}} \boldsymbol{v}^{(i)} \cdot \alpha^{(i)}\ \boldsymbol{b}^{(i)}(\boldsymbol{x})\, \mathrm{d}\Omega + \sum_i \int_{\Gamma_N^{(i)}} \boldsymbol{v}^{(i)} \cdot \hat{\boldsymbol{t}}^{(i)}\, \mathrm{d}\Gamma \\
&+ \sum_i \beta_D^{(i)} \int_{\Gamma_D^{(i)}} \boldsymbol{v}^{(i)} \cdot \hat{\boldsymbol{u}}^{(i)}\, \mathrm{d}\Gamma
\end{aligned} \tag{9b}$$

with the jump operator

$$[\![\boldsymbol{u}]\!]_{ij} = \boldsymbol{u}^{(i)} - \boldsymbol{u}^{(j)} \tag{10}$$

and

$$\alpha^{(i)}(\boldsymbol{x}) = \begin{cases} 1 & \forall \boldsymbol{x} \in \Omega^{(i)} \\ \epsilon^{(i)} \ll 1 & \forall \boldsymbol{x} \in \Omega^{(i)}_{\text{fict}}. \end{cases} \tag{11}$$

The additional integral along Γ_{ij} adds a numerical penalty to the displacement jump across the interface. With a suitable choice of the penalty parameter β_{ij}, the kinematic compatibility condition is weakly enforced.

Journal publication: https://doi.org/10.1002/cnm.2951

3 Multi-level *hp*-refinement

For problems with local solution characteristics such as stress concentration and singularities, *hp*-refinement is a powerful method of discretization, as it is more efficient than uniform *h*- or *p*-refinement [38–40]. Smaller elements with low polynomial degree close to the singularities are able to capture the local solution and confine the discretization error, whereas large elements with high polynomial degree are best suited to describe the smooth solution further away from singularities.

However, the Cartesian grid nature of the FCM discretization does not easily incorporate local mesh refinement without having to introduce hanging nodes. In this work, we apply the multi-level *hp*-refinement scheme to locally refine FCM grids, without the difficulties of constraining hanging nodes. Thereby, the method is able to achieve higher convergence rates, while retaining the advantages of using Cartesian finite cell grids. This section provides a brief review of the multi-level *hp*-FEM and elaborates on the refinement strategy used for material-interface problems.

3.1 Basic refinement concept

The principal idea of the multi-level *hp*-FEM is to enhance the local approximation of the solution through a local superposition of coarse base elements in the refinement zones with finer overlay elements. The final approximation $\boldsymbol{u}$ is the sum of the base mesh solution $\boldsymbol{u}_b$ and the fine-scale overlay solution $\boldsymbol{u}_o$. This *refine-by-superposition* approach is related to the pioneering work of Mote [55]. This concept was applied in the context of *hp*-domain decomposition, to overlay coarse *p*-FEM meshes with fine linear *h*-elements [56], and in the hierarchic *hp-d* scheme, which adds several layers of linear meshes [57].

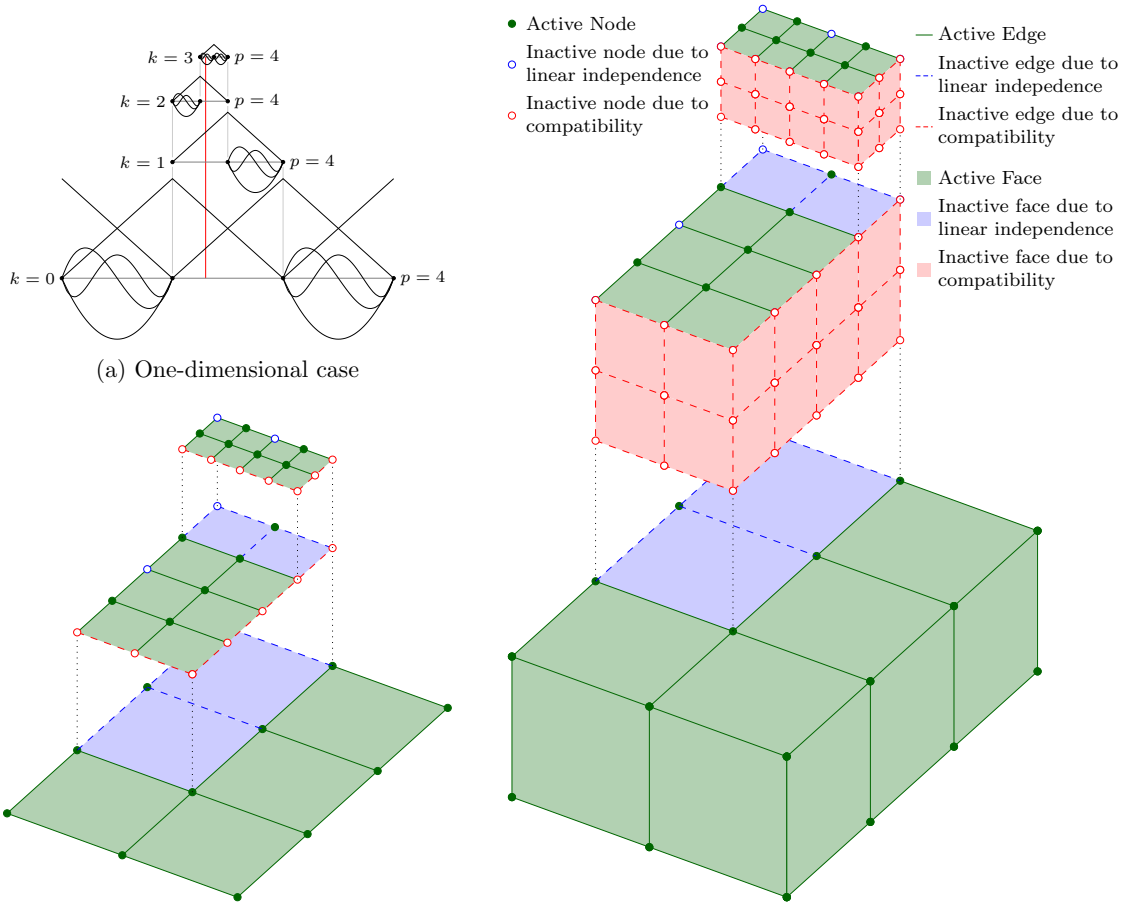
The key feature of the multi-level *hp*-scheme is the use of high-order hierarchic overlay meshes. The support of the high-order basis functions is chosen on the finer overlay meshes—instead of the base discretization, as illustrated in Figure 2. This corresponds to an *h*-refinement of the high-order basis functions, as the size of their support is reduced. This limited support—combined with an adequate geometric grading—has the desirable effect of confining the (global) pollution error in the smallest elements around the singularities. Consequently, the multi-level *hp*-FEM leads to superior convergence rates for problems with singularities and sharp features. Numerical experiments demonstrate that the multi-level *hp*-FEM converges exponentially, for problems including vertex and edge singularities in two and three dimensions [43, 45, 44].

The multi-level *hp*-FEM avoids the implementational burden of constraining hanging nodes. A simple implementation is possible, wherein two requirements are satisfied to ensure convergence: *compatibility* of the discretization, and *linear independence* of the basis functions. The overlay meshes, generated by recursive sub-divisions of the base mesh, are geometrically incompatible. To guarantee compatibility of the discretization, homogeneous Dirichlet boundary conditions are prescribed on the boundary of each layer of overlay meshes. This maintains a C^0-continuous approximation by construction. To ensure linear independence of the basis functions, the high-order shape functions on overlay elements are excluded from their respective parent elements, as demonstrated in Figure 2. The fulfillment of these requirements leads to a simple yet powerful discretization technique, that exploits the benefits of the *hp*-FEM without the burden of constraining hanging nodes. For an in-depth discussion of the multi-level *hp*-FEM for two- and three-dimensional problems, with elaboration of their implementational aspects, the interested reader is referred to [43, 45, 44]

3.2 Refinement strategy

In general, stress concentration and sharp solution features can be caused either by sudden changes in the boundary conditions, the applied body load, or sharp geometric features—such as re-entrant corners. In this work, we mainly consider stress concentrations induced by sharp geometric features of the material interfaces. To resolve such features, an *a priori* refinement scheme is applied. Here, the design of the discretization is guided by existing knowledge of the solution features, based on engineering experience.

Journal publication: https://doi.org/10.1002/cnm.2951



(a) One-dimensional case

(b) Two-dimensional case

(c) Three-dimensional case

Figure 2: Conceptual idea of the multi-level *hp*-method [45].

3.2.1 *h*-refinement

Typically, *hp*-mesh design includes a geometric progression of elements towards the singularities, with a linear increase in the polynomial degree of the basis functions. In the framework of the multi-level *hp*-FEM, the high-order overlay meshes are generated by recursive bisection of the base mesh. To geometrically grade the mesh towards a point $\boldsymbol{P_x} = (p_x, p_y, p_z)$, for each leaf element K, the point is mapped into the index space of the element $\Omega_\square = [-1, 1]^d$ as follows:

$$\boldsymbol{P}_{\boldsymbol{\xi}_K} = \boldsymbol{Q}_K^{-1}(\boldsymbol{P_x}), \tag{12}$$

where $\boldsymbol{P}_{\boldsymbol{\xi}_K} = (p_\xi, p_\eta, p_\zeta)$ are the local coordinates of $\boldsymbol{P_x}$ in the index space of the leaf-element K, and $\boldsymbol{Q}_K(\boldsymbol{x})$ denotes the geometric mapping between the global and local spaces. The element is sub-divided if the mapped point lies within an extended index space $\hat{\Omega}_\square$, which is slightly larger than $\Omega_\square$

$$\boldsymbol{P}_{\boldsymbol{\xi}_K} \in \hat{\Omega}_\square \tag{13}$$

Thereby, not only the elements containing $\boldsymbol{P_x}$ are refined, but also the neighboring elements. By carrying out several recursive refinement steps, this approach produces a geometrically graded mesh—with decreasing element size towards the point of interest. For Cartesian grid meshes, which are used in FCM, this inverse mapping carries a trivial computational cost.

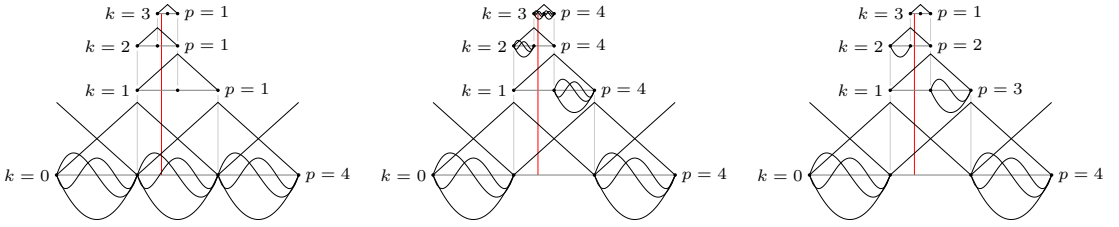
Journal publication: https://doi.org/10.1002/cnm.2951

3.2.2 p-distribution

The distribution of the polynomial degree p across the layers determines the convergence properties of the hp-discretization. The optimum p-distribution is not known *a priori*.

In one case, the high-order basis functions can be kept on the base mesh, and only linear basis functions are used on the overlay meshes (Figure 3a). This case corresponds to the hierarchical hp-d scheme [57], as a special case of the multi-level hp-scheme. The linear overlay meshes introduce a moderate number of additional unknowns. However, the large support of the high-order basis functions causes the global error to propagate through the domain, yielding a minimal improvement in the convergence rates [43].

On the other end of the spectrum, the high-order basis functions are all shifted to the highest (finest) overlay mesh (Figure 3b). Obviously, this uniform multi-level hp-approach introduces significantly more unknowns. However, with the limited support of the high-order functions, the discretization error is confined to the smallest elements, which yields superior convergence properties.



(a) Linear overlay, hp-d-approach (b) Uniform multi-level hp-approach (c) Non-uniform multi-level hp-approach

Figure 3: Comparison of different approaches for hierarchical, high-order refinement [45].

In this work, we adopt the uniform multi-level hp refinement approach, which is equivalent to uniform p-elevation on a geometrically graded mesh. This scheme has been shown to be robust and efficient, although it might not be optimal. For boundary conforming problems, an *a posteriori* error estimator for multi-level hp-FEM coupled with a smoothness indicator was developed [58], where automatically generated multi-level hp-FEM discretizations with non-uniform p-distributions further improved the efficiency.

4 Numerical experiments

In this section, we examine a series of benchmarks involving material interfaces where the geometry induces high solution gradients and singularities. This includes the two-dimensional benchmark of a plate with an elliptical inclusion, where we perform a systematic study of the influence of the numerical integration accuracy on the quality of the approximation. The second benchmark considers a circular plate with a sharp inclusion, where the kink in the material interface introduces a vertex singularity. In three dimensions, we examine the problem of an embedded ellipsoidal inclusion, as well as a singular benchmark of a cylinder with an embedded cubical inclusion, which causes a vertex-edge singularity. In all examples, we examine the p-convergence and the numerical approximation of the first derivatives of the solution.

4.1 Plate with an elliptical inclusion

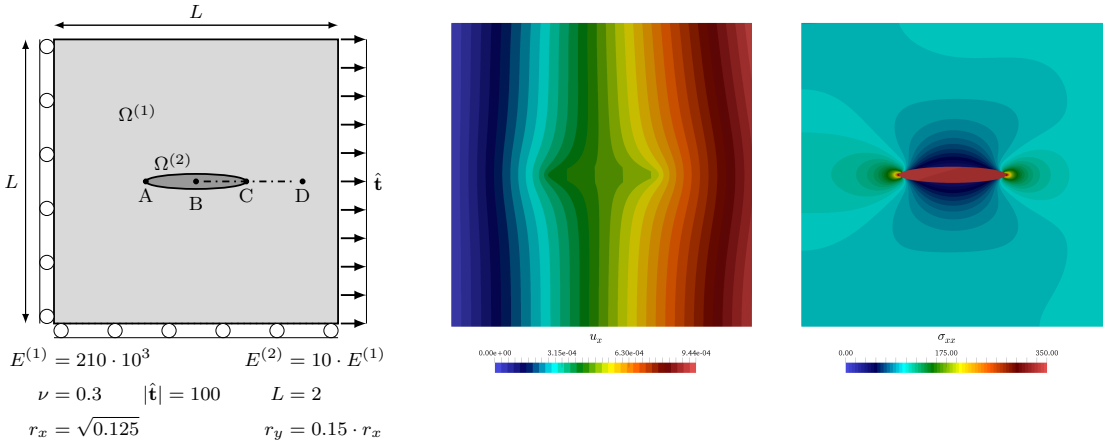
4.1.1 Problem setup

The first material interface benchmark we study is a plane stress problem of a plate $\Omega^{(1)}$ with an embedded elliptical inclusion $\Omega^{(2)}$, which has a large aspect ratio, $r_y/r_x = 0.15$, and a high

Journal publication: https://doi.org/10.1002/cnm.2951

stiffness ratio to the plate, $E^{(2)}/E^{(1)} = 10$ (see Figure 4a). The displacement field within the plate is distorted by the curved inclusion, leading to high stress concentration in the plate. The stress concentration is amplified by the high curvature of the interface at the major axis of the ellipse (points A and C) as shown in Figures 4c and 4b. The displacement field within the elliptic inclusion remains smooth. The exact solution for the displacements is piecewise-analytic, i.e. it can be *exactly* represented by a Taylor expansion in each sub-domain. An overkill solution was obtained for reference using a conforming p-FEM mesh which has 26 elements with a uniform polynomial degree $p = 30$ and 47,102 degrees of freedom. In the conforming mesh, the elements at the interface are blended on the ellipse's exact geometry using a quasi-regional mapping [59]. For the setup shown here, the strain energy obtained using the overkill discretization is

$$\mathcal{U}_{\text{ex}} = 9.10131116644 \times 10^{-2}. \tag{14}$$



(a) Benchmark setup (b) Reference u_x displacement (c) Reference σ_{xx} stress

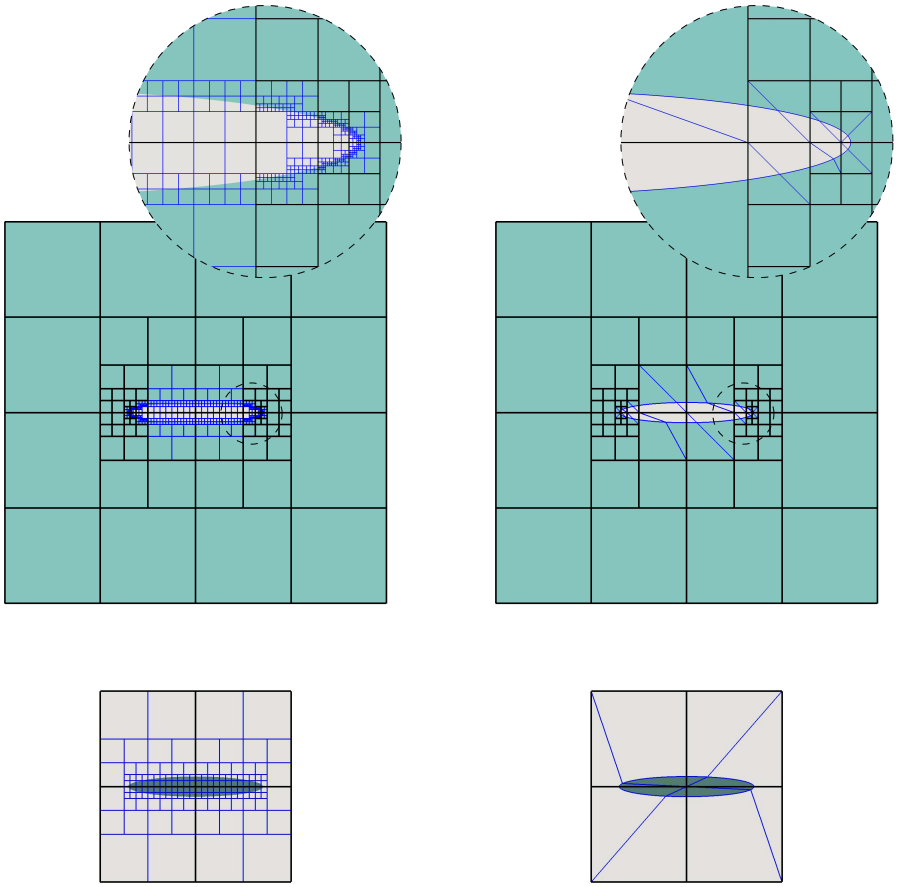
Figure 4: Plate with elliptical inclusion

4.1.2 Discretization

The problem was solved numerically based on the FCM and a weak enforcement of the interface conditions. Two separate FCM meshes were used to discretize the plate and the inclusion. A base discretization matching the outer boundary of the plate with 4×4 finite cells was used. The same parametrization was used for the inclusion's mesh, excluding cells that do not intersect the ellipse's domain, leaving 2×2 finite cells—as shown in Figure 5. The ellipse's dimensions and the mesh parameters are setup such that the mesh does not exactly match points A or C for any number of recursive bisection steps. To adequately describe the local solution in the plate, the mesh was locally refined using the uniform multi-level *hp*-refinement scheme. The mesh was refined *a priori*, with a geometrically driven grading towards points A and C, as described in Section 3.2. The FCM mesh of the inclusion was left without local refinement, as the solution within the inclusion $\Omega^{(2)}$ is smooth. To avoid ill-conditioning of the stiffness matrices, the fictitious domain in both meshes was assigned a low stiffness of $E_{\text{fict}} = E^{(1)} \cdot 10^{-11}, \nu_{\text{fict}} = 0.3$. Since the exterior boundary of the plate's mesh conforms to the true geometry, the Dirichlet and Neumann boundary conditions were enforced in a conventional manner.

For the numerical integration of the finite cells, two possibilities were considered. The first approach employs a recursive sub-division algorithm for integration sub-cells that are intersected by the geometric boundary, resulting in a quadtree—as shown in [6], for example. The second approach considered is the blended partitioning following [48], where the cut cells are automatically partitioned into boundary-conforming quadrilaterals and triangles, which are blended on the exact

Journal publication: https://doi.org/10.1002/cnm.2951



(a) Quadtree partitioning, $m = 4$, following [57] (b) Blended partitioning, following [48]

Figure 5: Domain discretization for the plate and the inclusion, refinement depth $k = 4$. The finite cells' edges are drawn with thick black lines, whereas the integration meshes are drawn with thin blue lines.

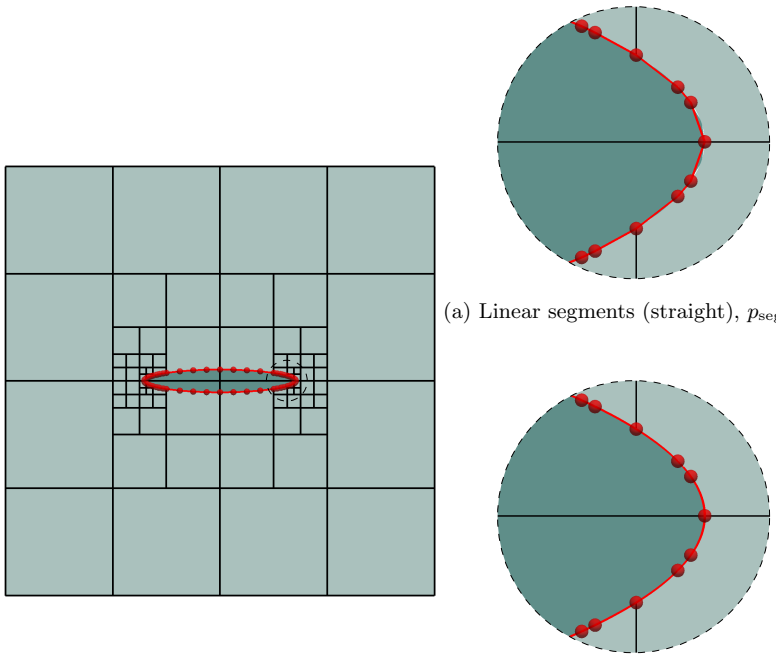
geometry. The integration meshes for both approaches are displayed in Figure 5.

The penalty method was used to weakly couple the two meshes at the material interface as described in Section 2.3. The penalty parameter $\beta_{12} = 10^7$ was chosen empirically. For the numerical integration of the interface coupling terms, a linear discretization of the embedded interface was generated using a marching squares algorithm[60]. The marching squares implementation uses the implicit geometric description of the ellipse and a grid of $n_g \times n_g$ points per sub-cell (leaf element) of the plate's mesh, yielding a total of n_{seg} linear segments, as can be seen in Figure 6a. An advantage of this approach is that the resulting segments conform to the finite cell edges, resulting in a more accurate evaluation of the interface integral, which the terms of which are only C^0-continuous. Additionally, the segments have the same size relative to the leaf element in which they lie. Consequently, the adaptive resolution of the FCM mesh carries over to the discretization of the interface.

An extension to this approach generates high-order segments using Lagrange interpolation polynomials with degree p_{seg}, by adding $(p_{\text{seg}} - 1)$ interpolation nodes within the linear segments. The implicit geometry description was used in conjunction with a line-search method to correctly place the additional interpolation points on the interface. This approach finally yields a high-order boundary discretization, as shown in Figure 6b (note that the figure shows only the extreme nodes). For

Journal publication: https://doi.org/10.1002/cnm.2951

a more detailed description of this high-order interface recovery algorithm, the reader is referred to [61]. This convenient approach for embedded boundary parametrization highlights another advantage of using Cartesian grids for the finite cell meshes and the same mesh parameters for both domains - in contrast to using unstructured FCM meshes, where the generation of cell-conforming boundary segments would be more challenging.



(a) Linear segments (straight), $p_{\text{seg}} = 1$

(b) Quadratic segments (curved), $p_{\text{seg}} = 2$

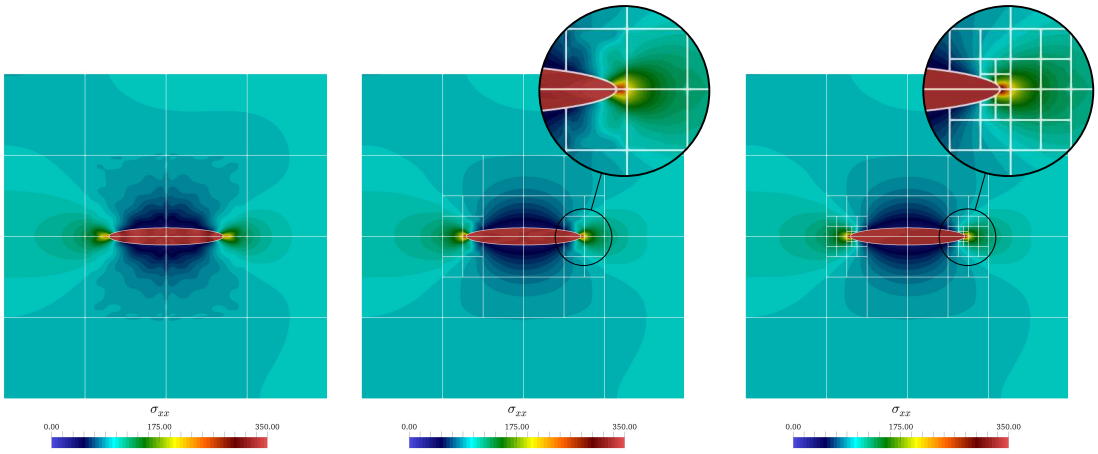
Figure 6: Discretization of the elliptical interface, $n_g = 4$, $n_{\text{seg}} = 72$

4.1.3 Numerical solution

Figure 7 shows the numerical approximation of the axial stresses σ_{xx} obtained using the FCM with weak coupling and multi-level *hp*-refinement. The plots shown here are the superposition of the solutions from the physical part of each domain, whereas the solution in the fictitious domain is omitted. Without applying any local refinement (Figure 7a), the FCM with weak coupling is able to reproduce the overall solution characteristics. However, the approximation for the stresses exhibits severe oscillations. In particular, the four finite cells in the plate's mesh that are intersected by the interface are affected by the high curvature of the concave geometry at points A and C, leading to stress concentration. This local solution characteristic cannot be represented well by the smooth basis functions of the coarse mesh, which leads to an oscillatory approximation of the stresses. Figure 7b depicts the numerical approximation obtained with the multi-level *hp*-scheme, using two levels of overlay meshes with uniform *p*-distribution. Applying local refinement to the FCM mesh of the plate improves the numerical approximation significantly. The solution benefits from the local support of the high-order basis functions with C^0-continuity, and the oscillations in the stresses are confined to the cells on the finest overlay mesh. Increasing the refinement depth with $k = 4$, the quality of the approximation improves further.

Figure 8 examines the stresses along the cutting line B-D (see Figure 4a) in closer detail. Since the cutting line is orthogonal to the ellipse's boundary, the exact solution for σ_{xx} is continuous along the cutting line. Hence, jumps in the numerical approximation of the stresses, either at the interface or at element boundaries, correspond to the approximation error directly. The numerical approximation obtained without local refinement exhibits severe oscillations and a large jump at the interface. Due to the coupling condition, the oscillations extend into both domains. Due

Journal publication: https://doi.org/10.1002/cnm.2951



(a) $p = 8$, no refinement (b) $p = 4, k = 2$ (c) $p = 4, k = 4$

Figure 7: Plate with elliptical inclusion: numerical approximation of the axial stresses σ_{xx}

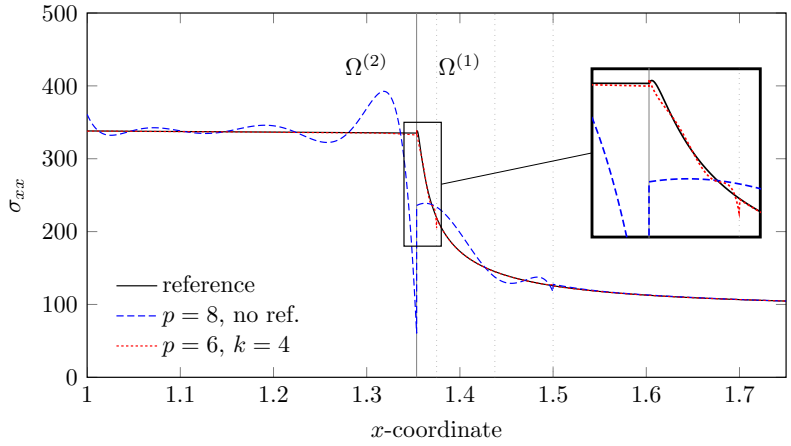


Figure 8: Plate with elliptical inclusion: numerical approximation of the axial stresses σ_{xx} along cutting line B-D

to the oscillations, the maximum stresses are overestimated and the local stress distribution is misrepresented. Applying multi-level *hp*-refinement to the plate's FCM mesh enhances the quality of the approximation appreciably, as the oscillations are mainly restricted to the finite cells on the finest overlay mesh. The zoomed plot at the interface reveals the fine-scale features of the solution in the plate. The local solution characteristics are not fully resolved by the refined mesh with $k = 4$, which suggests that this problem would benefit from further local refinement.

4.1.4 Influence of domain integral accuracy

To assess the accuracy of the numerical integration for the domain integral, Figure 9 compares the error in σ_{xx} along cutting line B-D, based on multi-level *hp*-refinement of the plate's mesh with $k = 4$, polynomial degree $p = 6$, and different domain partitioning schemes. The error here is calculated with respect to the reference solution generated using the overkill p-FEM discretization. The blended partitioning approach gives a highly accurate evaluation of the domain integrals, as

Journal publication: https://doi.org/10.1002/cnm.2951

it uses the exact geometric description of the interface. Using the quadtree partitioning scheme with a depth $m = 2$ leads to high approximation errors. The error is highest in the finite cell from inclusion's mesh, which is attributed to the coarser spatial resolution of the integration, as demonstrated by Figure 5a. The high error propagates through the mesh, carrying over to finite cells which are not cut by the material interface, and to the other mesh through the coupling condition. Increasing the depth m of the quadtree reduces the approximation error appreciably. Using $m = p + 1 = 7$, the achieved accuracy is comparable to that of the blended partitioning approach.

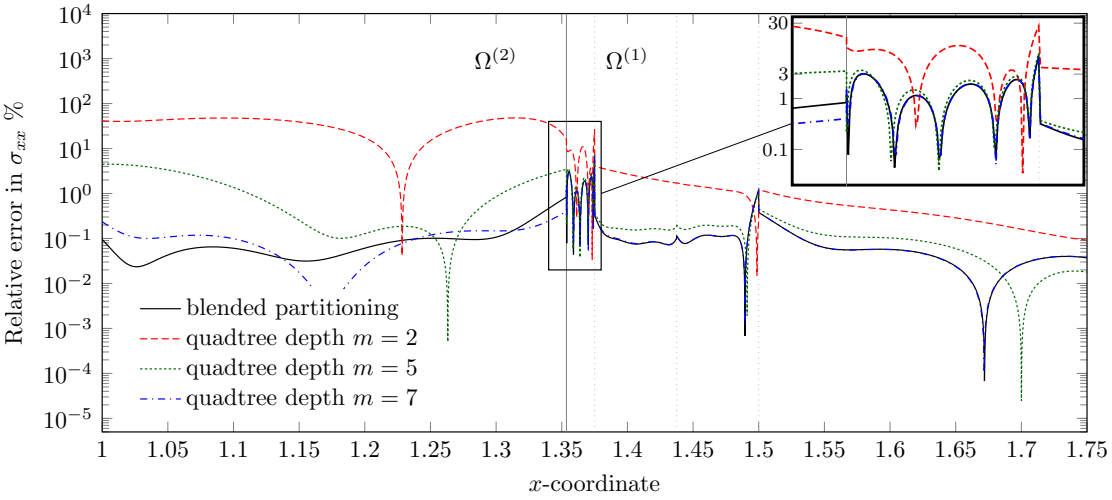


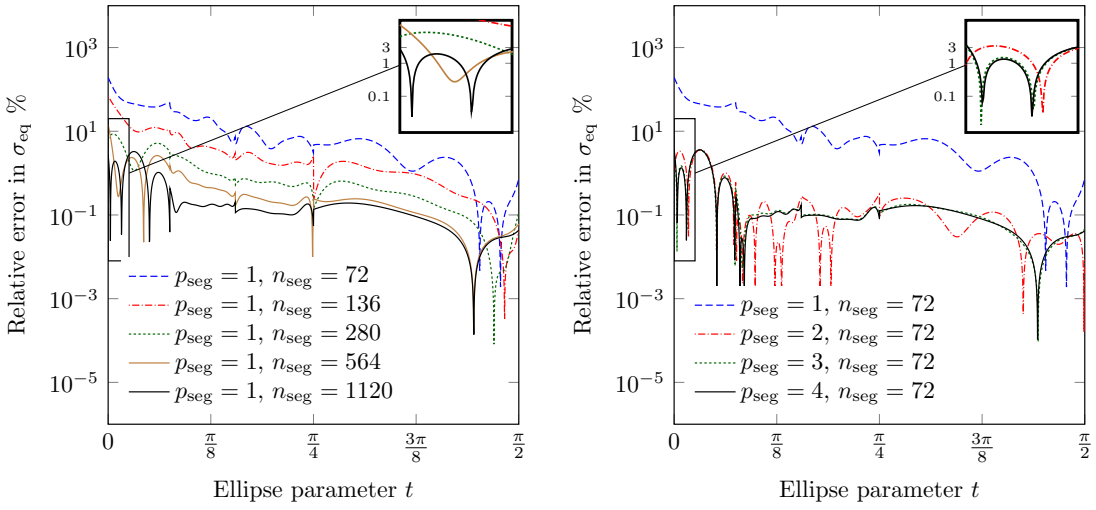
Figure 9: Error in σ_{xx} along cutting line B-D with different domain integration schemes, $p = 6$, $k = 4$

4.1.5 Influence of interface integral accuracy

To evaluate the effect of the integration accuracy of the interface integral, Figure 10 plots the local distribution of the approximation error for the von-Mises stresses σ_{eq} along the interface in the plate. Using a grid dimension of $n_g = 4$ for the marching squares implementation, $n_{\text{seg}} = 72$ linear segments are generated (see Figure 6a). However, this linear discretization of the curved interface introduces artificial stress singularities at the interpolation nodes between the segments, because only G^0-continuity is possible—where the segments touch at the extreme nodes but have discontinuous normals. At ellipse parameter $t = 0$ (point C), the artificial singularity takes over the approximation error, causing the entire solution to diverge. To reduce the error at the interface, h-refinement of the interface's discretization is carried out by refining the grid for the marching squares. Figure 10a shows the resulting error distribution with $n_g = \{4, 8, 16, 32, 64\}$, which yields $n_{\text{seg}} = \{72, 136, 280, 564, 1120\}$. The h-refinement of the segments delivers a uniform decrease in the approximation error, with a maximum error in σ_{eq} of approximately 3% using 1120 segments.

Alternatively, p-refinement of the interface segments using Lagrange polynomials, as described earlier, leads to a faster convergence as demonstrated by Figure 10b. Using 72 quadratic segments already decreases the error significantly, as the high-order discretization approximates the tangents at the interpolation nodes with better accuracy, as shown in Figure 6b. Increasing p_{seg} further from 3 to 4 in this example, gives a marginal improvement in accuracy, indicating that the numerical integral at the interface converges. The remaining approximation error is attributed to the FCM discretization with $p = 8$, and $k = 6$. The two refinement schemes for the interface discretization converge to the same local error distribution.

Journal publication: https://doi.org/10.1002/cnm.2951



(a) h-refinement (b) p-refinement

Figure 10: Error in σ_{eq} along the interface using different interface discretizations, $p = 8$, $k = 4$

4.1.6 Convergence study

A uniform p-refinement study was carried out to assess the overall convergence behavior of multi-level hp-refinement in conjunction with weak coupling. Figure 11a shows the relative error in the energy norm for different refinement depths

$$||e||_E = \sqrt{\frac{|\mathcal{U}_{\text{num}} - \mathcal{U}_{\text{ex}}|}{\mathcal{U}_{\text{ex}}}} \cdot 100\%, \tag{15}$$

where $\mathcal{U}_{\text{num}}$ denotes the numerically approximated strain energy, and $\mathcal{U}_{\text{ex}}$ denotes the reference strain energy. For this study, the blended approach is used for the domain integration, in conjunction with a high-order parametrization of the interface. As the exact solution is piecewise-analytic, FCM with a weak enforcement of the interface conditions converges exponentially under uniform p-elevation, even without applying any local refinement. For a more detailed analysis of the convergence rates, Figure 11b shows the convergence plot in a log-$\sqrt{\ }$ scaling. Here, the linear plots characterize exponential convergence in the form [62]:

$$||e||_E \leq C \exp\left(\gamma N^{\theta}\right) \quad \text{with } \theta = \frac{1}{2}, \tag{16}$$

where C is a positive constant, and γ is a negative constant describing the convergence rate. Increasing the refinement depth with the multi-level hp-scheme for this example leads to a steeper convergence rate γ, allowing for a higher gain in accuracy. To achieve an engineering accuracy of 1% error, the discretization with $k = 4$ needs approximately five times less the degrees of freedom than the non-refined case.

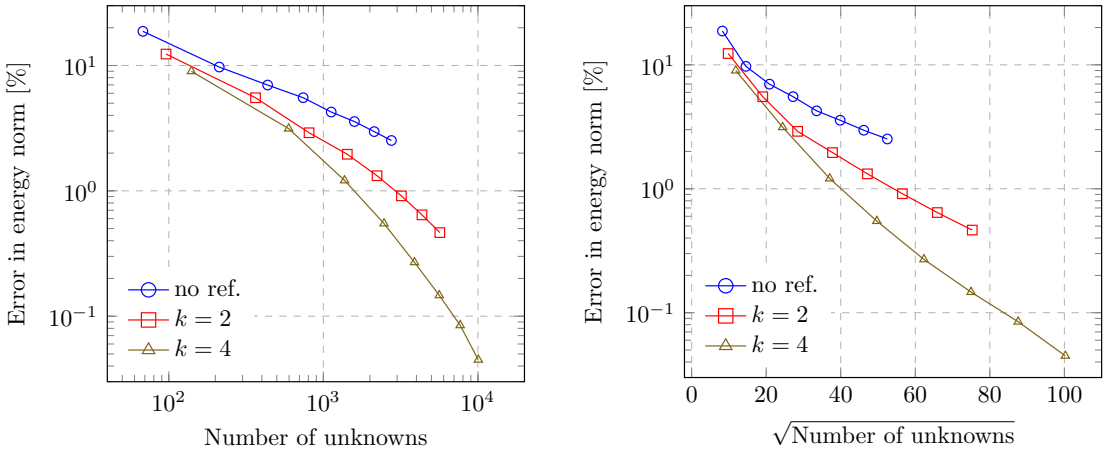
4.2 Bi-material inclusion corner

In the next study, we consider a singular material-interface benchmark problem adapted from [63]. The two-dimensional temperature problem is governed by the Poisson equation:

$$\kappa^{(i)} \nabla^2 \phi^{(i)} = -1 \qquad \forall \boldsymbol{x} \in \Omega^{(i)} \tag{17}$$

$$\phi = 0 \qquad \forall \boldsymbol{x} \in \Gamma_D, \tag{18}$$

Journal publication: https://doi.org/10.1002/cnm.2951

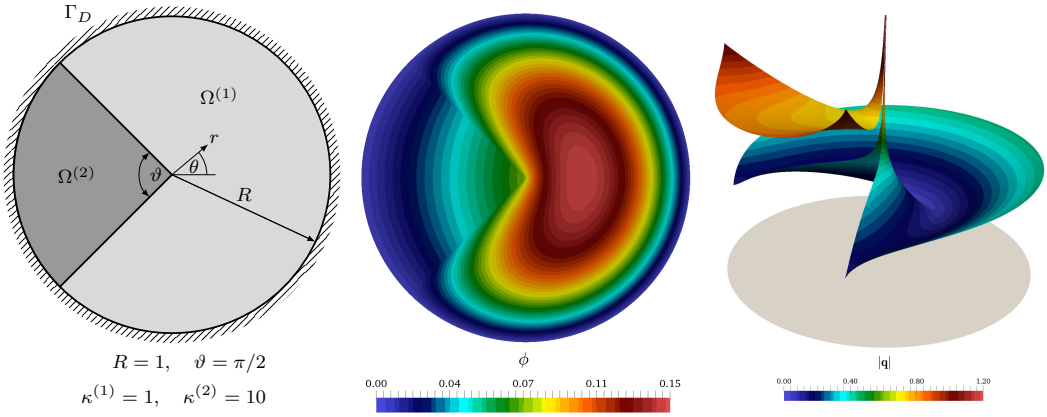


(a) Convergence in log-log scale

(b) Convergence in semi-log scale

Figure 11: Plate with elliptical inclusion: p-convergence for multi-level hp-refinement

where $\phi^{(i)}$ denotes the temperature, $\kappa^{(i)}$ the thermal diffusivity, $\Omega^{(i)}$ and Γ_D are defined as shown in Figure 12a. Here, the material interface Γ_{12} has a sharp corner, inducing a vertex singularity. Moreover, the intersection of the material interface with the Dirichlet boundary Γ_D for the applied boundary condition introduces two additional weak singularities, where the solution exhibits reduced continuity.



(a) Benchmark setup

(b) Reference temperature ϕ

(c) Reference heat flux $|\boldsymbol{q}|$

Figure 12: Bi-material inclusion corner

The exact solution to this problem is given in radial coordinates (r, θ) by [64] :

$$\phi(r, \theta) = A_1 r^{\lambda_1} h_1(\theta) + A_2 r^{\lambda_2} h_2(\theta) + \mathcal{O}(r^2), \tag{19}$$

where A_1 and A_2 are scalar constants, $h_1(\theta)$ and $h_2(\theta)$ are smooth sinusoidal functions and

$$\lambda_1 = 0.731691779, \quad \lambda_2 = 1.268308221. \tag{20}$$

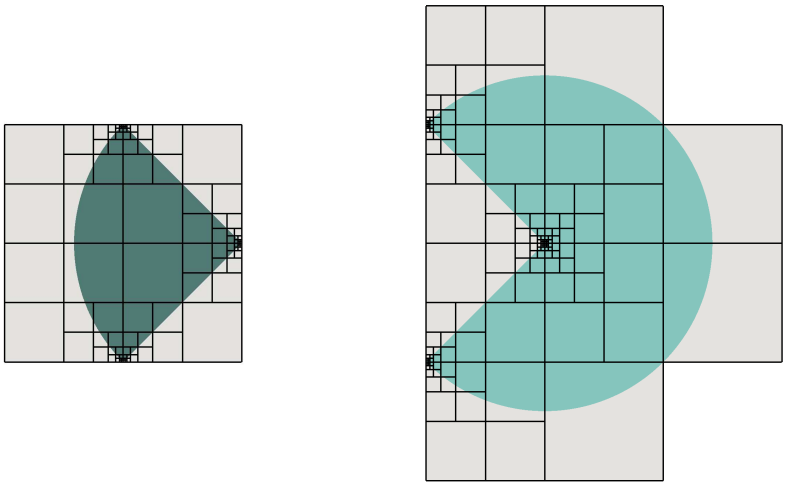
For reference, an overkill solution was generated using a conforming p-FEM with exact blending, where the mesh was geometrically graded towards the vertex singularity. The reference mesh has

Journal publication: https://doi.org/10.1002/cnm.2951

21 elements with polynomial degree $p = 30$ and 18,991 degrees of freedom. The strain energy for the setup shown here is

$$\mathcal{U}_{\text{ex}} = 1.0168443145 \times 10^{-1}. \tag{21}$$

Using FCM with weak coupling, separate meshes were used for the disc and the inclusion, as shown in Figure 13. Unlike the previous example, where the stress concentration in the plate was caused by the concave geometry, the singularities affect the solution in both sub-domains. Consequently, *both meshes* need local refinement to resolve the singularities. The meshes were graded towards the vertex singularity, and the two weak singularities on the boundary. The same refinement depth was applied to all three refinement points. Here, the discretizations match the singularities with nodes of the FCM mesh.



(a) Inclusion (b) Disc

Figure 13: Bi-material inclusion corner: discretization with refinement depth $k = 6$

The penalty method was used to couple the FCM meshes along the straight material interface. Additionally, the embedded Dirichlet boundary condition was weakly enforced on $\Gamma_D^{(i)}$ in each mesh. The fictitious domain was penalized with $\kappa_{\text{fict}} = \kappa^{(1)} \cdot 10^{-9}$. The blended partitioning approach was used to numerically evaluate the domain integral. The coupling integral was evaluated on linear segments, whereas the weak Dirichlet boundary conditions were integrated over curved segments, automatically generated with the same approach described for the previous example.

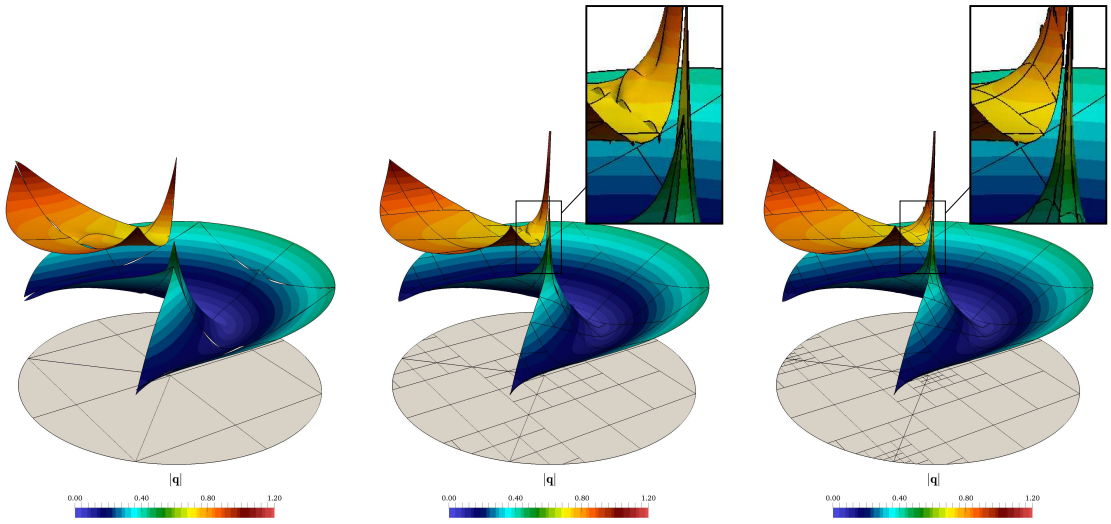
The magnitude of numerical solution for the heat flux, $\boldsymbol{q}^{(i)} = \kappa^{(i)} \nabla \phi^{(i)}$, is depicted in Figure 14. For the first case, without any local refinement of the meshes, the numerical approximations of the heat fluxes exhibit oscillations within the domain and jumps at element boundaries. The oscillatory behavior is attributed to the inability of the smooth polynomials to represent the high gradients in the vicinity of the singularities. For the second discretization, two levels of uniform high-order overlay meshes are used to grade the meshes towards the three singularities.The C^0-continuity of the high-order basis functions limits the pollution error from the singularities at the finest refinement level, as revealed in the zoomed plot. Using four refinement levels improves the solution further, as the geometric progression of element size arrests the propagation of the error, while having larger elements to describe the smooth solution away from the singularities.

Figure 15a shows the results of a p-elevation study carried out to assess the convergence properties. Without applying local refinement, the convergence under p-elevation is identified as being algebraic in the form

$$||e||_E \leq aN^{\eta}, \tag{22}$$

where a is a positive constant, and η is a negative constant indicating the convergence rate. The

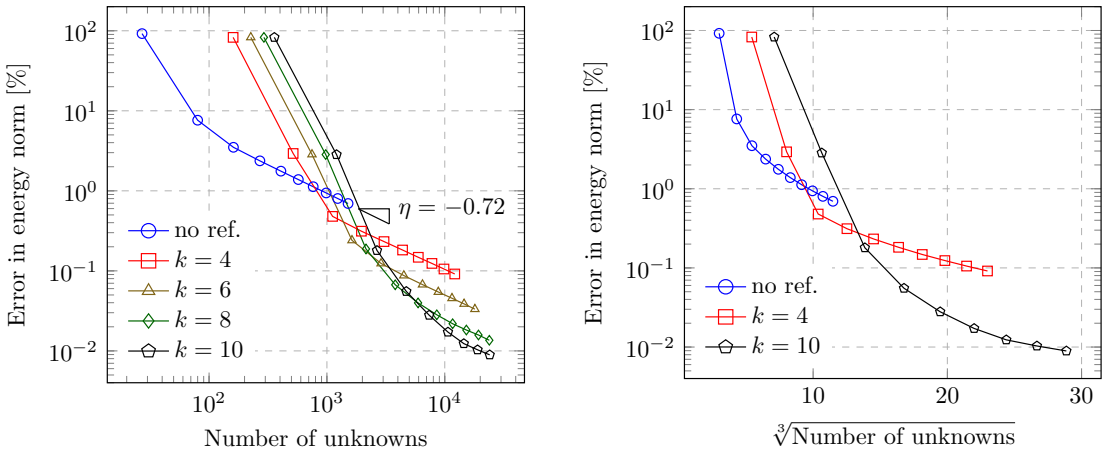
Journal publication: https://doi.org/10.1002/cnm.2951



(a) $p = 4$, no refinement (b) $p = 4, k = 2$ (c) $p = 4, k = 4$

Figure 14: Bi-material inclusion corner: numerical approximation of the flux $|\boldsymbol{q}|$

estimate for $\eta \approx -\lambda_1$ matches the theoretical expectation for a discretization where the vertex singularities are matched by nodes of the mesh [65]. With multi-level *hp*-refinement the convergence behavior shows a pre-asymptotic range with a higher convergence rate, and an asymptotic range, where the behavior returns to being algebraic with the same rate η. Increasing the refinement depth k extends the pre-asymptotic range, allowing for a significant decrease in the total approximation error. Changing the scaling to a log-$\sqrt[3]{\ }$ in Figure 15b, the extended pre-asymptotic range appears linear, indicating exponential convergence in the pre-asymptotic range.



(a) Convergence in log-log scale (b) Convergence in semi-log scale

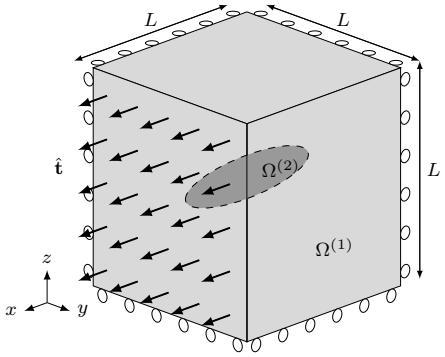
Figure 15: Bi-material inclusion corner: *p*-convergence for multi-level *hp*-refinement

Note the apparent leveling-off of the error in the energy norm around $10^{-2}\%$, which is attributed to the use of $\kappa_{\text{fict}} \neq 0$, adding a modeling error to the numerical approximation. Using a smaller value for κ_{fict} lowers the leveling-off threshold [66], at the cost of the conditioning of the system.

Journal publication: https://doi.org/10.1002/cnm.2951

4.3 Cube with ellipsoidal inclusion

Next, we consider a three-dimensional example, which is a generalization of the benchmark presented in Section 4.1. A linear elastic analysis is considered for a cube with an embedded ellipsoidal (prolate spheroid) inclusion - depicted in Figure 16. The spheroidal inclusion induces stress concentration in the cube around the two points on its major axis. The solution within the spheroidal inclusion remains smooth.



$$E^{(1)} = 210 \cdot 10^3, \quad E^{(2)} = E^{(1)} \cdot 10,$$
$$\nu = 0.3, \quad |\hat{\mathbf{t}}| = 100, \quad L = 1,$$
$$r_x = \sqrt{0.125}, \quad r_y = r_z = 0.25 \cdot r_x$$

Figure 16: Cube with ellipsoidal inclusion

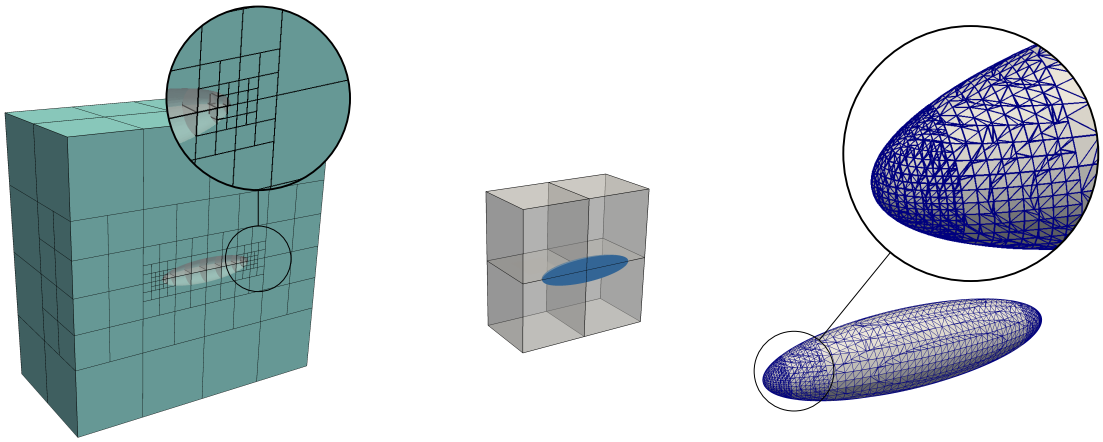
Two FCM meshes were used to discretize the sub-domains, using a base discretization of $4 \times 4 \times 4$ finite cells which match the outer boundaries of the cube. To resolve the stress concentration, the cube's mesh was refined towards the two points on the major axis of the spheroid, as shown in Figure 17. For the inclusion's mesh, $2 \times 2 \times 2$ elements remain after excluding all cells that are completely in the fictitious domain. Finite cells that are cut by the interface were further partitioned for integration using the octree approach described earlier.

The penalty coupling terms were integrated over a surface triangulation, automatically generated using marching cubes [60]. The implementation is a three-dimensional extension of the high-order parametrization algorithm used in Section 4.1, where a grid of $n_g \times n_g \times n_g$ points per sub-cell was used for the surface recovery. An example of the resulting triangulation is shown in Figure 17. Note that the surface mesh is irregular and that the triangles have severe aspect ratios. The high-order penalty terms are then integrated over the triangles by collapsing one side of the integration domain (bi-unit square) and mapping it to each triangle. As demonstrated by the two-dimensional ellipse benchmark the linear parametrization needs a fine resolution to achieve accurate results.

The numerical approximation for the axial stresses σ_{xx} is depicted in Figure 18. Note that half of the domain is removed to visualize the internal solution. For the unrefined meshes, the approximated stresses exhibit oscillations and non-physical jumps at the element boundaries. If the cube's mesh, which is affected by the concave geometry, is refined towards the two points, the stress concentration can be localized quickly. Similar to the two-dimensional case, the solution within the ellipsoid remains smooth, and does not require local mesh refinement.

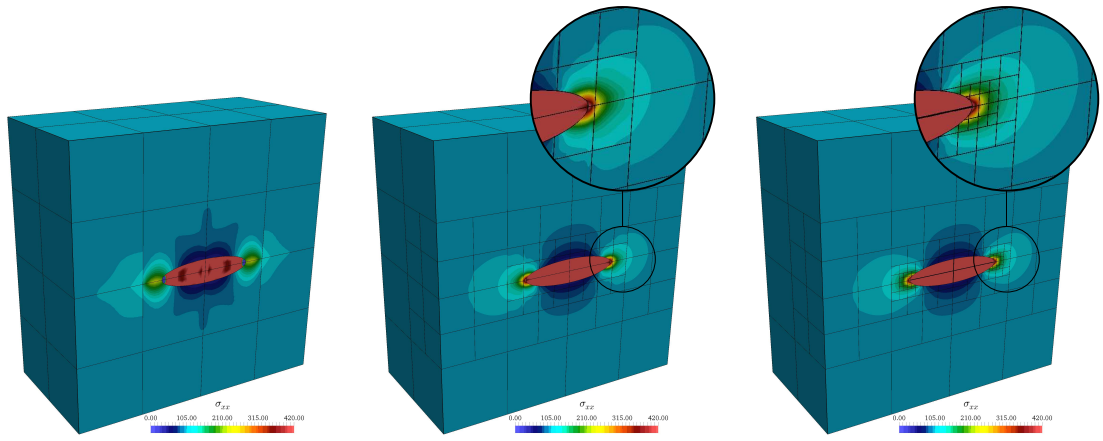
The results show that the improvement in accuracy brought on by the multi-level *hp*-refinement scheme also carries over to three-dimensional problems. The example also demonstrates that a surface mesh for the interface discretization is easily obtained for smooth implicit geometries. The surface mesh is only used to integrate the penalty terms, and hence can be irregular or have severe aspect ratios, unlike what is required of an analysis-suitable finite element mesh.

Journal publication: https://doi.org/10.1002/cnm.2951



(a) Cube (b) Ellipsoidal inclusion (c) Coupling surface

Figure 17: Discretization of cube and ellipsoid using $k = 4$ - Section view



(a) $p = 8$, no refinement (b) $p = 6, k = 2$ (c) $p = 6, k = 4$

Figure 18: Cube with ellipsoidal inclusion: numerical approximation of axial stresses

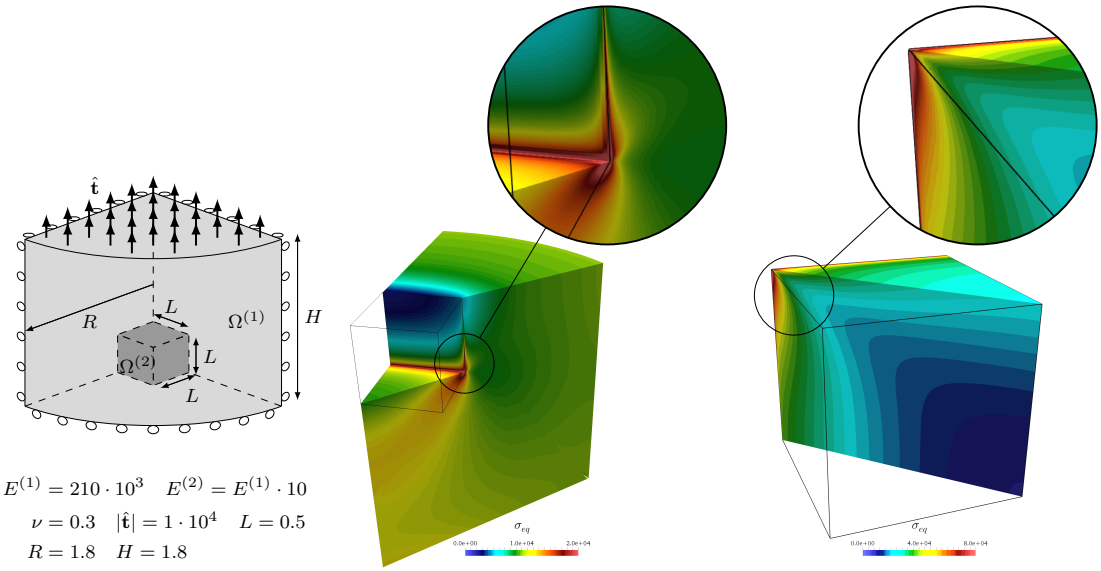
4.4 Cylinder with cubical inclusion

The last numerical benchmark for us to consider is a three-dimensional problem with a vertex-edge singularity. We study a cylinder under axial tension, which has an embedded cubical inclusion. Due to the symmetry of the problem, we consider only one-eighth of the system. An overkill solution was generated for reference, using seven conforming hexahedral p-FEM elements with exact blending, and a local refinement using the multi-level hp-scheme. Using 5 levels of refinement and a polynomial degree $p = 9$, the overkill discretization has $506,199$ degrees of freedom. The reference strain energy for the setup shown in Figure 19a is

$$\mathcal{U}_{\text{ex}} = 1.037455 \times 10^3. \tag{23}$$

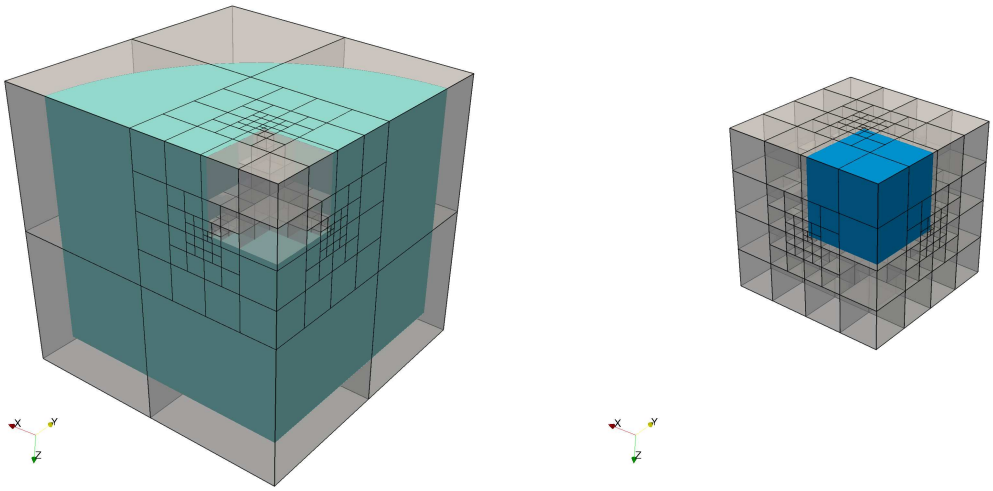
Two non-geometry-conforming FCM meshes were used for the cylinder and the cubical inclusion. A coarse base discretization of $2 \times 2 \times 2$ finite cells was used. The dimensions of the mesh were setup

Journal publication: https://doi.org/10.1002/cnm.2951



(a) Benchmark setup (b) Reference σ_{eq} in cylinder (c) Reference σ_{eq} in inclusion

Figure 19: Cylinder with cubical inclusion



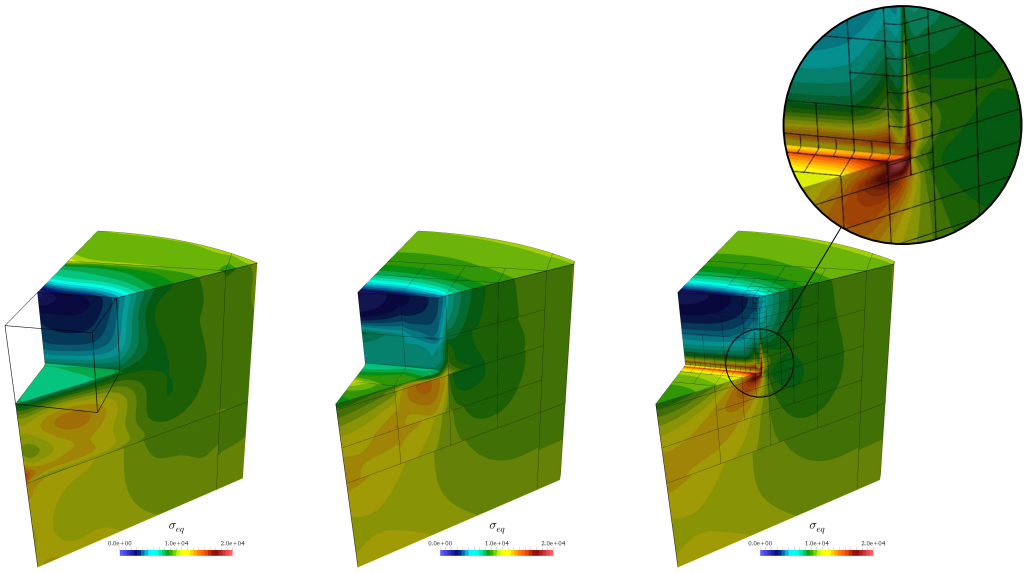
(a) Cylinder (b) Cubical inclusion

Figure 20: Discretization of cylinder and cubical inclusion, $k = 5$

in such a way that the edges of the FCM meshes do *not* exactly coincide with the cube's edges. Similar to the two-dimensional benchmark presented in Section 4.2, the vertex-edge singularity affects the solution in both domains. Accordingly, both meshes were refined towards the singular edges, as shown in Figure 20. The symmetry boundary conditions were enforced in a classical manner on the corresponding faces of the FCM meshes.

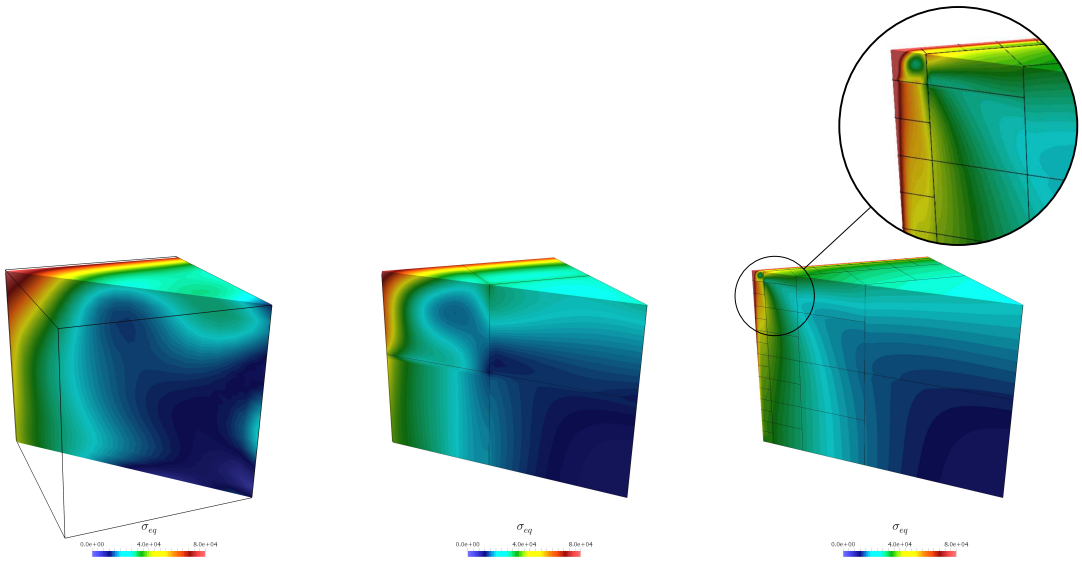
The numerical approximation for the von-Mises stresses, σ_{eq}, is shown in Figure 21 for the cylinder and in Figure 22 for the cubical inclusion. The complete domain is clipped diagonally to reveal the solution at the vertex-edge singularity. Note that the geometry of the cylinder is rotated

Journal publication: https://doi.org/10.1002/cnm.2951



(a) $p = 8$, no refinement (b) $p = 4, k = 2$ (c) $p = 4, k = 5$

Figure 21: Cylinder: numerical approximation of σ_{eq}



(a) $p = 8$, no refinement (b) $p = 4, k = 2$ (c) $p = 4, k = 5$

Figure 22: Cubical inclusion: numerical approximation of σ_{eq}

to show the solution along the singular edges. Whereas the solution obtained without refinement is highly oscillatory, the local refinement with the multi-level *hp*-scheme is able to confine the error to the finest level of sub-cells.

To assess the convergence behavior, a *p*-elevation study was carried out. The results, shown in Figure 23, indicate a slow algebraic rate for coupled FCM without refinement. Applying a local refinement starting with $k > 2$, a pre-asymptotic range can be identified as having a steeper

Journal publication: https://doi.org/10.1002/cnm.2951

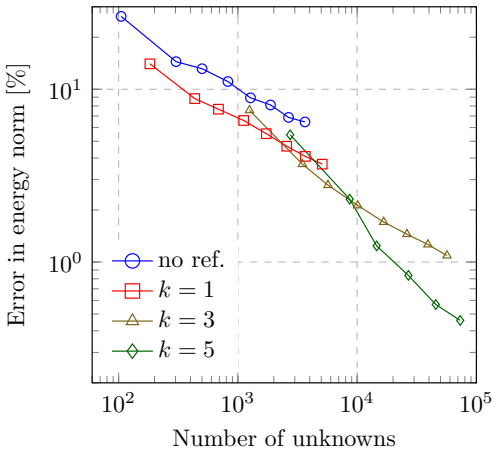


Figure 23: Cylinder with cubical inclusion: p-convergence for multi-level hp-refinement

convergence rate. This allows the discretization with $k = 5$ to achieve an engineering accuracy of 1% error using a moderate polynomial order. This would require at least an order of magnitude more degrees of freedom with uniform h- or p-refinement. The pre-asymptotic convergence is also characterized as algebraic, albeit with a higher rate. For vertex-edge singularities, an anisotropic refinement scheme is necessary to attain exponential convergence [65], which was not applied in this work. However, this is not always feasible for the general case of embedded vertex-edge singularities, where the (possibly curved) singular edges might not be parallel to edges of the mesh. Nevertheless, the improved algebraic convergence brought on by the multi-level hp-refinement allows for a significant reduction of the approximation error.

5 Application in biomechanics: vertebra with pedicle screws

Having investigated the convergence properties of the proposed refinement scheme for benchmark problems with stress concentration, we demonstrate the applicability of the refinement scheme for large-scale problems with complex three-dimensional geometries.

5.1 Problem setup

We consider a lower thoracic vertebra into which two cannulated pedicle screws are inserted. This is done as part of a procedure to stabilize a spinal segment of by fusing two vertebrae. We assume that the vertebra-screw interface is in full body contact, and model the contact as a material interface. We use the FCM with the weak coupling approach to solve the interface problem, and employ multi-level hp-refinement to resolve the high gradients around the screws' threads.

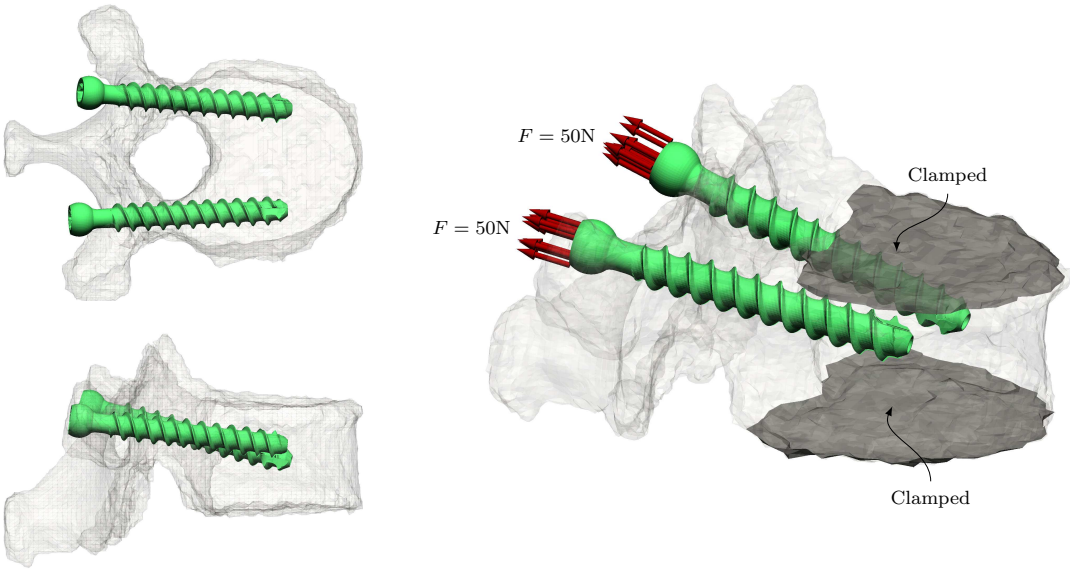
The geometry of the vertebra originates from a CT scan with high spatial resolution, carried out for a formalin-fixated specimen of the eleventh thoracic vertebra of an 84-year-old donor. She had dedicated her body for educational and research purposes to the local Institute of Anatomy prior to death, in compliance with the local institutional and legislative requirements. Images were acquired by using a whole-body 256-row CT scanner (iCT, Philips Medical Care, Best, The Netherlands) after 24 hours of degassing. Scan parameters were a tube voltage of 120 kVp, a tube load of 585 mAs, an image matrix of 1024×1024 pixels, and a field of view of 150 mm. Transverse sections were reconstructed with an interpolated voxel size of $146 \times 146 \times 146$ μm^3, and the intensity values of the CT images were calibrated with a reference phantom (Mindways Osteoporosis Phantom, San Francisco, CA, USA) to derive calcium hydroxyapatite values in (mg/cm^3).

The geometric model of the screws is a B-Rep CAD model of a Viper2-Screw (DePuy Synthes, Umkirch, Germany), provided by the manufacturer.

Journal publication: https://doi.org/10.1002/cnm.2951

The material properties for the vertebra were resolved on the voxel level, as a threshold was set to the intensity values to distinguish the trabecular structure. The voxels that are identified as inside the bone were assigned a Young's modulus $E^{(1)} = 10$ GPa, and Poisson ratio $\nu = 0.3$, which are commonly used parameters in HR-pQCT based voxel FEM [67]. The material of the screws is titanium with $E^{(2)} = 100$ GPa and $\nu = 0.3$.

The considered load case simulates a pull-out test, where the vertebral body is clamped and the screws are pulled axially outwards—as depicted in Figure 24. The Dirichlet boundary conditions are defined as clamping of superior and inferior end-plates, whereas the Neumann boundary condition is defined on the screws head, as a uniformly distributed axial force.



(a) Geometry (b) Boundary conditions

Figure 24: Pull-out test

5.2 Discretization

5.2.1 *hp*-FCM

The complete model of the vertebra-screw system was setup virtually by combining both geometric models as described in the following. First, the CT-scan was segmented using `ITK-SNAP` [68] to obtain a separate model of the vertebral body without the surrounding tissue. This segmentation only considers the outer boundary of the vertebra, and does not distinguish between cortical and trabecular bone.

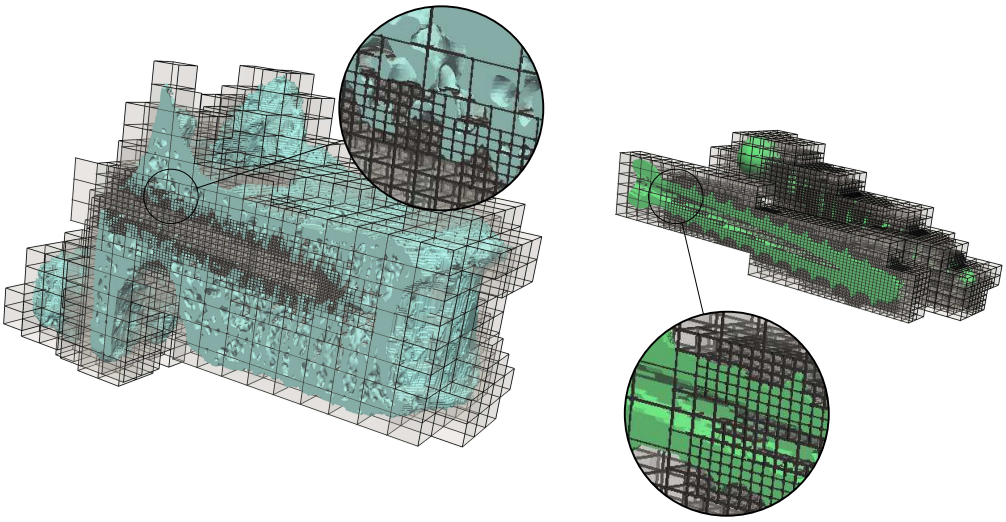
The CAD model of the screws was suitably positioned in the same coordinate space. To carry out the inside-outside test on the CAD model, a watertight surface triangulation of the screws was created using `Rhinoceros` [69]. The surface triangulation was then utilized by a ray intersection algorithm [70], to determine whether a given integration point lies within the screws' geometry.

The first FCM mesh is used to discretize the embedding domain of the vertebral body. The mesh is axis-aligned with the CT-scan. The geometry is defined by applying a threshold to the CT-scan to define the bone-structure and subtracting the screws through a Boolean operation. To account for the canals within the screws, an additional cylinder at the central axis of each screw was subtracted. The numerical evaluation of the domain integrals performs an intersection of the voxel grid with the refined FCM mesh to give the integration sub-cells.

Journal publication: https://doi.org/10.1002/cnm.2951

To resolve the stress concentration, the mesh is refined using the multi-level *hp*-scheme towards the cusps of the screws' threads (concave geometry) and towards the reentrant corner at the pedicle. The second FCM mesh is used to discretize the embedding domain of the screws. This mesh is refined towards the roots of the screws' threads, where stress concentration within the screws is expected. The octree scheme is used for the numerical integration on the second mesh.

The complete discretization results in a total of $1,322$ finite cells for first mesh and 155 finite cells for the second mesh. Using polynomial degree $p = 3$ and $k = 3$ refinement levels, the discretization has approximately 1.13 million degrees of freedom. The final refined meshes are depicted in Figure 25. To avoid ill-conditioning of the stiffness matrices, the fictitious domain in both meshes is given a stiffness of $E_{\text{fict}} = 10^{-4}$ GPa and $\nu_{\text{fict}} = 0.3$.



(a) Vertebra (b) Screws

Figure 25: Discretization using FCM

5.2.2 Material interface

The penalty approach was used to couple the meshes with $\beta_{12} = 10^9$. To integrate the coupling terms, a fine surface triangulations for the bone-screw interfaces was created using `Rhinoceros`, by meshing the outer surfaces of the screws. For an accurate and efficient evaluation of the surface integral, the triangulation was intersected with the refined finite cell grid, as shown in Figure 26. The triangles were then associated to the finite cell pairs form the two meshes, such that the coupling terms were added to the assembled stiffness matrices only once per finite cell pair.

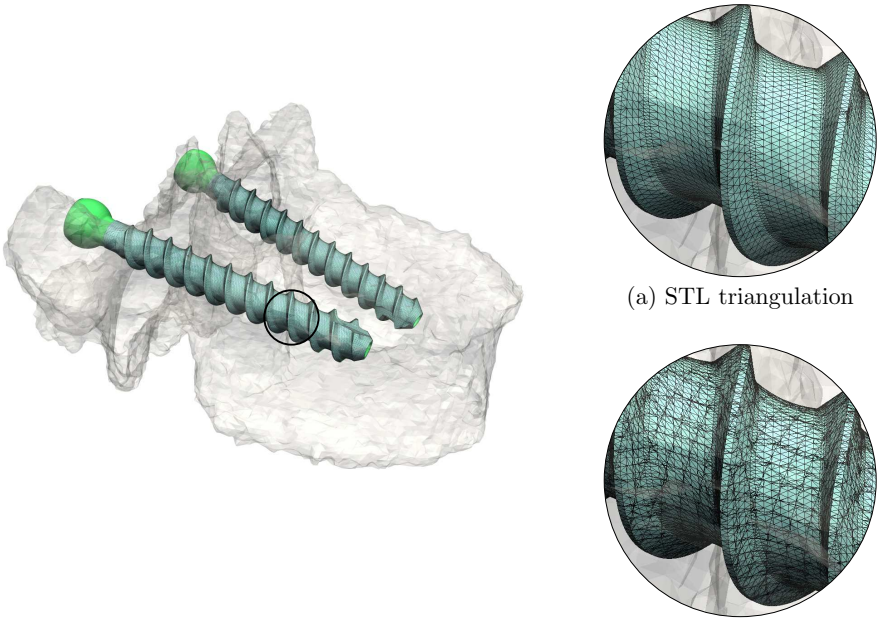
5.2.3 Boundary conditions

Dirichlet boundary conditions were applied at the superior and inferior end-plates of the vertebra. These were weakly enforced using the penalty method, with the penalty terms integrated over a surface triangulation. The surface description was generated using the marching cubes algorithm and the segmentation. The penalty value was selected empirically, $\beta_D = 10^7$. The Neumann boundary conditions modeling the axial load on the screws were applied on a surface mesh created from the CAD model. The surface mesh was also intersected with the faces of the refined FCM mesh, to allow for a more accurate evaluation of the Neumann integral.

5.3 Solution and numerical results

The resulting system of linear equations was solved using the parallel direct solver `Intel`® `Pardiso` which is provided as part of the Intel Math Kernel Library [71]. The simulation was run on two

Journal publication: https://doi.org/10.1002/cnm.2951



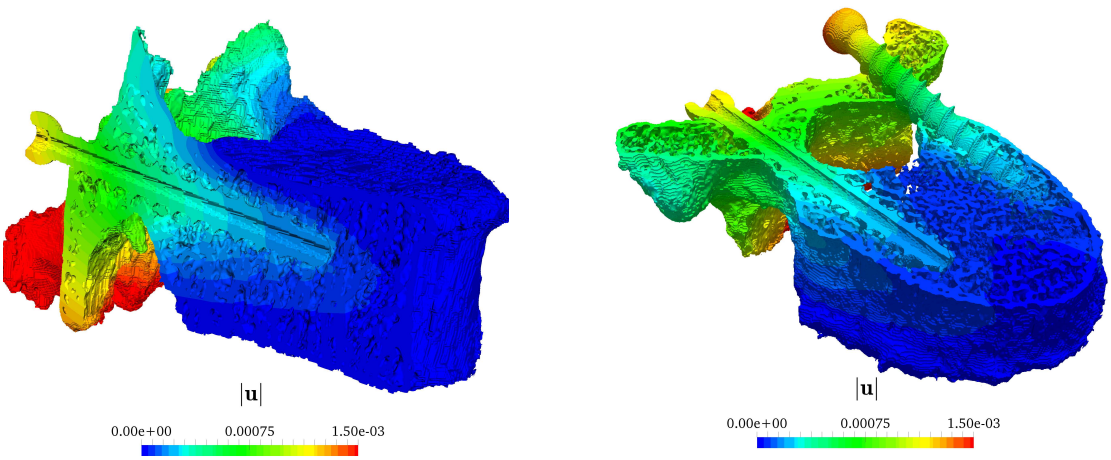
(a) STL triangulation

(b) Intersected triangulation

Figure 26: Discretization of coupling surface

eight-core Intel® Xeon® E5-2690 @ 2.9 GHz CPUs. The model takes approximately 40 minutes for the integration, solution and post-processing. Visualization was carried out using `ParaView` [72] on the same hardware.

The surface of the trabecular bone was recovered using `ParaView`, for an axial and a sagittal section. The recovered surface was used for the plots shown here. The computed displacements are depicted in Figure 27. It can be observed that the applied boundary conditions are fulfilled, and that the displacement field appears continuous across the interface.



(a) Sagittal section

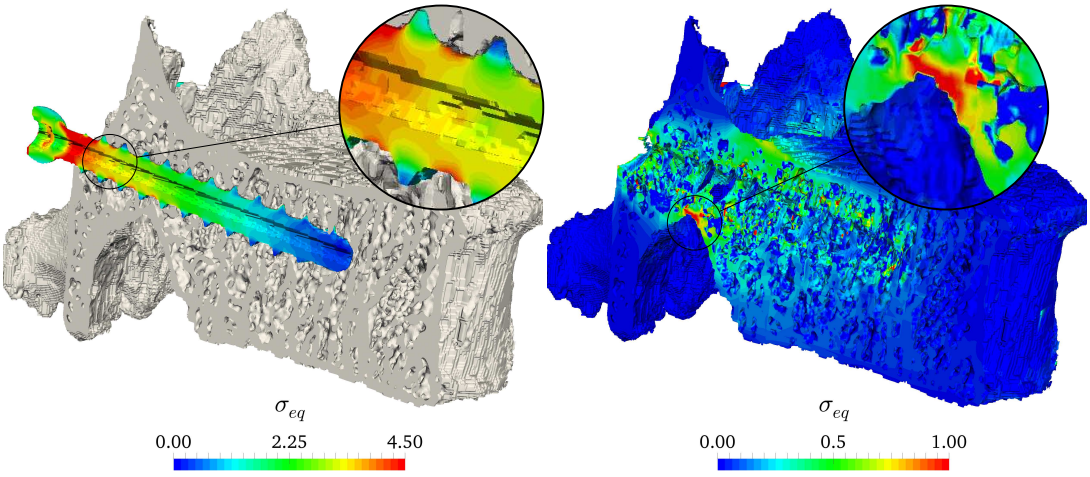
(b) Axial section

Figure 27: Vertebra with pedicle screws: numerical approximation of displacements

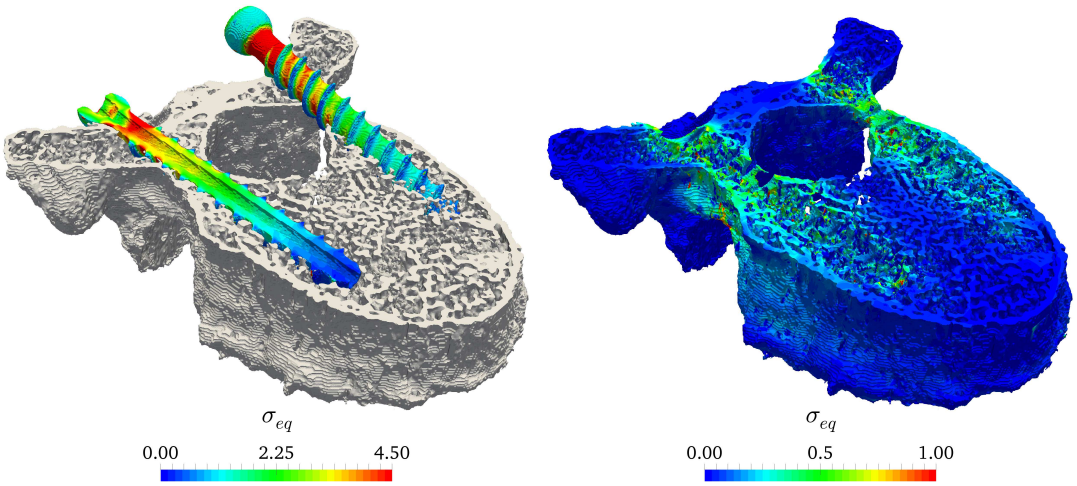
The von-Mises stresses calculated for the screws and the bones are depicted in Figure 28 for a sagittal and an axial section. The numerically approximated stresses are continuous within

Journal publication: https://doi.org/10.1002/cnm.2951

the screw's geometry, with no discernible jumps at element boundaries or large scale oscillations. This indicates a low discretization error. High stresses are visible in the neck region between the applied load and the vertebra. Additionally, a bending stress distribution is observed for the helical thread, with the maximum stress occurring at the roots of the thread. In the vertebra, the stress concentration is localized around the screws, and at the pedicle's reentrant corner. Similar stress distribution patterns were also reported for simulations using micro-FE models [73–75].



(a) Sagittal section



(b) Axial section

Figure 28: Vertebra with pedicle screws: numerical approximation of von-Mises stresses

Altogether, the numerical results are mechanically plausible, and correspond well to the applied load case. This demonstrates that the proposed refinement technique is suitable for solving problems involving complex three-dimensional geometries without the need to generate a boundary-conforming mesh. Furthermore, this application demonstrates the flexibility of the FCM in dealing with different types of geometric descriptions (CAD and image-based models), without the additional effort of re-parametrization.

Journal publication: https://doi.org/10.1002/cnm.2951

6 Conclusion and Outlook

In the present paper, we demonstrated that the use of multi-level *hp*-adaptive refinement for the finite cell method in conjunction with a weak enforcement of the interface conditions constitutes an efficient and robust approach for the solution of material interface problems involving complex geometries. The proposed combination of discretization techniques circumvents volumetric mesh generation for domains with multiple materials, while maintaining—locally—a high model resolution.

Several numerical benchmarks of two- and three-dimensional material interface problems show that the convergence behavior is significantly improved by the multi-level *hp*-refinement scheme for problems with stress concentration and singularities. The results also highlight the importance of a high-order accurate geometric description of the interface and the geometric boundaries.

In the context of biomechanical problems, we demonstrated that the proposed scheme is directly applicable for the simulation of geometrically complex implant-vertebra models. Furthermore, we demonstrated that the flexible nature of the FCM in dealing with different types of geometric description allows for an easy combination of the image-based model of the vertebra and the CAD-based model of the screws. Additionally, the accurate approximation of the mechanical stresses, which was corroborated by several numerical benchmarks, further illustrates the potential of the proposed scheme as a simulation tool for bone-implant models.

The results presented here offer several possibilities for further developments. Future work will address the experimental validation of the vertebra model. Furthermore, extending the vertebra-implant model to include several vertebral bodies would allow for more realistic boundary conditions and loading scenarios. Also, the development of a robust and efficient error estimator for FCM would open the door for automatic refinement.

Acknowledgements

The authors gratefully acknowledge the financial support of the German Research Foundation (DFG) under Grant RA 624/26-1, and of the European Research Council under Grant ERC-2014-StG 637164.

References

[1] T. J. R. Hughes, *The Finite Element Method: Linear Static and Dynamic Finite Element Analysis*. Mineola, NY: Dover Publications, 2000.

[2] K. J. Bathe, *Finite Element Procedures*. New Jersey: Prentice Hall, 2007.

[3] J. Parvizian, A. Düster, and E. Rank, "Finite cell method," *Computational Mechanics*, vol. 41, pp. 121–133, Apr. 2007.

[4] 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*, vol. 197, pp. 3768–3782, Aug. 2008.

[5] N. Zander, S. Kollmannsberger, M. Ruess, Z. Yosibash, and E. Rank, "The Finite Cell Method for linear thermoelasticity," *Computers & Mathematics with Applications*, vol. 64, pp. 3527–3541, Dec. 2012.

[6] 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*, vol. 50, pp. 445–478, Feb. 2012.

[7] M. Elhaddad, N. Zander, S. Kollmannsberger, A. Shadavakhsh, V. Nübel, and E. Rank, "Finite Cell Method: High-Order Structural Dynamics for Complex Geometries," *International Journal of Structural Stability and Dynamics*, vol. 15, p. 1540018, Apr. 2015.

[8] M. Joulaian, S. Duczek, U. Gabbert, and A. Düster, "Finite and spectral cell method for wave propagation in heterogeneous materials," *Computational Mechanics*, vol. 54, pp. 661–675, Apr. 2014.

[9] 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*, vol. 11, pp. 425–37, Mar. 2012.

[10] C. Verhoosel, G. 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*, vol. 284, pp. 138–164, Feb. 2015.

[11] N. Zander, T. Bog, M. Elhaddad, R. Espinoza, H. Hu, A. 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*, vol. 74, pp. 49–63, Aug. 2014.

Journal publication: https://doi.org/10.1002/cnm.2951

[12] 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*, vol. 200, pp. 3358–3380, Nov. 2011.

[13] M. Joulaian and A. Düster, "Local enrichment of the finite cell method for problems with material interfaces," *Computational Mechanics*, vol. 52, pp. 741–762, Oct. 2013.

[14] J. M. Melenk and I. Babuška, "The partition of unity finite element method: Basic theory and applications," *Computer Methods in Applied Mechanics and Engineering*, vol. 139, pp. 289–314, Dec. 1996.

[15] T. Strouboulis, K. Copps, and I. Babuv ska, "The generalized finite element method," *Computer methods in applied mechanics and engineering*, vol. 190, no. 32, pp. 4081–4193, 2001.

[16] T.-P. Fries and T. Belytschko, "The extended/generalized finite element method: An overview of the method and its applications," *International Journal for Numerical Methods in Engineering*, vol. 84, pp. 253–304, Oct. 2010.

[17] N. Moës, M. Cloirec, P. Cartraud, and J. F. Remacle, "A computational approach to handle complex microstructure geometries," *Computer Methods in Applied Mechanics and Engineering*, vol. 192, pp. 3163–3177, July 2003.

[18] T.-P. Fries, "A corrected XFEM approximation without problems in blending elements," *International Journal for Numerical Methods in Engineering*, vol. 75, pp. 503–532, July 2008.

[19] G. Legrain, R. Allais, and P. Cartraud, "On the use of the extended finite element method with quadtree/octree meshes," *International Journal for Numerical Methods in Engineering*, vol. 86, pp. 717–743, May 2011.

[20] G. Legrain, N. Chevaugeon, and K. Dréau, "High order X-FEM and levelsets for complex microstructures: Uncoupling geometry and approximation," *Computer Methods in Applied Mechanics and Engineering*, vol. 241-244, pp. 172–189, Oct. 2012.

[21] S. Soghrati, A. M. Aragón, C. Armando Duarte, and P. H. Geubelle, "An interface-enriched generalized FEM for problems with discontinuous gradient fields," *International Journal for Numerical Methods in Engineering*, vol. 89, pp. 991–1008, Feb. 2012.

[22] S. Soghrati and P. H. Geubelle, "A 3D interface-enriched generalized finite element method for weakly discontinuous problems with complex internal geometries," *Computer Methods in Applied Mechanics and Engineering*, vol. 217–220, pp. 46–57, Apr. 2012.

[23] S. Soghrati and H. Ahmadian, "3D hierarchical interface-enriched finite element method: Implementation and applications," *Journal of Computational Physics*, vol. 299, pp. 45–55, Oct. 2015.

[24] M. Safdari, A. R. Najafi, N. R. Sottos, and P. H. Geubelle, "A NURBS-based interface-enriched generalized finite element method for problems with complex discontinuous gradient fields," *International Journal for Numerical Methods in Engineering*, vol. 101, pp. 950–964, Mar. 2015.

[25] M. Safdari, A. R. Najafi, N. R. Sottos, and P. H. Geubelle, "A NURBS-based generalized finite element scheme for 3D simulation of heterogeneous materials," *Journal of Computational Physics*, vol. 318, pp. 373–390, Aug. 2016.

[26] J. Nitsche, "Über ein Variationsprinzip zur Lösung von Dirichlet-Problemen bei Verwendung von Teilräumen, die keinen Randbedingungen unterworfen sind," *Abhandlungen aus dem Mathematischen Seminar der Universität Hamburg*, vol. 36, pp. 9–15, July 1971.

[27] 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*, vol. 191, pp. 5537–5552, Nov. 2002.

[28] J. Dolbow and I. Harari, "An efficient finite element method for embedded interface problems," *International Journal for Numerical Methods in Engineering*, vol. 78, pp. 229–252, Apr. 2009.

[29] C. Annavarapu, M. Hautefeuille, and J. E. Dolbow, "A robust Nitsche's formulation for interface problems," *Computer Methods in Applied Mechanics and Engineering*, vol. 225–228, pp. 44–54, June 2012.

[30] W. Jiang, C. Annavarapu, J. E. Dolbow, and I. Harari, "A robust Nitsche's formulation for interface problems with spline-based finite elements," *International Journal for Numerical Methods in Engineering*, vol. 104, pp. 676–696, Nov. 2015.

[31] D. Schillinger, I. Harari, M.-C. Hsu, D. Kamensky, S. K. F. Stoter, Y. Yu, and Y. Zhao, "The non-symmetric Nitsche method for the parameter-free imposition of weak boundary and coupling conditions in immersed finite elements," *Computer Methods in Applied Mechanics and Engineering*, vol. 309, pp. 625–652, Sept. 2016.

[32] 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*, vol. 269, pp. 46–71, Feb. 2014.

[33] A. Stavrev, L. H. Nguyen, R. Shen, V. Varduhn, M. Behr, S. Elgeti, and D. Schillinger, "Geometrically accurate, efficient, and flexible quadrature techniques for the tetrahedral finite cell method," *Computer Methods in Applied Mechanics and Engineering*, vol. 310, 2016.

[34] S. Kollmannsberger, A. Özcan, J. Baiges, M. Ruess, E. Rank, and A. Reali, "Parameter-free, weak imposition of Dirichlet boundary conditions and coupling of trimmed and non-conforming patches," *International Journal for Numerical Methods in Engineering*, vol. 101, pp. 670–699, Mar. 2015.

[35] V. Varduhn, M.-C. Hsu, M. Ruess, and D. Schillinger, "The tetrahedral finite cell method: Higher-order immersogeometric analysis on adaptive non-boundary-fitted meshes," *International Journal for Numerical Methods in Engineering*, vol. 107, pp. 1054–1079, Jan. 2016.

[36] S. Duczek, F. Duvigneau, and U. Gabbert, "The finite cell method for tetrahedral meshes," *Finite Elements in Analysis and Design*, vol. 121, pp. 18–32, Nov. 2016.

[37] S. Duczek and U. Gabbert, "The finite cell method for polygonal meshes: Poly-FCM," *Computational Mechanics*, pp. 1–32, June 2016.

[38] W. Gui and I. Babuška, "The h, p and h-p versions of the finite element method in 1 dimension Part I: The error analysis of the p-version," *Numerische Mathematik*, vol. 49, pp. 577–612, Nov. 1986.

Journal publication: https://doi.org/10.1002/cnm.2951

[39] W. Gui and I. Babuška, "The h, p and h-p versions of the finite element method in 1 dimension Part II: The error analysis of the h-and h-p versions," *Numerische Mathematik*, vol. 49, pp. 613–657, Nov. 1986.

[40] W. Gui and I. Babuška, "The h, p and h-p versions of the finite element method in 1 dimension Part III: The Adaptive h-p Version," *Numerische Mathematik*, vol. 49, pp. 659–683, Nov. 1986.

[41] P. Šolín and J. Červený, "Automatic hp-adaptivity with arbitrary-level hanging nodes," Tech. Rep. Research Report No. 2006-07, The University of Texas at El Paso, Department of Mathematical Sciences, 2006.

[42] L. Demkowicz, W. Rachowicz, and P. Devloo, "A fully automatic hp-adaptivity," *Journal of Scientific Computing*, vol. 17, pp. 117–142, Dec. 2002.

[43] 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*, vol. 55, pp. 499–517, Feb. 2015.

[44] N. Zander, *Multi-Level hp-FEM: Dynamically Changing High-Order Mesh Refinement with Arbitrary Hanging Nodes*. Ph.D. thesis, Technische Universität München, Munich, 2017.

[45] N. Zander, T. Bog, M. Elhaddad, F. Frischmann, S. Kollmannsberger, and E. Rank, "The multi-level hp-method for three-dimensional problems: Dynamically changing high-order mesh refinement with arbitrary hanging nodes," *Computer Methods in Applied Mechanics and Engineering*, vol. 310, pp. 252–277, Oct. 2016.

[46] N. Zander, M. Ruess, T. Bog, S. Kollmannsberger, and E. Rank, "Multi-level hp-adaptivity for cohesive fracture modeling," *International Journal for Numerical Methods in Engineering*, vol. 109, 2017.

[47] 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*, vol. 91, pp. 457–471, Aug. 2012.

[48] L. Kudela, N. Zander, T. Bog, S. Kollmannsberger, and E. Rank, "Efficient and accurate numerical quadrature for immersed boundary methods," *Advanced Modeling and Simulation in Engineering Sciences*, vol. 2, pp. 1–22, June 2015.

[49] L. Kudela, N. Zander, S. Kollmannsberger, and E. Rank, "Smart octrees: Accurately integrating discontinuous functions in 3D," *Computer Methods in Applied Mechanics and Engineering*, vol. 306, pp. 406–426, July 2016.

[50] M. Joulaian, S. Hubrich, and A. Düster, "Numerical integration of discontinuities on arbitrary domains based on moment fitting," *Computational Mechanics*, pp. 1–21, Mar. 2016.

[51] A. Abedian, J. Parvizian, A. Düster, H. Khademyzadeh, and E. Rank, "Performance of Different Integration Schemes in Facing Discontinuities in the Finite Cell Method," *International Journal of Computational Methods*, vol. 10, p. 1350002, June 2013.

[52] V. Thiagarajan and V. Shapiro, "Adaptively weighted numerical integration over arbitrary domains," *Computers & Mathematics with Applications*, vol. 67, pp. 1682–1702, May 2014.

[53] C. A. Felippa, "Introduction to Finite Element Methods," lecture notes, Department of Aerospace Engineering Sciences, University of Colorado at Boulder, Boulder, Colorado, USA, 2013.

[54] 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*, vol. 95, pp. 811–846, Sept. 2013.

[55] C. D. Mote, "Global-local finite element," *International Journal for Numerical Methods in Engineering*, vol. 3, pp. 565–574, Oct. 1971.

[56] E. Rank, "Adaptive remeshing and h-p domain decomposition," *Computer Methods in Applied Mechanics and Engineering*, vol. 101, pp. 299–313, Dec. 1992.

[57] 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*, vol. 89, no. 9, pp. 1171–1202, 2012.

[58] D. D'Angella, N. Zander, S. Kollmannsberger, F. Frischmann, E. Rank, A. Schröder, and A. Reali, "Multi-level hp-adaptivity and explicit error estimation," *Advanced Modeling and Simulation in Engineering Sciences*, vol. 3, p. 33, Dec. 2016.

[59] G. Királyfalvi and B. A. Szabó, "Quasi-regional mapping for the p-version of the finite element method," *Finite elements in analysis and design*, vol. 27, no. 1, pp. 85–97, 1997.

[60] C. Maple, "Geometric design and space planning using the marching squares and marching cube algorithms," in *2003 International Conference on Geometric Modeling and Graphics, 2003. Proceedings*, pp. 90–95, IEEE, 2003.

[61] T. Bog, A. Mongeau, N. Zander, S. Kollmannsberger, and E. Rank, "Weak imposition of contact constraints on automatically recovered high-order, embedded interfaces using the finite cell method," vol. in review, 2017.

[62] B. A. Szabó, A. Düster, and E. Rank, "The p-version of the finite element method," in *Encyclopedia of Computational Mechanics* (E. Stein, ed.), Chichester, West Sussex: John Wiley & Sons, 2004.

[63] Z. Yosibash, "Numerical analysis on singular solutions of the Poisson equation in two-dimensions," *Computational Mechanics*, vol. 20, pp. 320–330, Sept. 1997.

[64] R. B. Kellogg, "On the poisson equation with intersecting interfaces," *Applicable Analysis*, vol. 4, pp. 101–129, Jan. 1974.

[65] I. Babuška and B. Guo, "Approximation properties of the $h-p$ version of the finite element method," *Computer Methods in Applied Mechanics and Engineering*, vol. 133, pp. 319–346, July 1996.

[66] M. Dauge, A. Düster, and E. Rank, "Theoretical and Numerical Investigation of the Finite Cell Method," *Journal of Scientific Computing*, vol. 65, pp. 1039–1064, Mar. 2015.

[67] D. H. Pahr, E. Dall'Ara, P. Varga, and P. K. Zysset, "HR-pQCT-based homogenised finite element models provide quantitative predictions of experimental vertebral body stiffness and strength with the same accuracy as μFE models," *Computer Methods in Biomechanics and Biomedical Engineering*, vol. 15, pp. 711–720, July 2012.

Journal publication: https://doi.org/10.1002/cnm.2951

[68] P. A. Yushkevich, J. Piven, H. C. Hazlett, R. G. Smith, S. Ho, J. C. Gee, and G. Gerig, "User-guided 3D active contour segmentation of anatomical structures: Significantly improved efficiency and reliability," *NeuroImage*, vol. 31, pp. 1116–1128, July 2006.

[69] Robert McNeel, "Rhinoceros 3D." http://www.rhino3d.com/.

[70] S. Bindick, M. Stiebler, and M. Krafczyk, "Fast kd-tree-based hierarchical radiosity for radiative heat transport problems," *International Journal for Numerical Methods in Engineering*, vol. 86, pp. 1082–1100, June 2011.

[71] Intel, "Intel Math Kernel Library." http://software.intel.com/en-us/intel-mkl.

[72] U. Ayachit, *The ParaView Guide: A Parallel Visualization Application*. USA: Kitware, Inc., 2015.

[73] A. J. Wirth, J. Goldhahn, C. Flaig, P. Arbenz, R. Müller, and G. H. van Lenthe, "Implant stability is affected by local bone microstructural quality," *Bone*, vol. 49, pp. 473–478, Sept. 2011.

[74] A. J. Wirth, R. Müller, and G. Harry van Lenthe, "The discrete nature of trabecular bone microarchitecture affects implant stability," *Journal of Biomechanics*, vol. 45, pp. 1060–1067, Apr. 2012.

[75] D. Ruffoni, A. J. Wirth, J. A. Steiner, I. H. Parkinson, R. Müller, and G. H. van Lenthe, "The different contributions of cortical and trabecular bone to implant anchorage in a human vertebra," *Bone*, vol. 50, pp. 733–738, Mar. 2012.

Journal publication: https://doi.org/10.1002/cnm.2951

4.4 Additive Manufacturing

Additive manufacturing is a process in which an artifact is produced in a layer-wise fashion. This is in contrast to subtractive manufacturing techniques such as milling, for example, in which material is subtracted from a larger, usually homogenous block of material to generate an engineering object. By now, there are so many additive manufacturing processes and such a large variety of individual techniques that an attempt to classify them is a science in itself (see e.g. [80] for a functional classification or Wohler's report [81] for a summary of processes with relevance to the industry of mechanical engineering.). It is convenient to divide the computational modeling of additive manufacturing into the modeling of the process itself and the computational modeling of the products generated by these processes.

The individual types of processes of additive manufacturing are very diverse. Within this treatise, the focus was placed on metal additive manufacturing because it is the most common method to produce load bearing artifacts. Specifically, the Finite Cell Method was extended to simulate the process of selective laser melting (SLM).

Selective laser melting is a powder bed technique. Powder is selectively melted by a rapidly moving laser beam. The liquid metal then locally re-solidifies before a new layer of powder is added. The process is repeated until a finished artifact is obtained. The process is multi-physical by nature as many physical phenomena occur, see e.g. fig. 30 for a graphic depiction of the process and a selection of the involved physical phenomena.

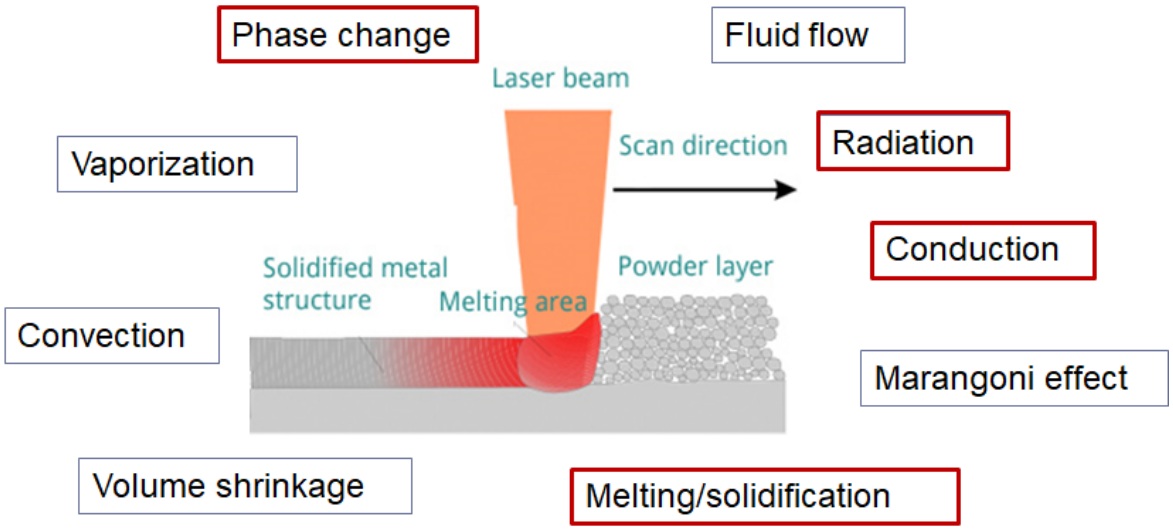


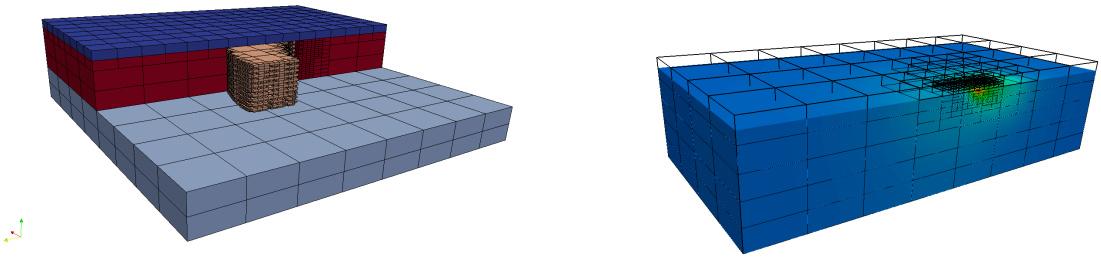
Figure 30: physical effects involved in the process of selective laser melting

The process also spans many scales. Assuming a typical diameter of the laser spot of $100\mu m$, the simulation of the local solidification needs a resolution of $10\mu m$ or better, resulting in 3-4 orders of magnitude to reach the size of a (small to moderately large) artifact. A 3D model to uniformly discretize the structure over the full range of these scales would have to be based on $O(10^9\text{-}10^{12})$ of the smallest units (voxels) with an edge-length of $10\mu m$ – which is far beyond the reach of any conventional transient multi-physics simulation. Therefore, it is common to split the scales into three different sections with partial overlaps.

The micro-scale is responsible for resolving the physics below $1mm$. It delivers insight into the melting/phase-change/solidification process whereby individual grains of powder are discretely resolved. The macro-scale resolves physics from $100\mu m$ up to the part level at a range of 1-10cm. Its main purpose is to model the thermo-mechanical behavior at part level. In between lies the meso-scale, which is responsible for the range from 10-50μm up to 1-10mm.

Since no computational method is able to capture the full multi-scale multi-physics nature of the SLM process in one single computation, a computational model must focus on one scale. Therein it is only possible to include a few physical effects that are specifically relevant for the scale under investigation.

A meso-scale model was developed for the simulation of the process of SLM. The physical effects that were considered to be of relevance at this scale are boxed in red in fig. 30. The developed meso-scale model is able to bridge a larger scale than classical uniform discretizations can because it employs the hierarchic enrichments described in section 3.4.2 to resolve the locally induced strongly changing gradients of the solution fields of the process. To track the state of the material in space and time, it additionally introduces a hierarchical octree-like treatment of the state variables (i.e. the material coefficients). This material grid may refine or coarsen independently of the discretization used for the temperature field, and it is submerged into the finite cells, see fig. 31 for a visualization of the material grid and the discretization.



(a) Discretization of the material coefficients: air in dark-blue, powder in red, solidified domain in brown, and base plate in light blue

(b) Discretization of the temperature field of the computational domain without air.

Figure 31: The separate discretization of material coefficients (upper left) and the temperature field (upper right) in one simulation by means of two grids

This separation of the treatment of state variables from the treatment of field variables is motivated by the fact that both exhibit a completely different behavior. The field variables are diffusive by nature because the high temperature gradients diffuse away once the laser has passed, until a practically uniform temperature field emerges. By contrast, the state variables do not diffuse. For example, once metal powder is melted, its state will never convert back to powder again. Once a residual stress state is introduced, it remains local unless further heat treatment is involved. This can also occur if the laser passes by again. It is interesting to note that the separation of state and field variables fully corresponds to, or rather is a consequent application of the Finite Cell Method. This is due to the fact that, in the Finite Cell Method, the material (for example the elastic modulus) is also not directly connected to the discretizations of the field variables themselves. Thus, the proposed methodology is merely an extension to transient simulations. However, it opens up new modeling possibilities

since neither laser scan vectors nor layers of material need to conform to the discretization.

More importantly, the introduced dynamic management of the discretization in an embedded domain sense allows for a more or less constant number of degrees of freedom throughout the simulation. This is a crucial point. Otherwise, the required fine discretization around the laser impact area combined with the growing computational domain due to the subsequent addition of layers would cause the involved degrees of freedom to grow out of bounds. In the subsequent chapters the three main aspects of the simulation of selective laser melting are discussed by including the corresponding journal publication. Section 4.4.1 treats the thermal evolution of the process. Its coupling to an elasto-plastic model in order to determine the residual stresses is presented in section 4.4.2. Most importantly, aspects of verification and validation are discussed in section 4.4.3 at the specific case of forecasting melt pool geometries and cooling rates.

4.4.1 Thermal behavior

The computational model of heat transfer is described in detail in [64] and presented next. From the physical side, it involves the treatment of the phenomena boxed in red in fig. 30.

Journal Publication

title:	A hierarchical computational model for moving thermal loads and phase changes with applications to selective laser melting
authors:	S. Kollmannsberger, A. Özcan, M. Carraturo, N. Zander, E. Rank
published at:	*Computers & Mathematics with Applications*
publisher:	Elsevier
year:	2018
volume:	75
pages:	1483–1497
doi:	https://doi.org/10.1016/j.camwa.2017.11.014

A hierarchical computational model for moving thermal loads and phase changes with applications to Selective Laser Melting

S. Kollmannsberger[a,*], A. Özcan[a], Massimo Carraturo[a,b], N. Zander[a], E. Rank[a]

[a] *Chair for Computation in Engineering, Technische Universität München, Arcisstr. 21, 80333 München, Germany*
[b] *Department of Civil Engineering and Architecture, University of Pavia, via Ferrata 3, 27100 Pavia, Italy*

Abstract

Computational heat transfer analysis often involves moving fluxes which induce traveling fronts of phase change coupled to one or more field variables. Examples are the transient simulation of melting, welding or of additive manufacturing processes, where material changes its state and the controlling fields are temperature and structural deformation. One of the challenges for a numerical computation of these processes is their multi-scale nature with a highly localized zone of phase transition which may travel over a large domain of a body. Here, a transient local adaptation of the approximation, with not only a refinement at the phase front, but also a de-refinement in regions, where the front has past is of advantage because the de-refinement can assure a bounded number of degrees of freedom which is independent from the traveling length of the front.

We present a computational model of this process which involves three novelties: a) a very low number of degrees of freedom which yet yields a comparatively high accuracy. The number of degrees of freedom is, additionally, kept practically constant throughout the duration of the simulation. This is achieved by means of the multi-level *hp*-finite element method. Its exponential convergence is verified for the first time against a semi-analytic, three-dimensional transient linear thermal benchmark with a traveling source term which models a laser beam. b) A hierarchical treatment of the state variables. To this end, the state of the material is managed on a separate, octree-like grid. This material grid may refine or coarsen independently of the discretization used for the temperature field. This methodology is verified against an analytic benchmark of a melting bar computed in three dimensions in which phase changes of the material occur on a rapidly advancing front. c) The combination of these technologies to demonstrate its potential for the computational modeling of selective laser melting processes. To this end, the computational methodology is extended by the finite cell method which allows for accurate simulations in an embedded domain setting. This opens the new modeling possibility that neither a scan vectors no a layer of material needs to conform to the discretization of the finite element mesh but can form only a fraction within the discretization of the field- and state variables.

Keywords: additive manufacturing, *hp*-FEM, transient thermal problems, dynamically changing meshes

Contents

*Corresponding author

Journal publication: https://doi.org/10.1016/j.camwa.2017.11.014

Journal publication: https://doi.org/10.1016/j.camwa.2017.11.014

1. Introduction

The computational analysis of powder bed fusion processes such as e.g. selective laser melting (SLM) is challenging due to many reasons. The most prominent include:

1. highly localized and moving strong temperature gradients
2. non-linearities due to temperature dependent coefficients and phase changes of the material
3. growing and possibly geometrically complex computational domains
4. large range of scales in both space and time
5. coupled multi-physics

This article presents a methodology for the computational modeling of the temperature evolution in a powder bed fusion process taking into account the first three issues. While we incorporate the non-linearities due to temperature dependent coefficients and phase changes of the material using the rather standard latent heat model first presented in [1], special focus lies on the discretization of the highly localized and moving strong temperature gradients and on the representation of growing computational domains.

The evolution of temperature fields in space is a diffusion dominated process which can be well resolved by the finite element method. Many commercial packages are available, of which `ABAQUS`® and `ANSYS`® are two popular choices for applied research. These and other commercial packages provide a wealth of physical models, but their discretizational technology is mostly limited to linear, at most quadratic finite elements. Therefore, the resolution of local gradients is limited to h- refinements, i.e. refining the mesh size towards singularities.

Strong gradients, however, can most efficiently be resolved by hp-finite elements which vary the size of the element h locally as well as the polynomial degree of the trial/test space p [2, 3]. While hp-fem leads to efficient discretizations where error estimators are used to drive an adaptive scheme, it also provides excellent accuracy in cases where the solution characteristic is known a priori. This is the case for the simulation of powder bed fusion processes because the area of refinement is well defined by the location of the laser spot where sudden, high temperatures cause phase changes in the material.

Moreover, most simulations of powder bed fusion processes use static discretization schemes, i.e. the mesh is refined towards the entire laser path and kept fixed at all time-steps. As a consequence, the necessary number of degrees of freedom is directly proportional to the length of the laser path. However, high gradients are local to the laser spot itself and not distributed along all of its path. Therefore, the number of degrees of freedom should be independent of the length of the laser path and at best constant over time. To this end, transient refinement and de-refinements of the discretization throughout the run time of the simulation is necessary to keep the refinement local to the current position of the laser. Only recently, discretizations have appeared that utilize these kind of transient meshes for computational SLM analysis, see e.g. [4, 5, 6] and references therein. To the authors' knowledge, all of these contributions exploit h-refinements for low order polynomials, only. Transient h-refinements for higher order polynomials have not been used in that context although even transient hp codes do exist along with instructive literature, see e.g. [7, 8, 9, 10] and the introduction of [11] for a recent overview.

Another important aspect is the treatment of the state variables. While the evolution of temperature is a diffusive process, the evolution of the material state is not. Solidified material does not diffuse into regions containing powder. Additionally, material interfaces may not coincide with the boundaries of the finite elements. For example, material may need to be added in form of powder in a way which does not necessarily conform to the finite element discretization. In the paper at hand, we propose to provide this flexibility by discretizing the material coefficients independently of the underlying discretization of the field variables.

The article is structured as follows: We start by introducing the governing equations in section 2.1 and present its discretized weak from in section 2.2. We then give a quick introduction into the recently introduced multi-level hp-finite element method [11], which provides hp-discretizations on transient meshes. To evaluate its accuracy, we first present results for a transient but linear, three-dimensional benchmark resembling a SLM process in section 2.4.1 before proceeding to evaluate the scheme against a transient non-linear benchmark involving phase changes and latent heat in section 2.4.2.

Journal publication: https://doi.org/10.1016/j.camwa.2017.11.014

We then proceed to combine the multi-level *hp*-method with the finite cell method in section 3.1 which was initially designed to avoid boundary conforming mesh generation for complex domains. We use this concept to treat state and field variables on different discretizations. Section 3.2 then presents an example computing the evolving interface of a structure. Herein, two independent and transiently changing discretizations are used for state and field variables. Their separate treatment combined with the multi-level *hp* method allows for a relatively low number of degrees of freedom which stay almost constant throughout the simulation process.

2. Thermal analysis with phase changes

This section sets out to describe a new discretizational scheme for thermal analysis with phase changes. To clear the view, we neglect effects of radiation and mass transfer even though they have physical relevance in practical examples.

2.1. Governing equations

In this spirit, let us consider a domain, $\Omega \subset \mathbb{R}^n$ with boundary $\partial\Omega$, where n is the number of space dimensions. The governing nonlinear transient heat conduction equation with phase-change, written in terms of volumetric enthalpy $H = H(T)$ and temperature $T = T(\mathbf{x})$ fields, has been investigated by many researchers. In the sequel, we closely follow the presentation given in [12] which reads:

$$\frac{\partial H}{\partial t} - \nabla \cdot (k \nabla T) = Q \quad \text{in} \quad \Omega, \tag{1}$$

where t is the time, k is the thermal conductivity and Q is the heat source. Equation (1) is subjected to the initial condition

$$T(\mathbf{x}, t = 0) = T_0(\mathbf{x}) \quad \text{in} \quad \Omega. \tag{2}$$

Dirichlet, Neumann, convection and radiation boundary conditions can be defined on non-overlapping boundaries:

$$T = T_{\mathrm{w}}(\mathbf{x}) \quad \text{on} \quad \partial\Omega_T, \tag{3}$$

$$(k \nabla T) \cdot \mathbf{n} = q(\mathbf{x}) \quad \text{on} \quad \partial\Omega_q, \tag{4}$$

$$(k \nabla T) \cdot \mathbf{n} = h_{conv}(T_\infty - T(\mathbf{x})) \quad \text{on} \quad \partial\Omega_c, \tag{5}$$

$$(k \nabla T) \cdot \mathbf{n} = \sigma\epsilon \left(T^4 - T(\infty)^4\right) \quad \text{on} \quad \partial\Omega_r, \tag{6}$$

where T_{w}, q, h_{conv} and T_∞ are the prescribed temperature, prescribed heat flux, thermal convection coefficient and ambient temperature, respectively. Further, σ is the Stefan-Boltzmann constant and ϵ represents the emissivity.

The volumetric enthalpy function is defined as

$$H(T) = \int_{T_{\mathrm{ref}}}^{T} \rho c(T)\, \mathrm{d}T + \rho L \tilde{f}_{pc}(T), \tag{7}$$

where ρ,c,L,T_{ref} and f_{pc} denote density, specific heat capacity, latent heat, a reference temperature and a phase-change function, respectively. The function f_{pc} depends on the nature of the process. In an isothermal phase change, the temperature T_m stays constant during the phase change and is defined by a heavyside step function:

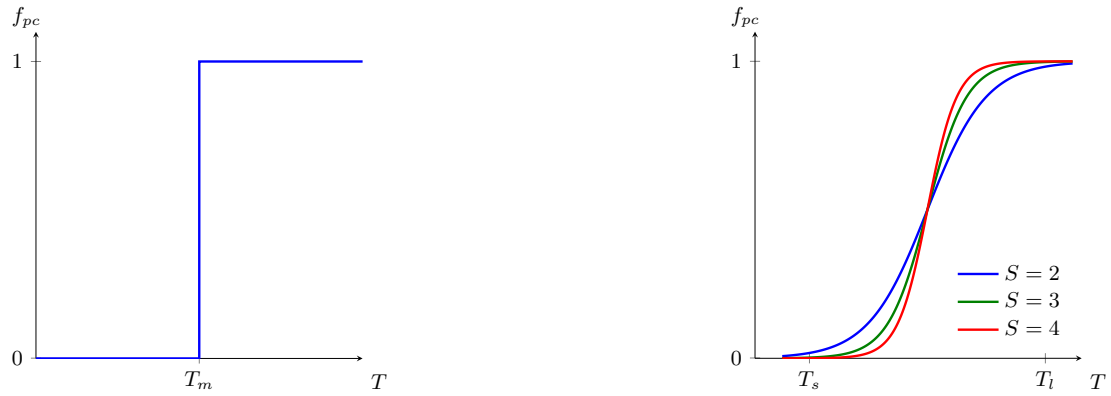
$$\tilde{f}_{pc}(T) = \begin{cases} 0 & T \leq T_m \\ 1 & T > T_m \end{cases}, \tag{8}$$

Journal publication: https://doi.org/10.1016/j.camwa.2017.11.014

see fig. 1a for an illustration. For numerical reasons, the function $\tilde{f}_{pc}$ is regularized by a smooth function as

$$f_{pc}(T) = \frac{1}{2}\Big[\tanh\Big(S\,\frac{2}{T_l - T_s}\Big(T - \frac{T_s + T_l}{2}\Big)\Big) + 1\Big], \tag{9}$$

which is depicted in fig. 1b for different values of S which controls the smoothing.



(a) Isothermal case

(b) Non-isothermal case

Figure 1: Phase change function f_{pc}

A substitution of eq. (7) in eq. (1) leads to the temperature based phase-change model:

$$\rho c\frac{\partial T}{\partial t} + \rho L\frac{\partial f_{pc}}{\partial t} - \nabla\cdot(k\nabla T) = Q \quad \text{in} \quad \Omega. \tag{10}$$

Equation (10) reduces to the classical transient heat conduction equation, when the latent heat term is neglected.

2.2. Discretized weak form

The governing partial differential eq. (10) subjected to the initial condition given by eq. (2) and the Dirichlet, Neumann and convection boundary conditions given in eqs. (3) to (5), respectively is now discretized in time and space. For the spatial discretization, the Bubnov-Galerkin finite element method is ideally suited due to the mainly diffusive nature of the described process. To this end, the set $\mathcal{V}$ of admissible solutions T and a set $\mathcal{V}_0$ of admissible test functions ψ is defined as

$$\mathcal{V} = \{v \in H^1(\Omega), \quad v = T_{\mathrm{w}}(\mathbf{x},t) \ \text{ on } \ \partial\Omega_T\} \quad \text{and} \quad \mathcal{V}_0 = \{v \in H^1(\Omega), \quad v = 0 \ \text{ on } \ \partial\Omega_T\}, \tag{11}$$

where H^1 is the Hilbert space. The weak form of eqs. (2) to (5) then reads:

$$\begin{aligned}&\text{Find} \quad T \in \mathcal{V}, \quad \text{such that}\\ &\int_\Omega \psi\rho c\frac{\partial T}{\partial t}\,\mathrm{d}\Omega + \int_\Omega \psi\rho L\frac{\partial f_{pc}}{\partial t}\,\mathrm{d}\Omega + \int_\Omega \nabla\psi\cdot(k\nabla T)\,\mathrm{d}\Omega = \int_\Omega \psi Q\,\mathrm{d}\Omega + \int_{\partial\Omega_n} \psi q\,\mathrm{d}\Gamma + \int_{\partial\Omega_c} \psi\, h(T_\infty - T)\,\mathrm{d}\Gamma.\end{aligned} \tag{12}$$

In the framework of the Bubnov-Galerkin finite element method, solution field and test functions are approximated by the same shape functions N_i as follows:

$$T(\mathbf{x},t) \approx T_h(\mathbf{x},t) = \sum_{i=1}^{n} N_i(\mathbf{x})T_i(t), \qquad \psi(\mathbf{x},t) \approx \psi_h(\mathbf{x},t) = \sum_{i=1}^{n} N_i(\mathbf{x})\psi_i(t), \tag{13}$$

Journal publication: https://doi.org/10.1016/j.camwa.2017.11.014

where T_i and ψ_i are the unknown coefficients. Substituting the approximations eq. (13) into the weak form eq. (12) yields the following semi-discrete equilibrium equation:

$$\begin{aligned}
&\mathbf{C}\dot{\mathbf{T}} + \dot{\mathbf{L}} + \mathbf{K}\mathbf{T} = \mathbf{F} \\
&C_{ij} = \int_\Omega \rho c N_i N_j \,\mathrm{d}\Omega \\
&\dot{L}_i = \int_\Omega \rho L N_i \frac{\partial f_{pc}}{\partial t} \,\mathrm{d}\Omega \\
&K_{ij} = \int_\Omega \nabla N_i \cdot (k \nabla N_j) \,\mathrm{d}\Omega + \int_{\partial\Omega_c} h N_i N_j \,\mathrm{d}\Gamma \\
&F_i = \int_\Omega N_i Q \,\mathrm{d}\Omega + \int_{\partial\Omega_q} N_i q \,\mathrm{d}\Gamma + \int_{\partial\Omega_c} h N_i T_\infty \,\mathrm{d}\Gamma
\end{aligned} \tag{14}$$

where $\mathbf{C}$ is the capacitance matrix, $\mathbf{K}$ is the conductivity matrix, $\mathbf{L}$ is the latent heat vector, $\mathbf{F}$ is the load vector and $\mathbf{T}$ is the temperature coefficient vector. The residual vector $\mathbf{R}$ for the transient nonlinear analysis is obtained by using the backward Euler time integration scheme for the terms $\dot{\mathbf{L}}$ and $\dot{\mathbf{T}}$ in eq. (14):

$$\mathbf{R}_{n+1} = \mathbf{F}_{n+1} - \mathbf{C}_{n+1}\frac{\mathbf{T}_{n+1} - \mathbf{T}_n}{\Delta t} - \frac{\mathbf{L}_{n+1} - \mathbf{L}_n}{\Delta t} - \mathbf{K}_{n+1}\mathbf{T}_{n+1} \stackrel{!}{=} \mathbf{0}. \tag{15}$$

The subscripts n and $n+1$ represent evaluations at time t and $t + \Delta t$, respectively. In order to solve this nonlinear equation, we use an iterative incremental scheme, where the current temperature vector is:

$$\mathbf{T}^{i+1}_{n+1} = \mathbf{T}^{i}_{n+1} + \Delta\mathbf{T}^{i}. \tag{16}$$

$$\mathbf{J}^{i}_{n+1}\Delta\mathbf{T}^{i} = \mathbf{R}^{i}_{n+1}. \tag{17}$$

Equation (17) shows the incremental system to be solved, where $\mathbf{J}$ is the tangent Jacobian matrix which is defined as

$$\mathbf{J}^{i}_{n+1} = -\frac{\partial \mathbf{R}}{\partial \mathbf{T}}\bigg|^{i}_{n+1} = \mathbf{K}^{i}_{n+1} + \frac{\mathbf{C}^{i}_{n+1}}{\Delta t} + \frac{\mathbf{L}'|^{i}_{n+1}}{\Delta t}. \tag{18}$$

The latent heat contribution $\mathbf{L}'$ of the Jacobian matrix $\mathbf{J}$ is:

$$L'_{ij} = \int_\Omega \rho L \frac{\partial f_{pc}}{\partial T} N_i N_j \,\mathrm{d}\Omega, \tag{19}$$

where we approximate the temperature derivative of the function f_{pc} as suggested in [12]:

$$\frac{\partial f_{pc}}{\partial T}\bigg|^{i}_{n+1} = \frac{f_{pc}(T^{i}_{n+1}) - f_{pc}(T_n)}{T^{i}_{n+1} - T_n}, \tag{20}$$

2.3. Multi-level hp-FEM: Discretization of the primal unknowns

It lies in the nature of the SLM process to induce phase change locally by application of a highly focused laser beam. This heat flux is discretized in $\mathbf{F}$ and the induces high temperatures locally which diffuse rapidly into the domain. The resulting high but non-singular gradients are best captured by *hp* finite element schemes.

Implementations of *hp*-finite elements are widely available in the scientific community, see e.g. [3, 13, 14, 15, 16]. Research in the field of isogeometric analysis has further amplified the available code-basis, see

Journal publication: https://doi.org/10.1016/j.camwa.2017.11.014

e.g. [17]. However, the situation is less comfortable in cases where dynamic *hp*-discretizations are necessary in three dimensions. This is due to the fact that handling degrees of freedom on changing mesh topologies proves to be difficult in 3D. The recently introduced multi-level *hp*-method aims at alleviating this burden. An *en-detail* description of the method is given in [18, 11, 19] along with a review of other, related methods.

Classic *hp*-approaches replace finite elements with refined elements and constrain hanging nodes, edges and faces to re-establish a C^0-compatible, global trial and test space. The multi-level *hp*-method takes a completely different approach. Its underlying idea is to retain coarse elements in the mesh. The refinement is then constructed hierarchically such that global C^0-continuity and linear independence is maintained by construction. This renders post-constraining unnecessary. The principle idea is depicted in fig. 2a) in one dimension. Compatibility is ensured by applying homogeneous boundary conditions on all boundaries of the overlay mesh. In one dimension this translates to deactivating all nodal degrees of freedom on the overlay meshes which correspond to the boundary of the overlay. Linear independence is guaranteed by deactivating the high-order modes on the lower levels. Thereby, high-order shape functions are *h*-refined as well and finite element spaces are constructed which are very close those generated by classical *hp*-finite element methods [20]. The simple rule set of activating and de-activating nodal and edge modes directly translates to two- and three-dimensions as depicted in fig. 2b) and c) if face and internal modes are accounted for likewise.

2.4. Examples

This section addresses the first two computational challenges stated in the introductory section 1. To this end we evaluate the accuracy of the multi-level *hp*-method by means of a comparison to two semi-analytical benchmarks, which resemble SLM-typical problems: a moving laser source in linear thermodynamics in section 2.4.1 and a variant of Stefan's problem involving phase changes in section 2.4.2.

2.4.1. Linear thermal analysis

It was already demonstrated in [11] that strong gradients in a stress field can be captured accurately on moving discretizations. The paper at hand investigates the (parabolic) heat equation commonly used for modelling SLM processes. We consider the following simplified form of eq. (21)

$$\rho c \frac{\partial T}{\partial t} - \nabla \cdot (k \nabla T) = q \quad \text{in} \quad \Omega. \tag{21}$$

where q is the Gaussian surface distributed heat source:.

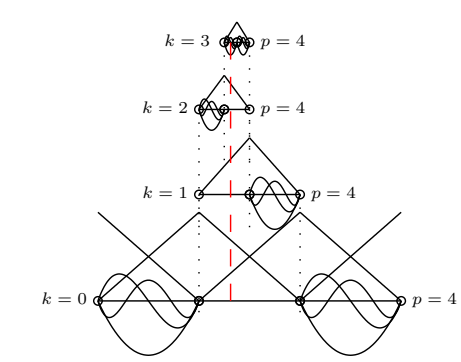
$$\begin{aligned} q(x,z,t) &= \lim_{b\to 0} \frac{6\sqrt{3}Q}{\pi\sqrt{\pi}ab} \int_0^{\inf} \exp\left(-3\frac{y^2}{b^2}\right) dy \times \frac{1}{c} \exp\left[-3\frac{x^2}{a^2} - 3\frac{(z-vt)^2}{c^2}\right] \\ &= \frac{3Q}{\pi a} \times \frac{1}{c} \exp\left[-3\frac{x^2}{a^2} - 3\frac{(z-vt)^2}{c^2}\right], \end{aligned} \tag{22}$$

which is also commonly referred to as an elliptical disk heat source, see e.g [21]. Its distribution parameters a and c are referred to as its radii and the maximum heat power is denoted by Q. The center of the heat source travels with a constant speed v along the path A—B on the upper boundary of the semi-infinite body given in fig. 3a. It is depicted in fig. 3b along with its local coordinate system.

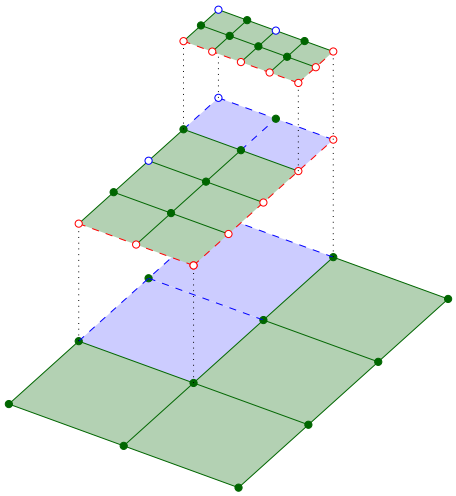
The analytical solution of this transient temperature field was first introduced by [21]. Herein, the space and time dependent temperature $T(x,z,t)$ is given as the initial temperature T_0 plus a time integral from the start of the process at $t = 0$ to the time of interest t:

$$T(x,z,t) = T_0 + \frac{3\sqrt{3}}{\pi\sqrt{\pi}} Q/(\rho h_c) \times \int_0^t \frac{\exp\left[-3\frac{x^2}{12\kappa(t-t')+a^2}\right]}{\sqrt{12\kappa(t-t')+a^2}\sqrt{12\kappa(t-t')}} \times 2A\,dt', \tag{23}$$

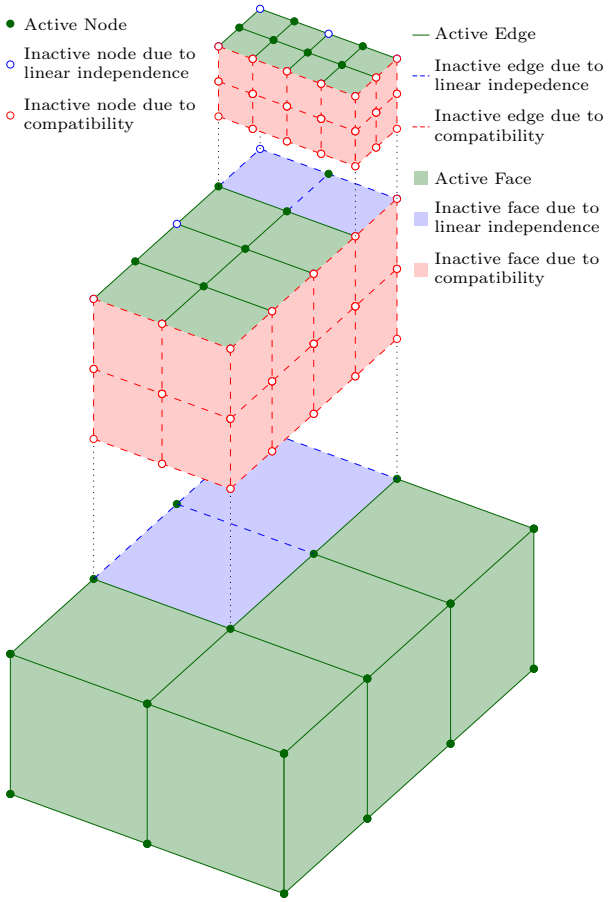
Journal publication: https://doi.org/10.1016/j.camwa.2017.11.014



(a) One-dimensional case



(b) Two-dimensional case



(c) Three-dimensional case

Figure 2: Conceptual idea of the multi-level *hp*-method following [18, 11, 19]

whereby the abbreviation A

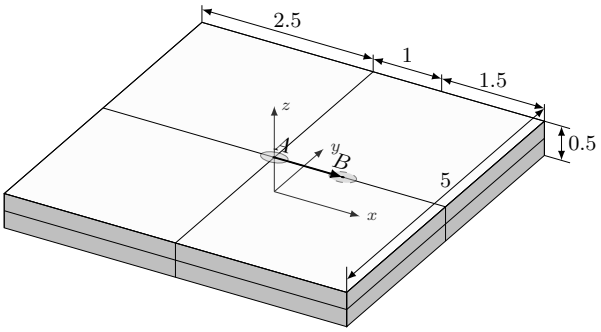
$$A = A(z,t,t') = \frac{\exp\left[-3\frac{(z-vt')^2}{12\kappa(t-t')+c^2}\right]}{\sqrt{12\kappa(t-t')+c^2}}, \tag{24}$$

and $\kappa = k/(\rho h_c)$ is the thermal diffusivity. The parameters of the setup are chosen in the range of a typical SLM scanning process and listed in table 1.

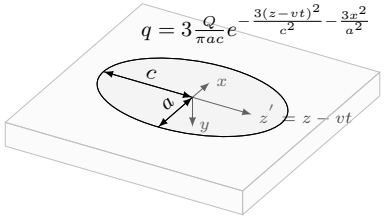
Radiation-convection boundary conditions are imposed on the bottom and side surfaces by setting the environmental temperature to $T_{env} = 0°\text{C}$ and the convection coefficient to $h_{conv} = 0.0$ [W/m^2°C]. This leads to an approximation of the temperature at the surfaces cutting the considered block out of the half-domain which would otherwise be given by eq. (23). However, these cut-off surfaces are far enough away for this approximation to have any notable effect on the temperature distribution along the path A—B. The time domain was discretized by 500 hundred time steps with $\Delta t = 4$ [μsec].

The base mesh of the multi-level *hp*-discretization is depicted in fig. 3a and consists of $2 \times 2 \times 2$ elements. This base mesh is refined by successively superposing finer overlay elements that halve the size of their

Journal publication: https://doi.org/10.1016/j.camwa.2017.11.014



(a) Geometric dimensions in mm and initial mesh of $2 \times 2 \times 2$ hexahedral elements



(b) moving heat source with characteristic parameters

Figure 3: Problem Setup

Specific heat (h_c)	600 J/(kg°C)
Density (ρ)	7820 kg/m^3
Heat Conductivity (k)	29 W/(m°C)

Heat Power (Q)	50.83 W
Laser speed (v)	0.5 m/s
Half radius a	0.1 mm
Half radius c	0.15 mm

Table 1: Material and load parameters for the benchmark defined in fig. 3

parent. Figure 4 gives an impression of the resulting mesh for five overlays. A possible measure of resolution is to relate the numbers of elements per twice the smallest distribution parameter of q defined in eq. (22), here $a = 0.1[mm]$. For the base mesh, this ratio is 0.2[mm]/2.5[mm]=0.08 which is then doubled by each level of refinement i.e. levels one to five lead to: 0.16, 0.32, 0.64, 1.28, 2.56 elements per $2a$. The refinement is not carried out uniformly but towards a bounding box which defines the zone of maximum h refinements. It is initially located at the center of the laser beam, has the initial dimensions 0.125[mm]/0.0625[mm]/0.03[mm], and is oriented along the global x/y/z-axis. During the scanning process, the rear face of the bounding box is kept fixed until it has an elongation of 0.7[mm]. The size of the bounding box then stays fixed and the bounding box follows the laser path with constant dimensions until its end.

The results for the temperature are depicted for different time-steps in fig. 5. While the colors in the picture correspond to the refinement depth, the discretization is warped in the global z-direction using the computed temperature field. The picture series demonstrates the dynamic change of the discretization over time, which allows the refinement zone to stay local to the moving laser spot.

Figure 6a depicts the spatial solution of the temperature along the laser path, i.e. the cutline between point A and B in fig. 3 for different polynomial degrees. Figure 6b records the temperature history of a material point located at the coordinates x=0.25, y=0.0, z=0.1. It can clearly be seen how increasing the polynomial degree p of the approximation helps in increasing the solution accuracy.

To obtain a better insight into the convergence behavior, p-extensions (i.e. sequence of computations with increasing polynomial degrees) were carried out on different h-refinements. To this end, the analytical solution given by eq. (23) was computed using the function `integral` in matlab®. This function performs an adaptive quadrature with a relative tolerance of $1e-6$ [22]. The computation was carried out at 1000 equidistant points along the laser path between point A and B[1]. and its deviation from the numerical approximation obtained by the multi-level hp-method served as an error measure in the sense of a discrete-L_2 norm:

$$\|e\|_{L_2} = \sqrt{\frac{\sum_{i=0}^{1000}(T_{san,i} - T_i)^2}{\sum_{j=0}^{1000} T_{san,j}^2}} \times 100. \tag{25}$$

[1] in fig. 3a

Journal publication: https://doi.org/10.1016/j.camwa.2017.11.014

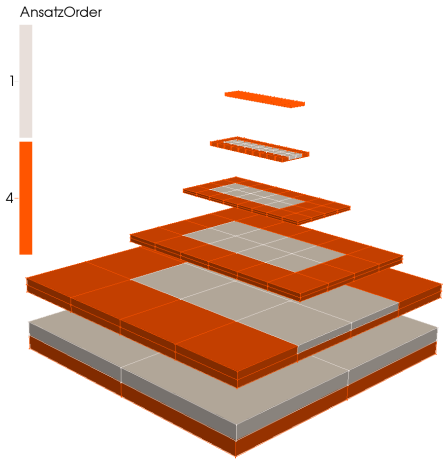


Figure 4: Local 3D multi-level *hp* mesh

Herein T_{san} represents the semi-analytical solution of eq. (23), and T_i is the temperature obtained by the multi-level *hp*-method at the i_{th} point.

The convergence plots are presented in two forms for the same data. Figure 7a gives the number of degrees of freedom versus the discrete, relative L_2 error computed by eq. (25) whereby both axis possess logarithmic scale. Figure 7b displays the same data, but the abscissa is scaled by the third root of the degrees of freedom. The blue line with the filled dots labeled 'uniform h, p=1' represents the accuracy obtained by discretizing the domain uniformly starting with a mesh of size $2 \times 2 \times 2$, then $4 \times 4 \times 4$, then $8 \times 8 \times 8$ up to $32 \times 32 \times 32$ elements. The relative error does not fall below 27% at 35,937 degrees of freedom for this strategy. This demonstrates how poorly a uniform refinement converges. All other curves are obtained by performing a multi-level *hp*-refinement towards the bounding box as described above. Consider the blue line with the blue circles labeled "1 ml, p=1...10". The mesh here consists of the base mesh plus one refinement. For each dot, the mesh stays fixed and the polynomial degree is increased from one to ten anlong the line. The error now drops to 6% for $p = 10$, and its decrease is exponential. This is indicated by the straight line in fig. 7b. It is noteworthy that each added level of h-refinement upon which a p-extension is carried out uniformly leads to better results with less degrees of freedom only until level four. Level five is worse again. Here, too many degrees of freedom are spent in parts of the domain where the error is already low. This may be avoided by use of an error estimator. In any case, a very good discretization is obtained using four refinements with a polynomial degree of four. Here, only 3857 degrees of freedom are needed to obtain an error of 1.3%. At this point, the results delivered by the multi-level *hp*-strategy are approximately twenty times more accurate at ten times less the degrees of freedom than a bold h-refined strategy for this setup.

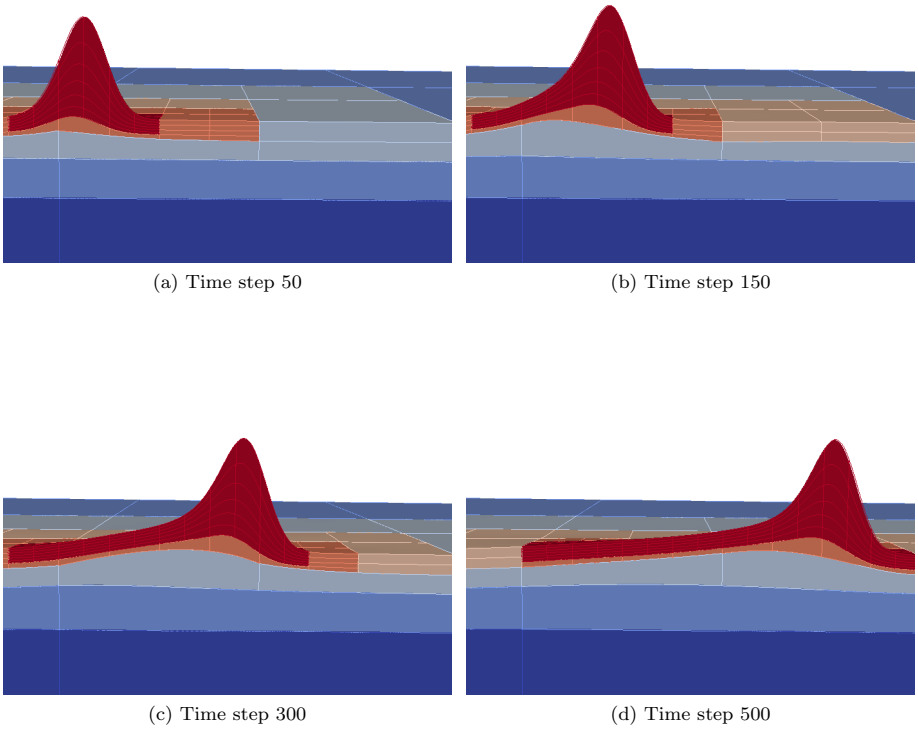
2.4.2. Melting bar

In this section, we evaluate the numerical approximation to the isothermal phase change model that is introduced in section 2. We are especially interested in the methods ability to accurately resolve the interface between liquid and solid parts of the domain. Unfortunately, exact solutions to eq. (10) are only available for very few idealized situations in a dimensionally reduced setting.

We consider the two-phase problem of a one-dimensional semi-infinite bar. The bar is initially solid with a constant temperature T_s. The boundary condition at $x = 0$ is then suddenly changed to a stationary temperature T_l which is larger than the melting temperature T_m. The analytical solution was originally described in [23] and is known in the literature as *Neumanns's method*, see e.g. [24, 25].

The sudden change of the temperature from T_s to T_l causes the bar to melt. The position of the interface

Journal publication: https://doi.org/10.1016/j.camwa.2017.11.014



(a) Time step 50 (b) Time step 150

(c) Time step 300 (d) Time step 500

Figure 5: Temperature solution at different time steps and adapted refinements depth

between melt and solid is given by:

$$X(t) = 2\lambda\sqrt{\alpha_l t}, \tag{26}$$

where t is the time and α_l is the diffusivity of the material in its liquid state. The constant λ in equation 26 is computed by solving the following non-linear equation

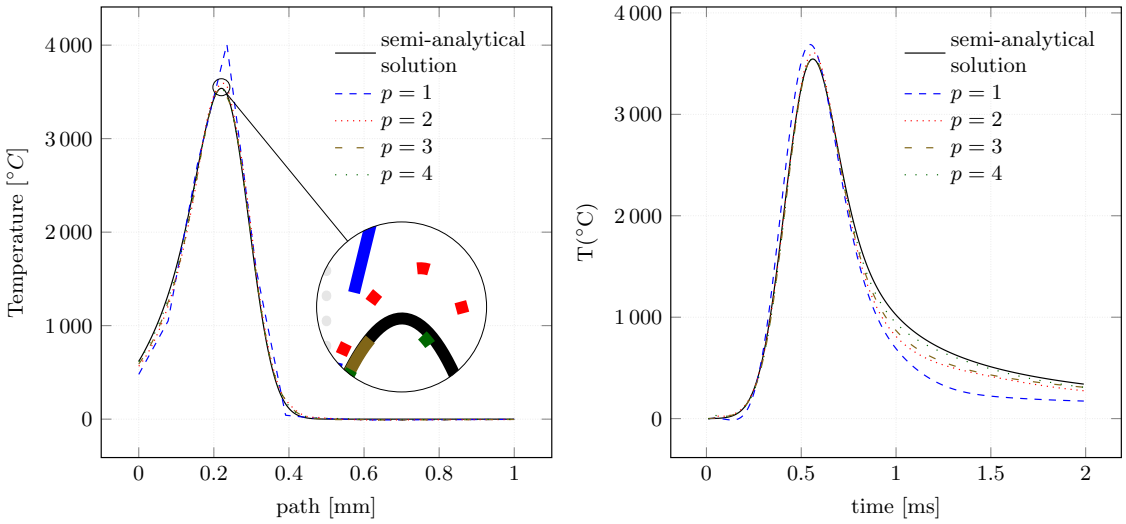
$$\frac{\mathrm{St}_l}{\exp(\lambda^2)\mathrm{erf}(\lambda)} - \frac{\mathrm{St}_s\sqrt{\alpha_s}}{\sqrt{\alpha_l}\exp(\alpha_l\lambda^2/\alpha_s)\mathrm{erfc}(\lambda\sqrt{\alpha_l/\alpha_s})} = \lambda\sqrt{\pi}, \tag{27}$$

where St_l and St_s are the Stefan number of liquid and solid phases respectively. They can be computed as follows:

$$\mathrm{St}_l = \frac{C_l(T_l - T_m)}{L}, \quad \mathrm{St}_s = \frac{C_s(T_m - T_s)}{L}. \tag{28}$$

In eq. (28), L is the latent heat of fusion, and C_l and C_s are the heat capacity of the liquid and solid phases,

Journal publication: https://doi.org/10.1016/j.camwa.2017.11.014



(a) Spatial solution along the laser path after 0.5 msec

(b) Temporal solution at the observing point (0.25, 0.0, 0.1)

Figure 6: Temperature solution for 4 multi-level refinements for polynomial orders $p = 1...4$.

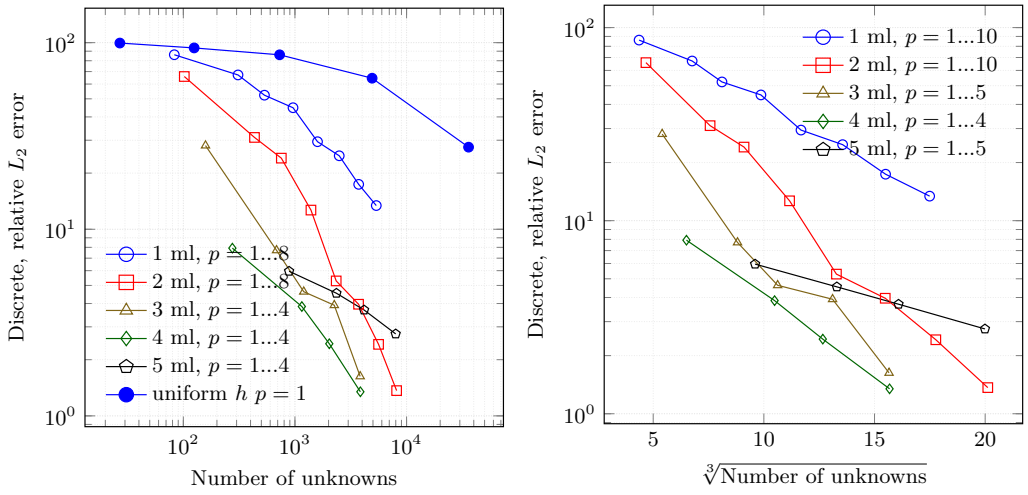
respectively. The analytical temperature distribution over the semi-infinite slab is given by

$$T(\mathbf{x},t) = \begin{cases} T_l - (T_l - T_m)\dfrac{\operatorname{erf}(x/2\sqrt{\alpha_l t})}{\operatorname{erf}(\lambda)} & \text{if} \quad x \leq X(t) \\ \\ T_s + (T_m - T_s)\dfrac{\operatorname{erfc}(x/2\sqrt{\alpha_s t})}{\operatorname{erfc}(\lambda\sqrt{\alpha_l/\alpha_s})} & \text{if} \quad x > X(t) \end{cases} . \tag{29}$$

Figure 8 illustrates the dimensions of the bar which is used for validation of the numerical scheme. The bar consists of pure Titanium and is assumed to have the thermo-physical properties provided in table 2. These lead to the constant $\lambda = 0.388150542167233$, which is computed from eq. (27). On the face at $x = 0.1$ the analytical solution is imposed as a constant Dirichlet boundary condition to emulate the semi-infinite domain. The simulation was carried out on a base mesh with 10 finite elements of order $p = 3$ with four multi-level *hp*-refinements. The time domain was discretized by a backward Euler scheme with a time step of $dt = 1$[s].

Figure 9a shows the corresponding numerical solution of the temperature along the x-direction of the bar at different time steps together with the analytical solution. The kink in the solution at the melting temperature T_m is clearly visible. It stems from the latent heat contribution represented by $\dot{L}_i$ in the semi-discrete weak form given in eq. (14). The lens zoom depicts how close the numerical solution resembles its analytic counterpart. Figure 9b depicts the evolution of the temperature at the point $x = 0.01$[m] which is very close to the Dirichlet interface and therefore difficult to catch. The kink at T_m is also clearly visible and well captured by the numerical scheme.

Journal publication: https://doi.org/10.1016/j.camwa.2017.11.014



(a) Loglog plot (b) Semilog plot

Figure 7: Convergence in discrete L_2-norm starting from a $2 \times 2 \times 2$ base mesh and hierarchically refining in h 2, 3 or 4 times (used abbreviations: ml = multi-level hp refinement)

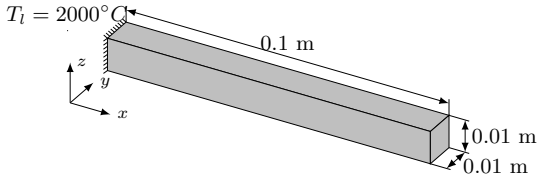


Figure 8: Problem setup of the melting bar

T_m	1670 °C
T_l	2000 °C
T_s	1500 °C
ρ	4.51×10^3 kg/m^3
$c_l = c_s$	520 J/(m^3 °C)
$k_l = k_s$	16 W/(m °C)
L	325×10^3 J/kg

Table 2: Thermo-physical properties for Titanium

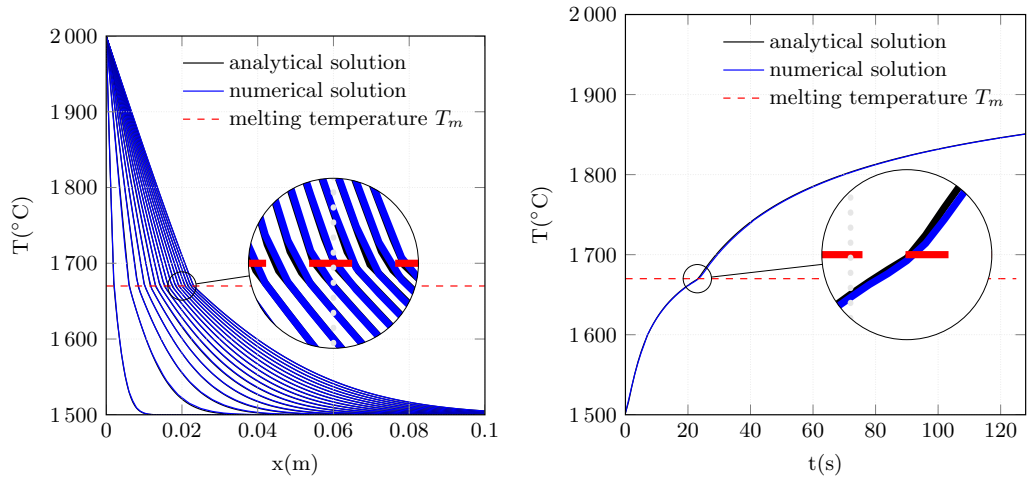
3. Modelling Selective Laser Melting

Goal of this section is to present a method that discretizes dynamically growing structures and in which refinements are only carried out where necessary. To this end, we first introduce the finite cell method, an embedded domain method for high-order finite elements, before moving on to a show-case example demonstrating the features of this approach.

3.1. Finite Cell Method

The main objective of the finite cell method is to avoid boundary conforming meshing of geometrically complex physical domains. To this end, a geometrically complex domain Ω_{phy} is extended by a fictitious domain Ω_{fict} such that the resulting domain Ω has a simple shape and can thus be meshed easily. (see fig. 10) and [26, 27].

Journal publication: https://doi.org/10.1016/j.camwa.2017.11.014



(a) Temperature distribution along x-direction at $t = 1, 9, 17, 25...129$[s]

(b) Temperature evolution at $x = 0.01$[m]

Figure 9: Melting bar example: evolution of temperature in time and space

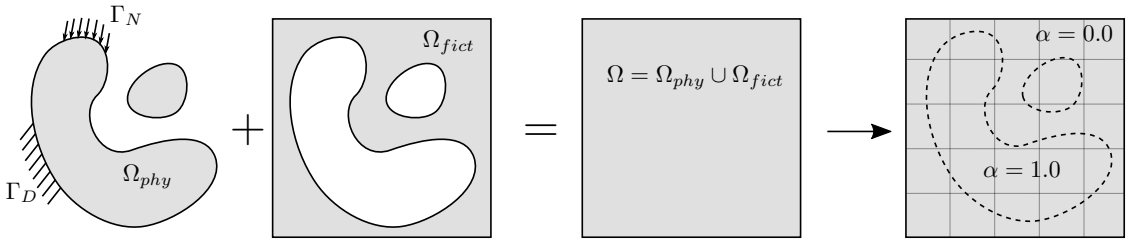


Figure 10: Concept of Finite Cell Method

In the simplest case, the mesh is a grid whose entities are called cells, henceforth the name finite cell method. It is on these cells where the shape functions are spanned. The original geometry of the domain is recovered at integration level by use of the following indicator function

$$\alpha = \begin{cases} 1 & \forall \mathbf{x} \in \Omega_{phy} \\ 10^{-g} & \forall \mathbf{x} \in \Omega_{fict} \end{cases} \tag{30}$$

where, ideally $g \to \infty$, although in practical applications it is usually sufficient to choose $g = 4$. The equality of a conforming to a non-conforming Galerkin formulation can easily be shown. For example for

Journal publication: https://doi.org/10.1016/j.camwa.2017.11.014

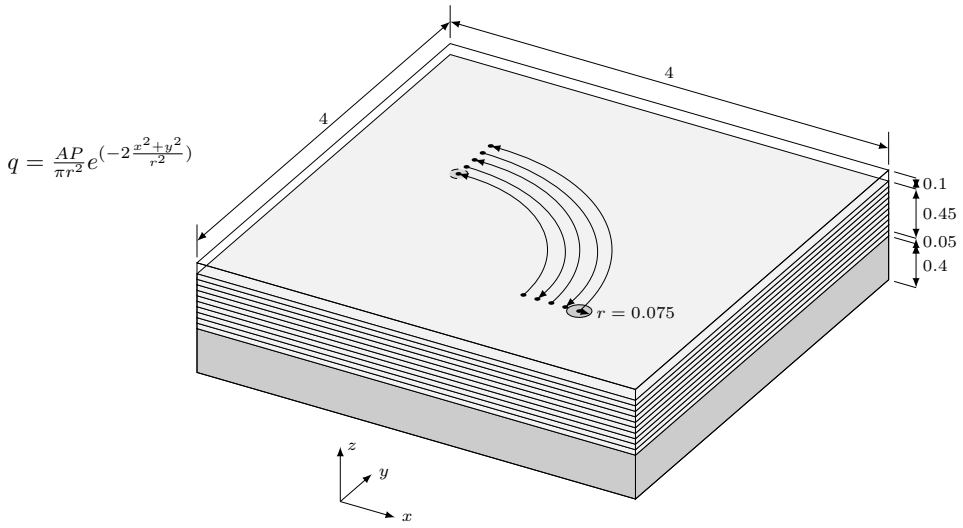


Figure 11: Setup of the process model

the volumetric term of K_{ij} given in eq. (14) it holds:

$$\begin{aligned} K_{ij} &= \int_{\Omega_{phy}} \nabla N_i \cdot (k \nabla N_j)\, \mathrm{d}\Omega_{phy} \\ &\approx \int_{\Omega_{phy}} \nabla N_i \cdot (1\, k \nabla N_j)\, \mathrm{d}\Omega_{phy} + \int_{\Omega_{fict}} \nabla N_i \cdot (10^{-g}\, k \nabla N_j)\, \mathrm{d}\Omega_{fict} \\ &= \int_{\Omega} \nabla N_i \cdot (\alpha\, k \nabla N_j)\, \mathrm{d}\Omega. \end{aligned} \tag{31}$$

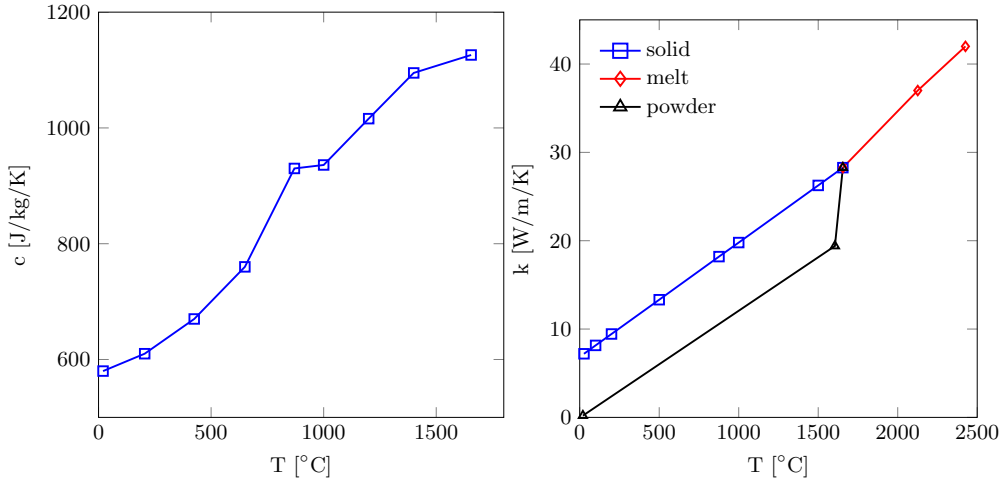
All other terms involving volume integrals in eq. (14) can be treated likewise. The convergence of this scheme is mathematically proven in [28] where it is additionally shown that the influence of a non-zero α is proportional to a (controllable) modeling error.

The discontinuity of α necessitates adaptive integration schemes, see e.g. [29, 30] for a recent overview of possible schemes. The simplest (although not most efficient) choice is a composed integration by means of an octree. This variant will be used to compute the examples in this section.

3.2. Multi-level hp method and the finite cell method at work

We now consider the computational modeling of a SLM process as depicted in fig. 11. The computational domain consists of a solid base plate upon which one powder layer resides. A laser then solidifies the powder along the path specified in the illustration. A new layer of powder is added and the process repeats until 10 layers are completed. Each layer has a thickness of 50[μm]. The three phases powder, solid and melt are assigned the temperature dependent material coefficients given in fig. 12. The dependency of the heat capacity is assumed to be the same for all three phases, while the conductivity is assigned individually to each phase. The initial temperature of deposited material and base plate is $T = 200^oC$. Radiation and convection boundary conditions were applied at the top surface using an emissivity of $\epsilon = 0.8$ and a convection coefficient of $h_{conv} = 5.7$ [W/m^2°C]. Homogeneous Neumann boundary conditions are applied elsewhere.

Journal publication: https://doi.org/10.1016/j.camwa.2017.11.014



(a) heat capacity of the solid

(b) conductivity of solid, powder and melt

Figure 12: Temperature dependent material properties

The discretizational treatment of the process itself is best explained by considering a time-step of the simulation process. Two grids are used. The grid depicted in fig. 13a describes the material in a voxel-like fashion, while the other grid depicted in fig. 13b spans the high-order shape functions used for finite cell discretization of the temperature.

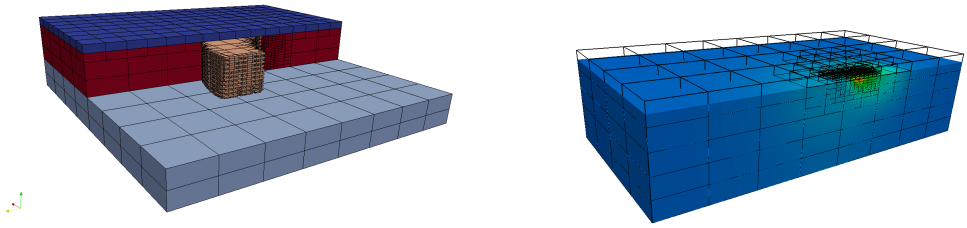
On the material side, four types of domains are to be distinguished: air, powder, solid/liquid and the base-plate. The distinction between air and powder is modeled by using the α defined by the finite-cell method (see 30). This interface is explicitly defined as a geometric input. The change between powder and melt, however, emerges as a result of the power input by the laser beam. This locally emerging change in material properties is modeled similar to the bar example presented in section 2.4.2. The difference is that once powder has changed to melt it cannot change back to powder; it can only vary between melt and solid thereafter.

The grid which spans the basis functions discretizing the temperature field depicted in Figure 13b initially consists of 8x8x5 base finite cells. It is refined by recursively bisecting the elements three times towards a (moving) bounding box in the close proximity of the impact point of the laser using the multi-level *hp*-method discussed in 2.3. The smallest elements have an element size of half of the layer thickness in z-direction and 62.5[μm]in in-plain direction. This corresponds to 2.4^2 finite elements of order $p = 3$ at the impact point of the laser which. Under the assumption that the accuracy scales with the number of elements within the impact point of the laser and the chosen polynomial degree as studied in section 2.4.1, it is possible to obtain a rough estimate on the accuracy of the computation. In that case the same resolution is obtained for fife multi-level refinements which led to an accuracy of approx. 4[%] at $p = 3$ (see fig. 7a black line with pentagon symbols). This is considered to be in the range of other modeling errors which are even more difficult to track but naturally occur in the modeling of powder bed fusion processes.

The base level of the grid describing the material coefficients is geometrically and topologically congruent to the one used for the temperature discretization but both grids refine and de-refine independently of one another. The maximum refinement of the grid discretizing the state variables is one level finer than the thermal counterpart. It refines towards sudden changes in the material coefficients. This grid is used for a partitioned integration of the bilinear forms. The emerging structure (logged in that grid) is depicted along with the temperature in all physical domains at the representative time steps 220, 1000 and 1670 in figs. 14a to 14c, respectively.

At this point it is interesting to note the difference to other approaches common in the modelling

Journal publication: https://doi.org/10.1016/j.camwa.2017.11.014



(a) Discretization of the material coefficients: air in dark-blue, powder in red, solidified domain in brown and base plate in light blue

(b) Discretization of the temperature field on its the computational domain without voxels containing air in a cut through half of the domain to better visualize the refinement in the vicinity of the laser impact.

Figure 13: Discretization of material and temperature by means of two grids

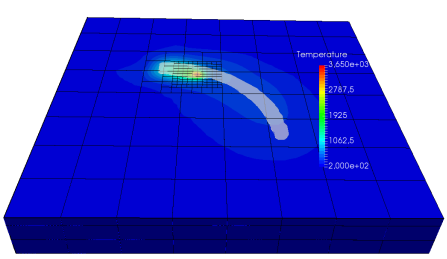
of powder bed fusion processes or metal deposition: the quiet element method and the inactive element method. A comprehensive overview of both strategies is found e.g. in [31, 32, 33] including a discussion of each methods' advantages and disadvantages. In essence, the inactive element method activates elements in the sense of including them in the global stiffness matrix only if material was deposited in the region covered by the element. In the quiet element method, finite elements are active throughout all time steps of the simulation but are assigned small conductivities and capacities if no material is present. In this sense, the presented methodology is more related to the quite element method because regions with no material are assigned low material properties. The difference lies in the fact that by using the finite cell method sub-regions within finite cells contain material while other regions of the cell may still be void. In the presented example, in the region of maximum refinement each layer consists of four layers of voxels. In turn, four layers are themselves contained in one finite cell at base level. One finite cell, thus, contains up to 16 voxels in z-direction. These voxels don't contribute to the number of degrees of freedom to be solved for. Nevertheless, they increase the resolution of the material properties provided that high-order shape functions are used to discretize the cells. As a consequence, a comparatively low number of degrees of freedom suffices for an accurate description of the field variables, e.g. the transient temperatures on evolving domains. The large gradients in the solution are captured accurately by using the multi-level hp-method and the necessary refinements can be kept local to the impact region of the laser beam. Figure 14d depicts the number of degrees of freedom for each time step. It varies between six- and eight thousand and increases only marginally throughout the process. The periodic spikes occur at time steps where the laser jumps from one scan path to another while the large plateaus show the change from one layer to another. The complete computation took approximately 10 hours for 2000 time steps on a standard desktop computer whereby only 45 minutes cpu time were actually used for solving the resulting non-linear equation system. This clearly indicates that there is room for optimizations.

4. Conclusions

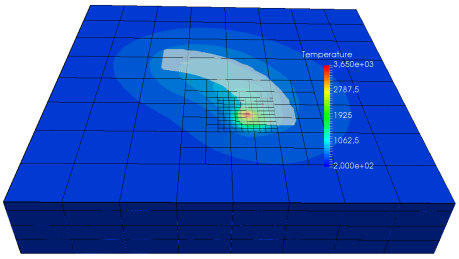
The article at hand presents a computational framework for the simulation of powder bed fusion processes. The scheme is motivated by the fact that the very strong temperature gradients introduced locally by the laser beam quickly diffuse away while the state of the material does not diffuse. Therefore, the discretization of the temperature field is separated from the discretization of the material. These two separate meshes can then refine and coarsen independently of each other. The computational methodology is verified against two (semi-)analytical benchmarks. It is demonstrated that the combination of local refinements and high polynomial degree of the discretizations leads to higher accuracies then only decreasing the mesh size.

The closing example serves to demonstrate the discretizational flexibility of the method for the simulation of the temperature evolution and the phase changes involved in SLM processes. Herein, the material layers do not conform to the discretization of the temperature field and the number of the degrees of freedom are

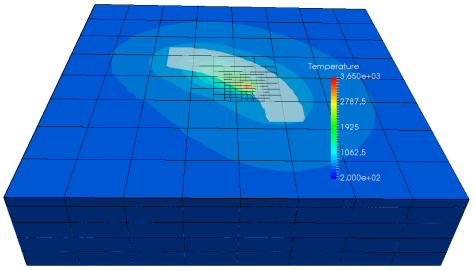
Journal publication: https://doi.org/10.1016/j.camwa.2017.11.014



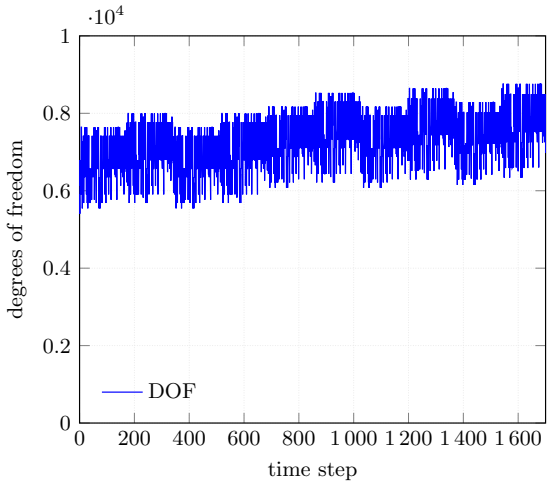
(a) time step 220



(b) time step 1000



(c) time step 1670



(d) DOF over time

Figure 14: Temperature field and its discretization with emerging structure at different time steps and number of degrees of freedom for all time steps throughout the process

Journal publication: https://doi.org/10.1016/j.camwa.2017.11.014

decoupled from the length of the laser scan path. This flexibility in the discretization allows for a practically constant number of degrees of freedom throughout the entire the computation.

Future research will be directed into extending the methodology to include multi-physical capabilities such as the computation of thermo-elasto-plastic phenomena in multi-layer processes.

Acknowledgements

The authors gratefully acknowledge the financial support of the German Research Foundation (DFG) under Grant RA 624/27-1.

[1] W. D. Rolph and K.-J. Bathe, "An efficient algorithm for analysis of nonlinear heat transfer with phase changes," International Journal for Numerical Methods in Engineering, vol. 18, no. 1, pp. 119–134, 1982.
[2] I. Babuška and B. Q. Guo, "The h, p and h-p version of the finite element method; basis theory and applications," Advances in Engineering Software, vol. 15, no. 3, pp. 159–174, 1992.
[3] L. Demkowicz, Computing with Hp-Adaptive Finite Elements, Vol. 1: One and Two Dimensional Elliptic and Maxwell Problems. Applied mathematics and nonlinear science series, Boca Raton: Chapman & Hall/CRC, 2007.
[4] D. Riedlbauer, P. Steinmann, and J. Mergheim, "Thermomechanical finite element simulations of selective electron beam melting processes: Performance considerations," Computational Mechanics, vol. 54, pp. 109–122, July 2014.
[5] J. Irwin and P. Michaleris, "A line heat input model for additive manufacturing," Journal of Manufacturing Science and Engineering, vol. 138, no. 11, 2016.
[6] Q. Wang, J. Li, M. Gouge, A. R. Nassar, P. P. Michaleris, and E. W. Reutzel, "Physics-Based Multivariable Modeling and Feedback Linearization Control of Melt-Pool Geometry and Temperature in Directed Energy Deposition," Journal of Manufacturing Science and Engineering, vol. 139, no. 2, p. 021013, 2017.
[7] W. Rachowicz and L. Demkowicz, "An hp-adaptive finite element method for electromagnetics: Part 1: Data structure and constrained approximation," Computer Methods in Applied Mechanics and Engineering, vol. 187, pp. 307–335, June 2000.
[8] W. Rachowicz, D. Pardo, and L. Demkowicz, "Fully automatic hp-adaptivity in three dimensions," Computer Methods in Applied Mechanics and Engineering, vol. 195, pp. 4816–4842, July 2006.
[9] Hermes, "Hermes - Higher-Order Modular Finite Element System," user's guide, University of Reno, Nevada, USA, 2016.
[10] P. Šolín and J. Červený, "Automatic hp-Adaptivity with Arbitrary-Level Hanging Nodes," Tech. Rep. Research Report No. 2006-07, The University of Texas at El Paso, Department of Mathematical Sciences, 2006.
[11] N. Zander, T. Bog, M. Elhaddad, F. Frischmann, S. Kollmannsberger, and E. Rank, "The multi-level hp-method for three-dimensional problems: Dynamically changing high-order mesh refinement with arbitrary hanging nodes," Computer Methods in Applied Mechanics and Engineering, vol. 310, pp. 252–277, Oct. 2016.
[12] D. Celentano, E. Oñate, and S. Oller, "A temperature-based formulation for finite element analysis of generalized phase-change problems," International Journal for Numerical Methods in Engineering, vol. 37, no. 20, pp. 3441–3465, 1994.
[13] W. Bangerth, R. Hartmann, and G. Kanschat, "deal.II—A general-purpose object-oriented finite element library," ACM Transactions on Mathematical Software, vol. 33, pp. 1–27, Aug. 2007. 00582.
[14] "Nektar++, A Spectral/hp Element Framework."
[15] W. F. Mitchell, "PHAML." `http://math.nist.gov/phaml/phaml.html`.
[16] P. Karban, F. Mach, P. Kůs, D. Pánek, and I. Doležel, "Numerical solution of coupled problems using code Agros2D," Computing, vol. 95, pp. 381–408, May 2013. bibtex karban_numerical:2013.
[17] L. Dalcin, N. Collier, P. Vignal, A. M. A. Côrtes, and V. M. Calo, "PetIGA: A framework for high-performance isogeometric analysis," Computer Methods in Applied Mechanics and Engineering, 2016.
[18] 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, vol. 55, pp. 499–517, Feb. 2015.
[19] N. Zander, M. Ruess, T. Bog, S. Kollmannsberger, and E. Rank, "Multi-level hp-adaptivity for cohesive fracture modeling," International Journal for Numerical Methods in Engineering, vol. in press, 2016.
[20] P. Di Stolfo, A. Schröder, N. Zander, and S. Kollmannsberger, "An easy treatment of hanging nodes in hp-finite elements," Finite Elements in Analysis and Design, vol. 121, pp. 101–117, Nov. 2016.
[21] N. T. Nguyen, A. Ohta, K. Matsuoka, N. Suzuki, and Y. Maeda, "Analytical solutions for transient temperature of semi-infinite body subjected to 3-D moving heat sources," WELDING JOURNAL-NEW YORK-, vol. 78, pp. 265–s, 1999. 00148.
[22] L. Shampine, "Vectorized adaptive quadrature in MATLAB," Journal of Computational and Applied Mathematics, vol. 211, pp. 131–140, Feb. 2008.
[23] H. Weber, "Die Partiellen Differentialgleichungen der Mathematischen Physik." `https://ia601408.us.archive.org/4/items/diepartiellendi00riemgoog/diepartiellendi00riemgoog.pdf`, 1912.
[24] H. Hu and S. A. Argyropoulos, "Mathematical modelling of solidification and melting: A review," Modelling and Simulation in Materials Science and Engineering, vol. 4, no. 4, p. 371, 1996.
[25] D. W. Hahn, Heat Conduction. Hoboken, N.J: Wiley, 3rd ed ed., 2012.
[26] J. Parvizian, A. Düster, and E. Rank, "Finite cell method," Computational Mechanics, vol. 41, pp. 121–133, Apr. 2007.
[27] 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, vol. 197, pp. 3768–3782, Aug. 2008.

Journal publication: https://doi.org/10.1016/j.camwa.2017.11.014

[28] M. Dauge, A. Düster, and E. Rank, "Theoretical and Numerical Investigation of the Finite Cell Method," Journal of Scientific Computing, vol. 65, pp. 1039–1064, Mar. 2015. 00000.
[29] A. Abedian, J. Parvizian, A. Düster, H. Khademyzadeh, and E. Rank, "Performance of Different Integration Schemes in Facing Discontinuities in the Finite Cell Method," International Journal of Computational Methods, vol. 10, p. 1350002, June 2013. 00027.
[30] L. Kudela, N. Zander, S. Kollmannsberger, and E. Rank, "Smart octrees: Accurately integrating discontinuous functions in 3D," Computer Methods in Applied Mechanics and Engineering, vol. 306, pp. 406–426, July 2016.
[31] P. Michaleris, "Modeling metal deposition in heat transfer analyses of additive manufacturing processes," Finite Elements in Analysis and Design, vol. 86, pp. 51–60, Sept. 2014. 00006.
[32] M. Chiumenti, M. Cervera, A. Salmi, C. Agelet de Saracibar, N. Dialami, and K. Matsui, "Finite element modeling of multi-pass welding and shaped metal deposition processes," Computer Methods in Applied Mechanics and Engineering, vol. 199, pp. 2343–2359, Aug. 2010.
[33] L.-E. Lindgren and E. Hedblom, "Modelling of addition of filler material in large deformation analysis of multipass welding," Communications in numerical methods in engineering, vol. 17, no. 9, pp. 647–657, 2001.

Journal publication: https://doi.org/10.1016/j.camwa.2017.11.014

4.4.2 Residual stresses

Further, the computational model was also extended to involve the mechanical behavior in order to compute residual stresses which are introduced due to the rapid cooling in the close vicinity of the melt pool. The corresponding publication which explains the modeling details of this approach is included next.

Clearly, the question whether a physical model is valid or not must be answered by checking the computational results against the physical phenomena observed in experiments in a quantitative manner. This is an involved process, since measurements are very difficult to obtain for the small physical scales at hand. For this reason, the validation of the chosen elasto-plastic model to compute the residual stresses was not carried out against the process of selective laser melting but against the process of fusion welding. This manufacturing process is similar in nature to SLM in the sense that liquid metal solidifies locally to form an artifact. However, fusion welding possesses larger time and spatial scales. Strictly spoken, the model presented in section 4.4.2 is, therefore, not yet validated for SLM due to the lack of experimental data at the considered scale.

Journal Publication

title:	Residual stresses in metal deposition modeling: Discretizations of higher order
authors:	A. Özcan, S. Kollmannsberger, J. Jomo, E. Rank
published at:	*Computers & Mathematics with Applications*
publisher:	Elsevier
year:	2018
volume:	in press: currently only available online
doi:	https://doi.org/10.1016/j.camwa.2018.10.027

Residual stresses in metal deposition modeling: discretizations of higher order

A. Özcan[a,*], S. Kollmannsberger[a], J. Jomo[a], E. Rank[a,b]

[a] *Chair for Computation in Engineering, Technische Universität München, Arcisstr. 21, 80333 München, Germany*
[b] *Institute for Advanced Study, Technische Universität München, Lichtenbergstraße 2a, 85748 Garching, Germany*

Abstract

This article addresses the research question if and how the finite cell method, an embedded domain finite element method of high order, may be used in the simulation of metal deposition to harvest its computational efficiency. This application demands for the solution of a coupled thermo-elasto-plastic problem on transient meshes within which history variables need to be managed dynamically on non-boundary conforming discretizations. To this end, we propose to combine the multi-level *hp*-method and the finite cell method. The former was specifically designed to treat high-order finite element discretizations on transient meshes, while the latter offers a remedy to retain high-order convergence rates also in cases where the physical boundary does not coincide with the boundary of the discretization. We investigate the performance of the method at two analytical and one experimental benchmark.

Keywords: *hp* finite elements, finite cell method, welding, metal deposition modeling, additive manufacturing

Contents

*Corresponding author
Email address: ali.oezcan@tum.de (A. Özcan)

Journal publication: https://doi.org/10.1016/j.camwa.2018.10.027

1. Introduction

The process of metal additive manufacturing involves a highly localized heat source which moves over a substrate. Its purpose is to change the state of the deposited metal either from powder to liquid or from solid to liquid such that the added material bonds with the substrate. The subsequent, rapid cooling of the heat affected zone induces undesired residual stresses, a process well understood in welding, see for example [1, 2]. The numerous physical phenomena involved in this process range from the simplified view stated above to models including detailed weld pool dynamics [3], to models resolving the micro-structure evolution in the cooling phase (see e.g. [4, 5, 6]). The necessary compromise between the complexity of the physical model and the ever increasing yet limited resources for its numerical resolution is mainly driven by the concrete question which needs to be answered by the simulation, (see e.g. [7] for a guideline in welding).

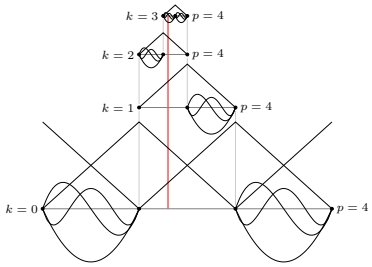
In this context, efficient numerical discretizations are desirable of which the finite element method is the most prominent choice. A simple measure of its complexity is the number of degrees of freedom involved in the computation. These represent the unknown coefficients of piecewise polynomials spanned on a mesh of finite elements which resolve the highly localized gradients. Three basic strategies are available to control the number of degrees of freedom: a) h-refinement, i.e. an increase of the number of finite elements in the mesh, b) p-refinement, i.e. an increase of the polynomial degree of each finite element or c) the combination of both: hp-refinement. The achievable rate of convergence increases from a) to c) for problems with locally high gradients. Unfortunately, so does the complexity of the implementation of the respective method. This is especially the case in transient problems where the necessary adaptions need to be kept local to the (traveling) heat affected zone.

Numerous approaches are reported in literature for locally h-refined discretizations, see e.g. [8] as a representative of an early work on the subject in the context of welding and [9] for a recent review of the available variants applied to metal additive manufacturing including a review of commercial packages for this purpose. Refinements in p beyond quadratic shape functions are scarce, but e.g. [10] demonstrates that it is possible to construct efficient discretizations for elasto-plastic problems by using high order discretizations whose boundary follows the plastic front. The application of hp-adaptive methods in this context is e.g. discussed in [11] where exponential convergence rates were achieved for boundary-conforming discretization. Recently, high-order h refinements have also become of interest in the context of Isogeometric Analysis, see e.g. [12]. All these publications treat boundary conforming discretizations. Extensions to non-boundary conforming finite element discretizations are presented in [13, 14] in the context of the finite cell method. Therein, it is demonstrated that high-order convergence rates may also be achieved even if the physical boundaries of the domain do not coincide with the discretization of the mesh. This is a desirable feature in the application at hand where the physical domain grows with time. Additionally, highly localized gradients need to be followed dynamically though the course of the simulation.

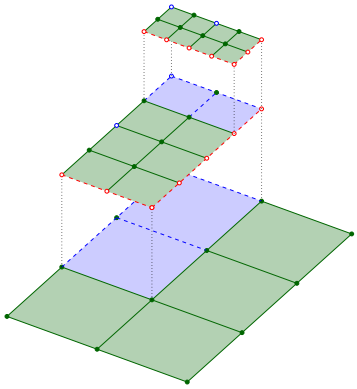
As a remedy for an accurate resolution of transient gradients on non-boundary conforming domains we advocate a combination of both, the multi-level hp-method and the finite cell method. To this end, we first introduce these numerical methods in section 2 where we begin by recalling the multi-level hp-method. It offers a relatively simple management of the degrees of freedom for transient discretizations of hp-type. Additionally, we present some novel but straight-forward extensions necessary for elasto-plastic computations on transient discretizations of hp-type. We then present the computation of elasto-plasticity in the framework of the finite cell method [15] and proceed to a short description of the thermo-elasto-plastic problem in section 2.3 .

After the basics of the numerical methods are presented, we evaluate their performance against three benchmark examples. In section 3.1 a sphere under internal pressure in which a plastic front evolves is investigated. While the exact location of the plastic front is known in this specific case, in a more general setting it would not be unknown. We, thus, chose a grid-like discretization whose boundaries neither coincide with the evolving plastic front nor with the physical boundaries of the problem setup itself. We demonstrate that it is possible to capture the plastic front and the stress states on non-boundary conforming domains using higher order $h-$ refinements and obtain higher order convergence rates. We then move to the thermo-elasto-plastic setting in section 3.2 to make the point that it is also possible to capture the plastic stress states in a coupled setting accurately. The final numerical example presented in section 3.3 is chosen to

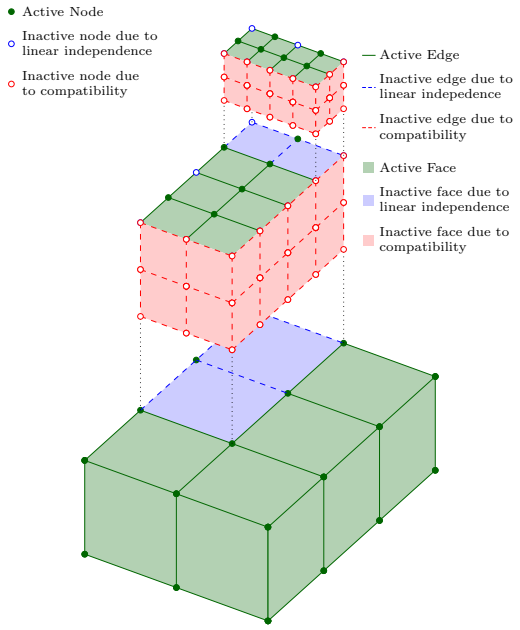
Journal publication: https://doi.org/10.1016/j.camwa.2018.10.027



(a) One-dimensional case



(b) Two-dimensional case



(c) Three-dimensional case

Figure 1: Main idea of the multi-level hp-method in different spatial dimensions [16].

demonstrate that the presented methodology is also capable of reproducing stress states of real experiments. We conclude the article by pointing out the potential and limits of the presented approach in section 4.

2. Numerical Methods

This section serves to introduce and further develop the numerical techniques which are used and evaluated in section 3. To this end, the multi-level *hp*-method is first presented in section 2.1 followed by a description on the finite cell method section 2.2. They are then combined to solve problems in elasto-plasticity as introduced in section 2.3. All these sections form the background for solving thermo-elasto-plastic problems as introduced in section 2.4.

2.1. The multi-level hp-method

The aforementioned multi-level hp-method is a powerful scheme for performing local mesh adaptation in an efficient manner. It was first introduced in [17] and aims at a simple degree of freedom management for dynamic discretizations of high order by performing local refinements based on superposition. For this purpose, an initial discretization of coarse base elements is overlayed locally with multiple layers of finer overlay elements so as to better capture the solution behavior such as locally high gradients, see fig. 1. This is in contrast to standard *hp*-methods which perform refinement by replacement where coarse elements are replaced by a set of smaller elements.

The methodology to overlay finite elements as in the multi-level *hp*-method requires the enforcement of linear independence of the basis functions and compatibility of the ansatz space at the boundary of the refinement zone. This is achieved in a straightforward manner by leveraging the direct association of topological components (nodes, edges, faces, volumes) with the degrees of freedom. Compatibility and linear

Journal publication: https://doi.org/10.1016/j.camwa.2018.10.027

independence are ensured through the deactivation of specific topological components. This deactivation is governed by simple rule-sets, which work for different spatial dimensions [17, 16] alike. This forms a key strength of the method as compared to classic *hp*-methods as the simplicity of the rule-sets allow for an easy treatment of dynamic meshes with arbitrary levels of hanging nodes. The multi-level *hp*-method results in C^0-continuity over the complete refinement hierarchy in base elements while C^∞-continuity is obtained in the leaf elements — elements on the highest refinement level with no children.

2.2. The finite cell method

The ultimate in the scope of this article is to provide a framework which leaves as much geometric and topological freedom for the emerging additively manufactured artifact as possible. At the same time, the effort for mesh generation should be kept as low as possible. These are two main features of immersed methods, a class of advanced discretization techniques that significantly reduce the effort of mesh generation by utilizing a non-boundary conforming domain discretization. They have, therefore, emerged as the method of choice to perform numerical simulations on bodies with a complex shape, topology or a combination of both. The finite cell method (FCM) introduced in [18, 19], is a prominent representative of immersed methods whose core idea is to combine the advantages of the fictitious domain approach with the computational efficiency of discretizations of high order. Its basic idea is depicted in fig. 2 . A body of complex shape and topology defined on a physical domain Ω_{phys} is extended by a fictitious domain Ω_{fict}. Their union yields a computational domain $\Omega = \Omega_{\text{phys}} \cup \Omega_{\text{fict}}$ with a simple boundary which can be discretized using well shaped finite elements. These form the support of the ansatz functions and are termed finite cells as their boundary is not conforming to the boundary of the original, physical boundary $\partial\Omega_{\text{phys}}$. The physical domain must then be recovered at the level of the numerical integration of the associated bi-linear and linear forms. An indicator function α is introduced to classify points belonging to the physical or fictitious domain. Points within the physical domain are assigned a value $\alpha = 1$, whereas points in the fictitious domain have a value $\alpha \ll 1$. While a detailed description of the FCM can be found in [18, 15, 19], the short and concise rehearsal

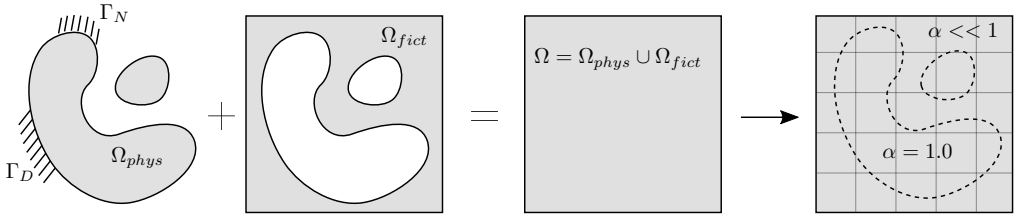


Figure 2: Core idea of the finite cell method.

of its basic ideas above suffices to set the stage for its extension to elasto-plastic problems as laid out in the next section.

2.3. Elasto-plasticity with the multi-level hp method

The stage is now set to develop an efficient elasto-plastic formulation which combines the multi-level *hp*-method with the finite cell method as presented in sections 2.1 and 2.2 .

The finite cell method has already been successfully used in the field of plasticity. The work most relevant to the paper at hand is published in [20] in which the FCM was implemented for the J_2 flow theory with nonlinear, isotropic hardening for small displacements and small strains. Further, in [13] it is demonstrated that the FCM leads to more efficient discretizations than the standard, boundary conforming *h*-version of the finite element method delivers. The FCM has recently also been extended to nearly incompressible finite strain plasticity with complex geometries, see [14]. All these investigations were carried out on static discretizations in the sense that no dynamic refinement was applied locally to capture transient plastic fronts. However, in layered deposition modeling the size of the traveling heat source is comparatively small w.r.t. the rest of the computational domain. In these applications, a dynamic refinement and coarsening offers

Journal publication: https://doi.org/10.1016/j.camwa.2018.10.027

a discretizational advantage because the computational effort can thereby be balanced with the accuracy needed locally. However, this requires a dynamic management of primal and internal variables.

To facilitate a comprehensive but concise presentation of this subject, the next sections start with the classic formulation of J_2 plasticity in 2.3.1. The standard setting is then cast into the FCM formalism in section 2.3.2 before the integration of the multi-level *hp*-method into the elasto-plastic FCM framework is treated in section 2.3.3 along with the associated management of primal and internal variables.

2.3.1. Classic J_2 plasticity

The classic weak form of equilibrium in a solid body is given by

$$G(\boldsymbol{u},\boldsymbol{\eta}) = \int_\Omega \boldsymbol{\varepsilon}(\boldsymbol{\eta}) : \boldsymbol{\sigma}\,\mathrm{d}\Omega - \int_\Omega \boldsymbol{\eta}\cdot\boldsymbol{b}\,\mathrm{d}\Omega - \int_{\Gamma_N} \boldsymbol{\eta}\cdot\boldsymbol{t}\,\mathrm{d}\Gamma = 0, \tag{1}$$

where $\boldsymbol{u}$ is the displacement field, $\boldsymbol{\eta}$ is the test function, $\boldsymbol{\sigma}$ the Cauchy stress tensor, $\boldsymbol{\varepsilon}$ the strain tensor, $\boldsymbol{b}$ are the body forces acting on the domain, Ω, and $\boldsymbol{t}$ is the prescribed traction on the Neumann boundary, Γ_N. When a nonlinear material is utilized, the stress state is not only a function of the instantaneous strain. Instead, the stress state also depends on the history of the loads the body was subjected to. Consequently, the weak form (1) becomes nonlinear and is solved incrementally for each time step $[t_n, t_{n+1}]$. Within each time step the internal variables, $\boldsymbol{\lambda}$, which contain the history of the material, are assumed constant. Linearization of the weak form with respect to the unknown $\boldsymbol{u}$ around $\boldsymbol{u}^{(i)}_{n+1}$, which is the solution at iteration i, is given e.g. in [21] and reads

$$\begin{aligned}\int_\Omega \boldsymbol{\varepsilon}(\boldsymbol{\eta}) : \boldsymbol{D} : \boldsymbol{\varepsilon}(\delta\boldsymbol{u})\,\mathrm{d}\Omega = &-\int_\Omega \boldsymbol{\varepsilon}(\boldsymbol{\eta}) : \boldsymbol{\sigma}_{n+1}(\boldsymbol{\lambda}_n, \boldsymbol{\varepsilon}(\boldsymbol{u}^{(i)}_{n+1}))\,\mathrm{d}\Omega \\ &+\int_\Omega \boldsymbol{\eta}\cdot\boldsymbol{b}_{n+1}\,\mathrm{d}\Omega + \int_{\Gamma_N} \boldsymbol{\eta}\cdot\boldsymbol{t}_{n+1}\,\mathrm{d}\Gamma.\end{aligned} \tag{2}$$

where the fourth order tensor $\boldsymbol{D}$ is the tangent modulus defined as

$$\boldsymbol{D} = \left.\frac{\partial\boldsymbol{\sigma}_{n+1}}{\partial\boldsymbol{\varepsilon}}\right|_{\boldsymbol{\varepsilon}(\boldsymbol{u}^{(i)}_{n+1})}. \tag{3}$$

2.3.2. The elasto-plastic finite cell method

In order to apply the finite cell method, the domain integrals in eq. (2) are multiplied with the indicator function α such that

$$\begin{aligned}\int_\Omega \alpha\,\boldsymbol{\varepsilon}(\boldsymbol{\eta}) : \boldsymbol{D} : \boldsymbol{\varepsilon}(\delta\boldsymbol{u})\,\mathrm{d}\Omega = &-\int_\Omega \alpha\,\boldsymbol{\varepsilon}(\boldsymbol{\eta}) : \boldsymbol{\sigma}_{n+1}(\boldsymbol{\lambda}_n, \boldsymbol{\varepsilon}(\boldsymbol{u}^{(i)}_{n+1}))\,\mathrm{d}\Omega \\ &+\int_\Omega \alpha\,\boldsymbol{\eta}\cdot\boldsymbol{b}_{n+1}\,\mathrm{d}\Omega + \int_{\Gamma_N} \boldsymbol{\eta}\cdot\bar{\boldsymbol{t}}_{n+1}\,\mathrm{d}\Gamma.\end{aligned} \tag{4}$$

Since the solution in the fictitious domain is unphysical, computing the tangent modulus or stresses in the fictitious domain causes unnecessary computational overhead. Therefore, the deformation in the fictitious domain is neglected and stresses are assumed to be zero ($\boldsymbol{\sigma}_{fict} = \boldsymbol{0}$). Moreover, the tangent modulus is taken as the elastic tangent ($\boldsymbol{D}_{fict} = \boldsymbol{D}^e$). Incorporating these assumptions into eq. (4) provides the following

Journal publication: https://doi.org/10.1016/j.camwa.2018.10.027

1. Linear elastic law:

$$\boldsymbol{\sigma} = \boldsymbol{D}^e : \boldsymbol{\varepsilon}^e$$

2. Yield function:

$$\Phi = \Phi(\boldsymbol{\sigma}, \bar{\varepsilon}^p) = \sqrt{3\, J_2(\boldsymbol{s}(\boldsymbol{\sigma}))} - \sigma_y(\bar{\varepsilon}^p)$$

3. Plastic flow rule:

$$\dot{\boldsymbol{\varepsilon}}^p = \dot{\gamma}\sqrt{\frac{3}{2}}\,\frac{\boldsymbol{s}}{\sqrt{\boldsymbol{s} : \boldsymbol{s}}}$$

4. Hardening law:

$$\dot{\bar{\varepsilon}}^p = \dot{\gamma} = \sqrt{\frac{2}{3}\,\dot{\boldsymbol{\varepsilon}}^p : \dot{\boldsymbol{\varepsilon}}^p}.$$

5. Kuhn-Tucker conditions:

$$\Phi \leq 0, \quad \dot{\gamma} \geq 0, \quad \dot{\gamma}\Phi = 0$$

Figure 3: Summary of the rate-independent von Mises associative model with nonlinear isotropic hardening

linearized weak form of equilibrium for elasto-plastic problems

$$\begin{aligned} &\int_{\Omega_{\mathrm{phys}}} \boldsymbol{\varepsilon}(\boldsymbol{\eta}) : \boldsymbol{D} : \boldsymbol{\varepsilon}(\delta\boldsymbol{u})\,\mathrm{d}\Omega + \int_{\Omega_{\mathrm{fict}}} \alpha\,\boldsymbol{\varepsilon}(\boldsymbol{\eta}) : \boldsymbol{D}^e : \boldsymbol{\varepsilon}(\delta\boldsymbol{u})\,\mathrm{d}\Omega = \\ &- \int_{\Omega_{\mathrm{phys}}} \boldsymbol{\varepsilon}(\boldsymbol{\eta}) : \boldsymbol{\sigma}_{n+1}(\boldsymbol{\lambda}_n, \boldsymbol{\varepsilon}(\boldsymbol{u}^{(i)}_{n+1}))\,\mathrm{d}\Omega + \int_{\Omega} \alpha\,\boldsymbol{\eta}\cdot\boldsymbol{b}_{n+1}\,\mathrm{d}\Omega + \int_{\Gamma_N} \boldsymbol{\eta}\cdot\bar{\boldsymbol{t}}_{n+1}\,\mathrm{d}\Gamma. \end{aligned} \tag{5}$$

In this paper, a rate-independent von Mises plasticity model with nonlinear isotropic hardening is considered whereby the displacements and strains are assumed to be small. The classic, additive decomposition of the strain tensor into an elastic and a plastic counterpart is then applied such that

$$\boldsymbol{\varepsilon} = \boldsymbol{\varepsilon}^e + \boldsymbol{\varepsilon}^p. \tag{6}$$

Figure 3 summarizes this model, in which the internal variables $\boldsymbol{\lambda}$ are given by

$$\boldsymbol{\lambda} = \{\, \boldsymbol{\varepsilon}^p, \bar{\varepsilon}^p \,\}, \tag{7}$$

where $\bar{\varepsilon}^p$ is the equivalent plastic strain.

2.3.3. Multi-level hp-adaptivity and the transfer of primary and internal variables

In the case of dynamic multi-level *hp*-refinements the primary variables given by the displacement field, $\boldsymbol{u}$, and internal variables given by $\boldsymbol{\lambda}$, need to be transferred from the old discretization prior to the refinement to the new discretization after the refinement was carried out. Since the displacement field over the domain is discretized by the basis functions, a global C^0 continuous description is available which is directly transferred to the new discretization by means of a global L_2-projection. However, in the general elasto-plastic finite

Journal publication: https://doi.org/10.1016/j.camwa.2018.10.027

element procedure, the evolution of the internal variables is computed at *local* integration points via the plastic flow rule and the hardening law. As no C^0 continuous discretization is readily available, the transfer of these variables from the integration points of the old discretization to the integration points of the new one is more involved.

As a remedy, several strategies have been developed in literature. The simplest is the point-wise transfer of the internal variables [22]. Therein, an area is associated to each old integration point within which constant values of the history variables are assumed. This leads to a discontinuous approximation of the history variables within the concerned finite elements. Another strategy is the element-wise transfer [23], where the internal variables are interpolated by local functions. This strategy can result in an approximation with discontinuities over element boundaries. A cure is offered by the large group of strategies involving nodal projections. Therein the internal variables are first transferred from integration points of the old discretization to the nodal degrees of freedom of that old discretization. This is achieved in [22] and [24] by extrapolating the values from integration points to the nodal dofs for each element locally and later averaging these extrapolated values. Another possibility is to use the super convergent patch recovery technique as presented in [25, 26, 27, 28], for the transfer of the values from the integration points of the old mesh to its nodes. In this approach a patch consists of elements that surround a node and the internal variables are fitted to continuous polynomials over each patch by a least squares method and are then interpolated to the nodal points. Independent of how the values are obtained at the nodes of the old discretization, the next step is to interpolate them to the new nodal points in the new mesh. Finally, the values of the internal variables at the new integration points are interpolated from the values of the new nodal points. Since this strategy acts as a smoothing operator, the localized internal variables are smeared out over a larger part of the domain. This can be a drawback in elasto-plastic analysis, because plasticity is a local effect in most cases.

Unfortunately, none of these techniques are suitable for the multi-level *hp*-adaptivity. This is mainly due to the fact that the high order basis functions are modal functions, which are not associated to specific nodes. For this reason a modified version of the element-wise transfer [23] is advocated in the paper at hand in which each component of the internal variables, λ_i, is approximated such that

$$\lambda_i \approx \lambda_i^h = \mathbf{P}(r,s,t)\,\mathbf{c} \tag{8}$$

where $\mathbf{P}$ is a vector of integrated Legendre polynomials as used in p-version of FEM [29] and $\mathbf{c}$ is the corresponding coefficient vector. It is explicitly pointed out that these polynomials only span the leaf element, where (r,s,t) denotes the local coordinates of this element. The coefficient vector, $\mathbf{c}$ is determined by applying a least square fit to the discrete integration point values, $\lambda_i^d(r_j,s_j,t_j)$, inside the leaf element. This is done by minimizing the function, $F(\mathbf{c})$,

$$F(\mathbf{c}) = \sum_{j=1}^{n_{gp}} \left(\lambda_i^d(r_j,s_j,t_j) - \mathbf{P}(r_j,s_j,t_j)\,\mathbf{c}\right)^2, \tag{9}$$

which is carried out by differentiating F with respect to $\mathbf{c}$ as

$$\frac{\partial F}{\partial \mathbf{c}} = \mathbf{b} - \mathbf{A}\,\mathbf{c} = 0, \tag{10}$$

with

$$\mathbf{A} = \sum_{j=1}^{n_{gp}} \mathbf{P}^T(r_j,s_j,t_j)\,\mathbf{P}(r_j,s_j,t_j), \quad \mathbf{b} = \sum_{j=1}^{n_{gp}} \mathbf{P}^T(r_j,s_j,t_j)\,\lambda_i^d(r_j,s_j,t_j). \tag{11}$$

Equation (10) leads to the following linear system of equations

$$\mathbf{A}\,\mathbf{c} = \mathbf{b}, \tag{12}$$

which needs to be solved for each component of the internal variables.

Journal publication: https://doi.org/10.1016/j.camwa.2018.10.027

If this projection strategy is applied to the elements that contain the plastic front, i.e. in which only a group of integration points in the element accumulated plastic strains, then the proposed least squares approximation may lead to oscillations and spurious values. In some cases it is even possible to obtain invalid negative values for the equivalent plastic strain. Figure 4 illustrates this for the polynomial order $p = 4$ for a simple one dimensional example in which the equivalent plastic strain should be nonzero for the two plastified integration points and zero for the three elastic integration points. This function cannot be represented by polynomials and the well known Gibbs phenomenon occurs.

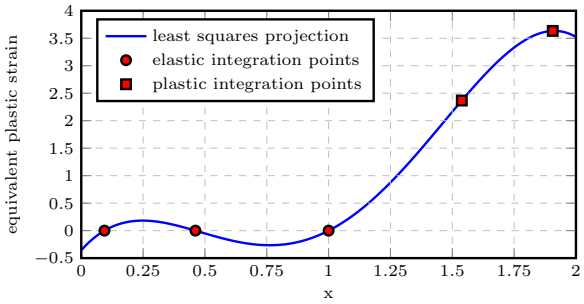


Figure 4: Convergence

In order to circumvent this problem, the values in those elements containing a elastic-plastic interface are not projected with the least squares method. Instead, for these elements we advocate the following approach depicted in Figure 5a. Four cases are distinguished and characterized by the location of the integration points w.r.t. the concerned interpolation point. The first case is given in Figure 5b where the interpolation point p_1 is located inside the gray bounding box defined by all integration points. In this case the internal variables, λ_i at p_1 are approximated by a trilinear interpolation from the values of the eight integration points closely surrounding it. In all the other cases the interpolation point p is located outside the gray bounding box. Therefore, its projection onto the corresponding surface, edge or corner of the bounding box, p^*, is used for interpolation. Figure 5c depicts the second case in which the projection of p_2 is surrounded by only four points. These are then used for bilinear interpolation. The third case occurs when the projected point is surrounded by only two points as demonstrated in Figure 5d. In this case linear interpolation is applied. The last case occurs when the original interpolation point lies in the corner quadrants of the finite cell. In this case values at the closest integration point are used. We will demonstrate in section 3 that this approach is feasible for dynamic multi-level *hp*-discretizations in an elasto-plastic setting.

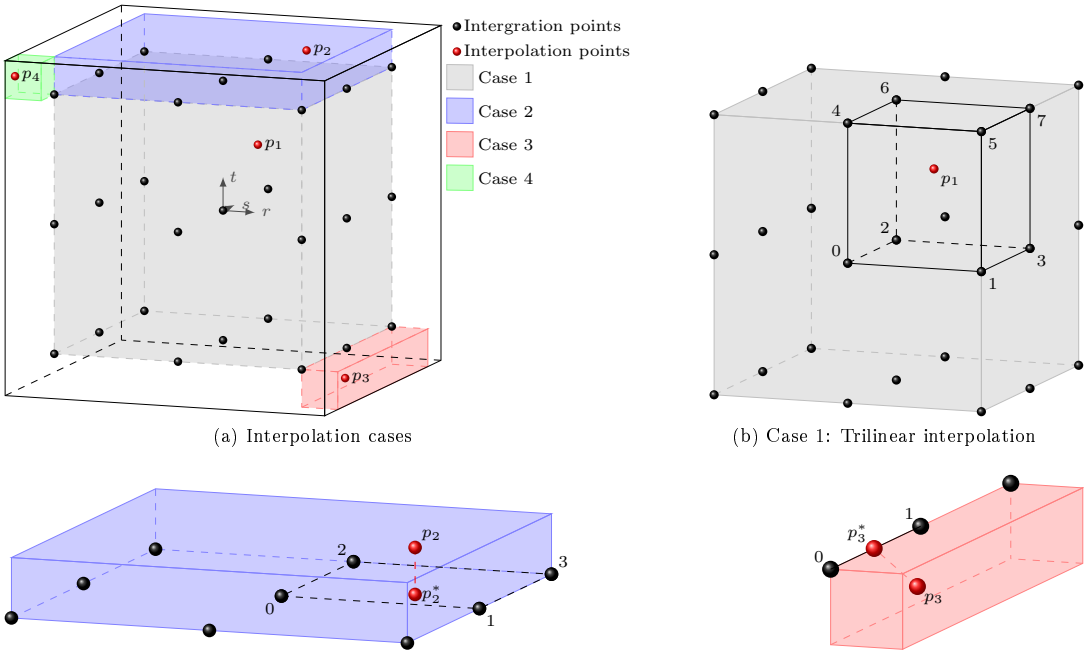
2.4. Coupling to thermal problems

Metal deposition is a multi-physics problem, where the highest temperature gradients as well as the phase changes occur in close vicinity of the moving laser beam. The phase changes between liquid and solid states are simulated by the model introduced by Celentano et. al. [30] which uses the discretized weak form of equation (14). Further details regarding this method and its application to thermal analysis of the selective laser melting process in the framework of multi-level *hp*-adaptivity can be found in [31].

$$\rho c \frac{\partial T}{\partial t} + \rho L \frac{\partial f_{pc}}{\partial t} - \nabla \cdot (k \nabla T) = Q \tag{13}$$

A dynamically adaptable data container (multi-level grid) is used to keep track of the physical domain Ω_{phys} during the deposition process. The container represents a dynamic octree. It stores the current material state of a voxel at all points in time. Material states are stored in this container and not associated to finite elements. This decoupling of material and discretization facilitates the use of the finite cell method.

Journal publication: https://doi.org/10.1016/j.camwa.2018.10.027



(a) Interpolation cases

(b) Case 1: Trilinear interpolation

(c) Case 2: Bilinear interpolation

(d) Case 3: Linear interpolation

Figure 5: Interpolation in interface element

Due to the changes in the temperature during the process, some regions in the domain expand, while others contract. This generates residual stresses in the part when the thermal load is removed and the material allowed to cool down to its initial temperature. These stresses are computed by the quasi-static mechanical model, given in Section 2.3 with the addition of the thermal strain defined as

$$\boldsymbol{\varepsilon}^{th} = \gamma \, \Delta T \, \boldsymbol{I}, \tag{14}$$

where γ is the thermal expansion coefficient and $\boldsymbol{I}$ is the second order identity tensor. Adding the thermal strain component extends the additive decomposition of strains given in equation (6) to

$$\boldsymbol{\varepsilon} = \boldsymbol{\varepsilon}^{e} + \boldsymbol{\varepsilon}^{p} + \boldsymbol{\varepsilon}^{th}. \tag{15}$$

It should be noted that, the von Mises plasticity depends solely on the deviatoric part of the strain tensor and the thermal strain is purely hydrostatic. Therefore, the basic structure of the elasto-plastic model introduced in Section 2.3 remains as is. Only the additive decomposition of the strains changes along with the fact that the yield stress and the thermal expansion coefficient are now temperature dependent functions.

It is assumed that displacements are small and do not produce heat, so that only a one-directional coupling has to be taken into account, i.e. only the displacement field is affected by the changes in the temperature field. As shown in Figure 6, a staggered approach is taken for the solution of the thermo-mechanically coupled problem. For each time step, the multi-level grid is first updated according to the metal deposition such that the physical domain, Ω_{phys}, is defined for both the thermal and the mechanical problems. Then, the thermal problem is solved to obtain the temperature distribution. Finally, the resulting temperature field is used to compute the thermal strains and temperature-dependent material properties used to solve the mechanical problem before the next time step is computed.

Journal publication: https://doi.org/10.1016/j.camwa.2018.10.027

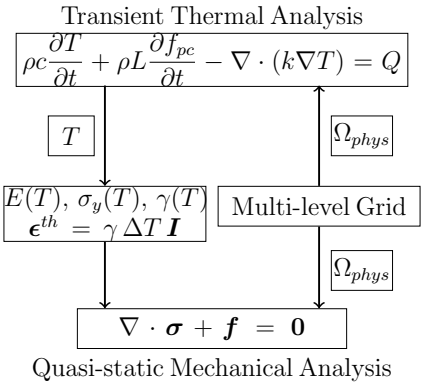


Figure 6: Thermomechanical coupling

3. Numerical Examples

All presented examples have benchmark character and are chosen thoroughly to test the main aspects of the proposed methodology. The first example discussed in section 3.1 is chosen to test if higher order convergence rates are possible in elasto-plastic computations and under what circumstances they decay. To this end, a dynamic multi-level *hp*-refinement as presented in section 2.1 is carried out towards the plastic front in several load steps whereby the plastic front itself is not resolved in a boundary conforming manner but travels through finite cells. This situation is expected for engineering applications. The example also tests the finite cell formulation presented in section 2.2 in the elasto-plastic embedded domain setting as described in section 2.3 by resolving the boundary of the physical domain only on integration level. It also serves as a test for the transfer of history variables as detailed in section 2.3.3 .

The second example given in section 3.2 is chosen to verify the thermo-elasto-plastic coupling procedure laid out in section 2.4 . Both fields are discretized on separate computational grids which are refined and de-refined individually according to the requirements of the corresponding field variable.

The final example presented in section 3.3 tests the combination of the methodology on a experimental benchmark used in welding. As such, it constitutes a first step towards a verification of the proposed methodology for layer deposition modeling.

3.1. Internally pressurized spherical shell

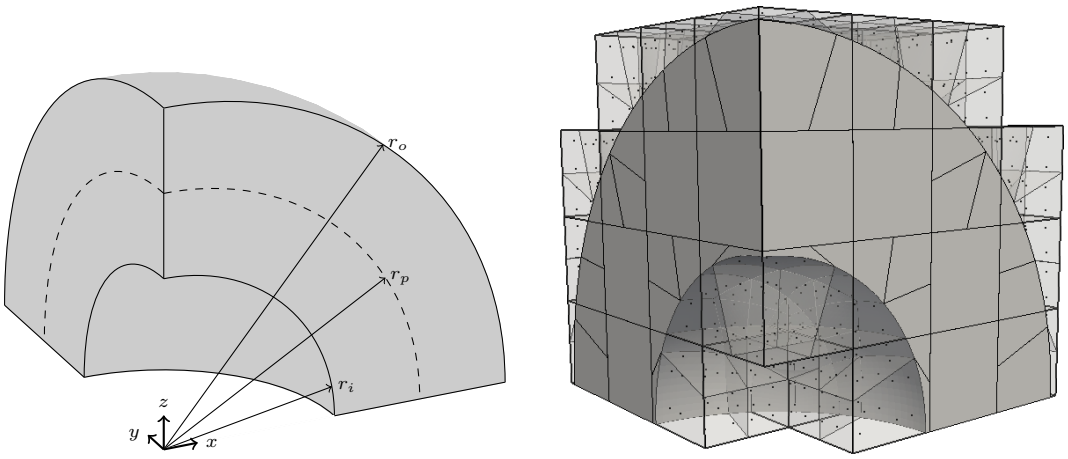
In this example internal pressure, P, is applied in increments to a spherical shell, which is composed of an elastic perfectly plastic material with a von Mises yield criterion. Figure 7a illustrates the problem setup, where r_i, r_o and r_p are inner radius, outer radius and the radius of the plastic front, respectively. Hill [32] provides an analytical solution to this problem. Equation (16) gives the relation between the location of the plastic front, r_p, and the applied internal pressure, P, wherein σ_y is the yield stress.

$$P = 2\,\sigma_y \ln\left(\frac{r_p}{r_i}\right) + \frac{2\,\sigma_y}{3}\left(1 - \frac{r_p^3}{r_o^3}\right) \tag{16}$$

The analytical solutions of the normal stresses in spherical coordinates are provided in equations (17) and (18). The shear components in spherical coordinates are zero.

$$\sigma_{rr} = \begin{cases} -2\,\sigma_y\left[\ln\left(\frac{r_p}{r}\right) + \frac{1}{3}\left(1 - \frac{r_p^3}{r_o^3}\right)\right] & \text{if } r \le r_p \\ -\frac{2\,\sigma_y\,r_p^3}{3\,r_o^3}\left(\frac{r_o^3}{r^3} - 1\right) & \text{if } r > r_p \end{cases} \tag{17}$$

Journal publication: https://doi.org/10.1016/j.camwa.2018.10.027



(a) An octant of the spherical shell (b) Smart octree integration on $4 \times 4 \times 4$ base mesh

Figure 7: Set up of internally pressurized sphere

$$\sigma_{\theta\theta} = \sigma_{\phi\phi} = \begin{cases} 2\,\sigma_y \left[\dfrac{1}{2} - \ln\left(\dfrac{r_p}{r}\right) - \dfrac{1}{3}\left(1 - \dfrac{r_p^3}{r_o^3}\right) \right] & \text{if } \; r \leq r_p \\ \dfrac{2\,\sigma_y\, r_p^3}{3\, r_o^3}\left(\dfrac{r_o^3}{2\, r^3} + 1\right) & \text{if } \; r > r_p \end{cases} \tag{18}$$

Due to symmetry, only an octant of the whole spherical shell is considered in the numerical model with the appropriate symmetry boundary conditions. The shell is embedded in a Cartesian grid of size $4 \times 4 \times 4$ elements, where finite cells located completely outside the spherical shell are removed from the computation. For the numerical simulations the spherical shell is selected to have an inner radius of 50 mm and outer radius of 100 mm. The Young's modulus, E, and Poisson's ratio, ν, are set to 10 GPa and 0.3, respectively, and the yield stress, σ_y, is set to 41.79389833783693 MPa.

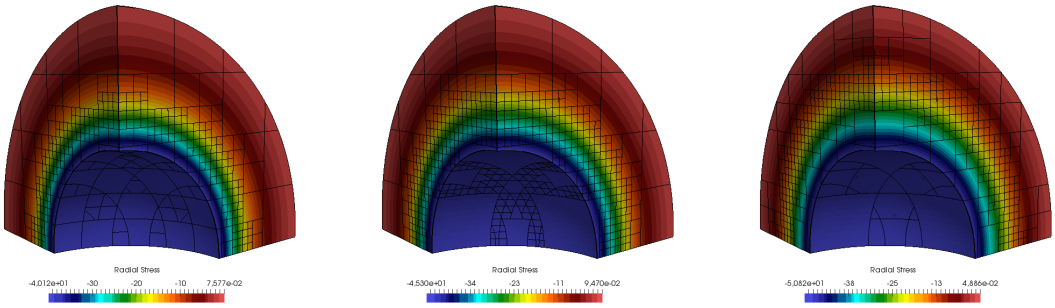
The geometry of the spherical shell is captured at the integration level by utilizing the smart octree algorithm as explained in [33]. Figure 7b illustrates the resulting integration points and cells for base elements.

Figure 8 shows the distribution of the radial stresses on a dynamically adaptive multi-level hp-discretization for three load steps, where the polynomial degree of the basis functions is chosen to be $p = 4$ and maximum refinement level is set to 3. In each step the elements that are close to the plastic front are refined and the elements that are further away from the plastic front are coarsened. Moreover, in each step the primary variables related to degrees of freedom and internal variables at integration points are transferred as explained in Section 2.3.3.

It can be seen from the equations (17) and (18) that the stresses vary only in r-direction and are constant in θ- and ϕ-directions. Therefore, it is sufficient to compare the computed stresses along the radius of the spherical shell to the analytical stresses as depicted in Figure 9 and Figure 10. Both radial and tangential stress results obtained from the numerical simulation match their analytical counterparts for each load step. The different material behavior in elastic and plastic regions introduces a kink in the stress field, which is more blatant for the tangential stress. The introduced multi-level hp-refinement towards that elastic-plastic interface allows the numerical solution to closely capture this behavior.

In order to investigate the convergence properties of this elasto-plastic problem in the framework of

Journal publication: https://doi.org/10.1016/j.camwa.2018.10.027



(a) Radial stresses at P = 40 MPa (b) Radial stresses at P = 45 MPa (c) Radial stresses at P = 50 MPa

Figure 8: Radial stresses on a dynamically adaptive multi-level *hp*-discretization with polynomial degree 4.

the multi-level *hp*-adaptive finite cell method as described in sections 2.1 to 2.3, p-convergence studies are performed by uniformly elevating the order p of the polynomial shape functions, while keeping the size of the base elements fixed. For these studies a $4 \times 4 \times 4$ base mesh is used, which is recursively refined towards the elastic-plastic interface. Moreover, an internal pressure $P = 50$ MPa is applied in one load step for each computation. To this end, the relative error in energy is monitored, which is defined as

$$e = \sqrt{\frac{|U_{ex} - U_{num}|}{U_{ex}}} \times 100, \tag{19}$$

with the exact internal energy, U_{ex}, and the numerical internal energy, U_{num}, computed as

$$U = \frac{1}{2} \int_{\Omega} \boldsymbol{\varepsilon} : \boldsymbol{\sigma} \, \mathrm{d}\Omega. \tag{20}$$

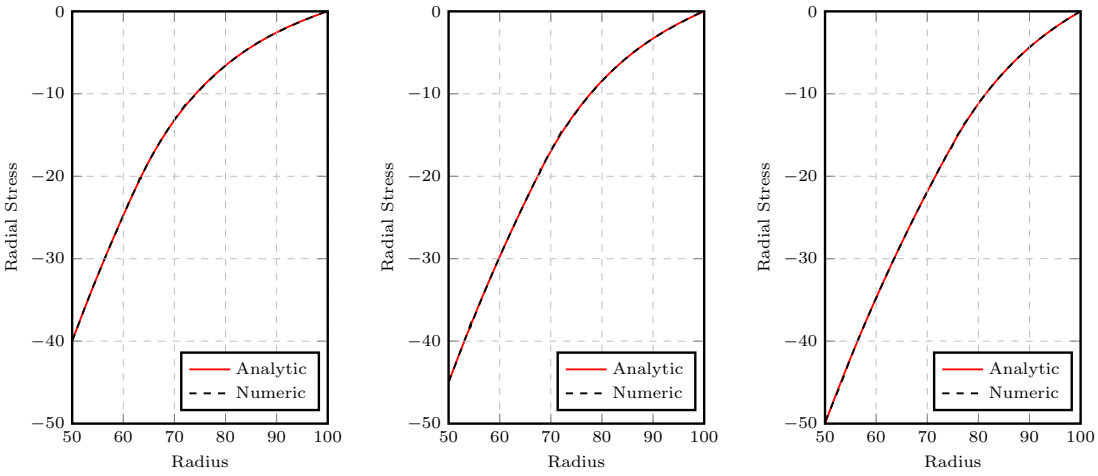
By using the analytical solutions in equation (20) the exact internal energy for an octant of the spherical shell is obtained as follows

$$U_{ex} = \frac{\pi \, P \, \sigma_y \, r_i^3}{4 \, E} \left[(1-\nu) \frac{r_p^3}{r_i^3} - \frac{2}{3}(1-2\,\nu) \left(1 - \frac{r_p^3}{r_o^3} + 3 \, \ln\left(\frac{r_p}{r_i}\right) \right) \right]. \tag{21}$$

Figure 11 shows the relative error in energy with respect to the number of degrees of freedom for studies where the maximum refinement level is increased from zero to three. To rule out any domain integration errors, all volumetric integrals were evaluated exactly following [33]. The curve without any multi-level refinement depicts exponential convergence in the pre-asymptotic range until the error is dominated by the kink in the solution field caused by the elastic-plastic interface. This kink is neither resolved by the boundaries of the finite cells nor is the error controlled by a refinement in the vicinity of this irregularity in the solution. Therefore, in the asymptotic range, the error decreases only algebraically when the polynomial degree is further increased. The kink in the solution field is better approximated by the basis functions as the elements are refined towards the elastic-plastic interface with hierarchical multi-level refinement. This can be seen in curves with one and two refinement levels towards the elastic-plastic interface, where the point at which the convergence levels off to its asymptotic value occurs much later. The curve with three refinement levels converges exponentially. Here, the error at the interface does not dominate the overall error.

This demonstrates that the combination of the techniques presented in sections 2.1 to 2.3 is able to maintain the expected convergence behavior for higher-order methods even if neither the physical boundaries of

Journal publication: https://doi.org/10.1016/j.camwa.2018.10.027



(a) P = 40 MPa (b) P = 45 MPa (c) P = 50 MPa

Figure 9: Radial stresses along radius of the spherical shell

an artifact nor an evolving elasto-plastic front are explicitly captured by the boundaries of the discretization.

3.2. Thermo-elasto-plastic bar example

In this section, the thermo-mechanical model introduced in Section 2.4 is investigated in the framework of multi-level *hp*-adaptivity. To this end, an idealized version of the metal casting process is simulated on a semi-infinite bar. Initially, the bar, which is given in Figure 12 , is liquid with a uniform temperature, T_i. Then, the temperature at the boundary $x = 0$ is instantaneously changed to T_w and kept constant. T_w is lower than the melting temperature, T_m. Therefore, the bar starts to solidify from this boundary onwards.

The analytical solution to the thermal part of this problem is described in [34] and is known in the literature as *Neumanns's method*, see e.g. [35, 36]. The position of the liquid-solid interface is given as

$$X(t) = 2\,\lambda\,\sqrt{\alpha\,t}, \tag{22}$$

where α is the diffusivity of the material and t is the time. The constant λ is computed by solving the following nonlinear equation

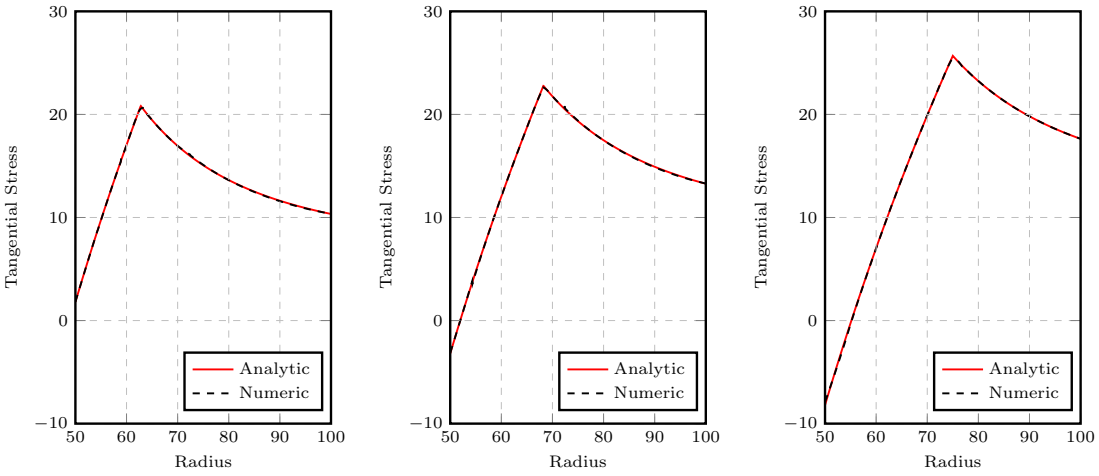
$$\frac{e^{-\lambda^2}}{\operatorname{erf}(\lambda)} + \frac{e^{-\lambda^2}\,(T_m - T_i)}{\operatorname{erfc}(\lambda)\,(T_m - T_w)} = \frac{\lambda\,L\sqrt{\pi}}{c\,(T_m - T_w)}, \tag{23}$$

where L is the latent heat and c is the heat capacity. The analytical temperature distribution is given in equation (24) for the semi-infinite bar.

$$T(x,t) = \begin{cases} T_w + (T_m - T_w)\dfrac{\operatorname{erf}(x/2\sqrt{\alpha t})}{\operatorname{erf}(\lambda)} & \text{if} \quad x \le X(t) \\ \\ T_i + (T_m - T_i)\dfrac{\operatorname{erfc}(x/2\sqrt{\alpha t})}{\operatorname{erfc}(\lambda)} & \text{if} \quad x > X(t) \end{cases} \tag{24}$$

By using the temperature distribution over the body obtained from the *Neumanns's method*, Weiner and Boley [37] developed an analytical solution for the thermal stresses for this problem. They assumed that

Journal publication: https://doi.org/10.1016/j.camwa.2018.10.027



(a) P = 40 MPa (b) P = 45 MPa (c) P = 50 MPa

Figure 10: Tangential stresses along the radius of the spherical shell

the body is composed of an elastic perfectly plastic material with a yield stress, σ_y, which linearly decreases as the temperature reaches the melting point, where it becomes zero:

$$\sigma_y(T) = \begin{cases} \sigma_{y_0} \dfrac{T_m - T}{T_m - T_w} & \text{if} \quad T_w \le T \le T_m, \\ \\ 0 & \text{if} \quad T > T_m. \end{cases} \tag{25}$$

Moreover, the following dimensionless quantities are introduced,

$$m = \frac{(1-\nu)\,\sigma_{y_0}}{\gamma\,E\,(T_m - T_w)}, \quad D = \frac{1}{\operatorname{erf}(\lambda)}, \quad \hat{x} = \frac{x}{X(t)}, \quad \hat{\sigma} = \frac{\sigma\,(1-\nu)}{\gamma\,E\,(T_m - T_w)}, \tag{26}$$

where E, ν and γ are Young's modulus, Poisson's ratio and thermal expansion coefficient, respectively. The analytical solution of the dimensionless normal stress components in y- and z-directions are then given as

$$\hat{\sigma}_{yy}(\hat{x}) = \hat{\sigma}_{zz}(\hat{x}) = \begin{cases} m\,[D\operatorname{erf}(\lambda\,\hat{x}) - 1] & \text{if} \quad 0 \le \hat{x} < \hat{x}_2 \\ \\ m\,[1 - D\operatorname{erf}(\lambda\,\hat{x}_1)] + D\,[\operatorname{erf}(\lambda\,\hat{x}_1) - \operatorname{erf}(\lambda\,\hat{x})] & \\ \quad -\dfrac{2}{\sqrt{\pi}}\,D\,(1-m)\,\lambda\,\hat{x}_1\,e^{-\lambda^2\hat{x}_1^2}\log(\dfrac{\hat{x}_1}{\hat{x}}) & \text{if} \quad \hat{x}_2 \le \hat{x} \le \hat{x}_1 \\ \\ m\,[1 - D\operatorname{erf}(\lambda\,\hat{x})] & \text{if} \quad \hat{x}_1 < \hat{x} \le 1 \\ \\ 0 & \text{if} \quad \hat{x} > 1 \end{cases} \tag{27}$$

where the coordinates $\hat{x}_1$ and $\hat{x}_2$ denote the elastic-plastic interfaces. According to this solution, the solid material behaves elastic within the range $[\hat{x}_2, \hat{x}_1]$ and plastic in ranges $[0, \hat{x}_2]$ and $[\hat{x}_1, 1]$. The positions of

Journal publication: https://doi.org/10.1016/j.camwa.2018.10.027

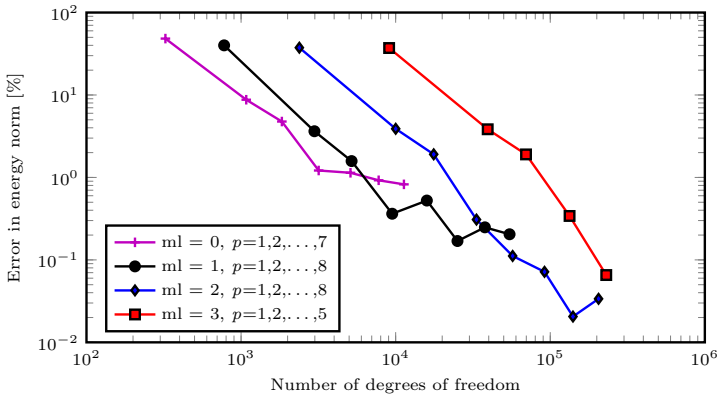
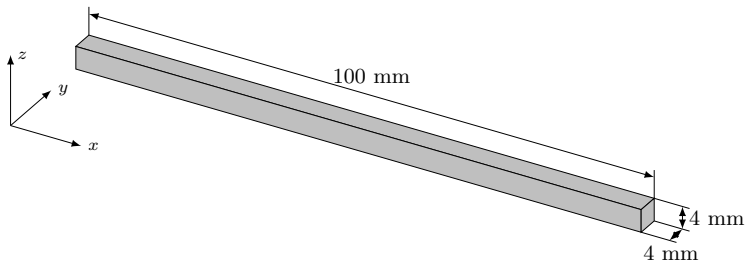


Figure 11: Convergence in energy norm starting from a $4 \times 4 \times 4$ base mesh with multi-level *hp* refinement (ml).



T_m	1490 °C
T_w	1362 °C
T_i	1550 °C
ρ	7200 kg/m^3
$c_l = c_s$	680 J/(m^3 °C)
$k_l = k_s$	34 W/(m °C)
L	272000 J/kg
E	40 GPa
ν	0.35
σ_{y_0}	40 MPa
γ	8.46354×10^{-5}/°C

Figure 12: Set up of solidifying bar

these interfaces are obtained by solving the following nonlinear equations

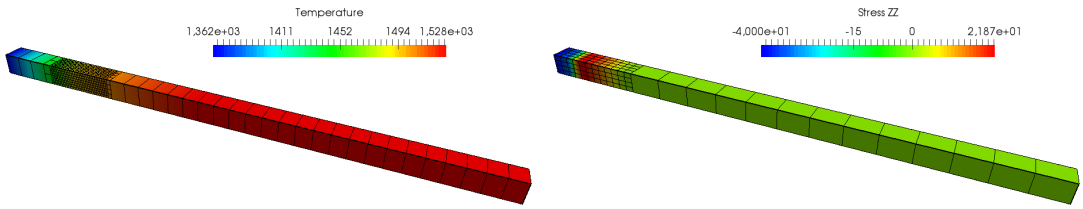
$$\begin{aligned} 2\,(1-m)\,\lambda^2\,\hat{x}_1\,e^{-\lambda^2\hat{x}_1^2}\,(\hat{x}_1-\hat{x}_2) &= (1+m)\,e^{-\lambda^2\hat{x}_2^2} - (1-m)\exp(-\lambda^2\,\hat{x}_1^2) - m(e^{-\lambda^2}+1), \\ \frac{2(1-m)\,\lambda\,\hat{x}_1\,e^{-\lambda^2\hat{x}_1^2}}{\sqrt{\pi}}\log(\frac{\hat{x}_1}{\hat{x}_2}) &= (1-m)\,\mathrm{erf}(\lambda\,\hat{x}_1) - (1+m)\,\mathrm{erf}(\lambda\,\hat{x}_2) + 2\,m\,\mathrm{erf}(\lambda) \end{aligned} \tag{28}$$

The values that are selected for the material properties of this problem are provided in Figure 12 . With these values, the constant λ is computed as 0.330825295611989, while the coordinates $\hat{x}_1$ and $\hat{x}_2$ are found to be 0.45487188 and 0.21570439, respectively.

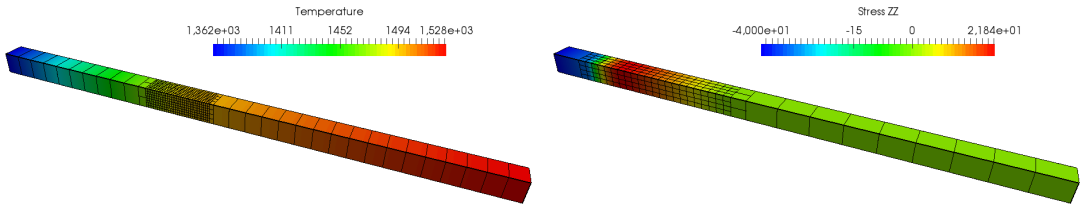
The transient thermal problem is solved on a base mesh with 32 hexahedral elements of order $p = 3$ that are distributed in x-direction. Homogeneous Neumann boundary conditions are applied along y- and z-directions and Dirichlet boundary conditions are applied in the yz-plane at $x = 0$ and $x = 100$. In order to better represent the kink in the temperature field, the elements are dynamically refined three times towards the solid-liquid interface. The time step for the backward Euler scheme is chosen to be $\delta t = 0.1$ seconds.

The simulation for the quasi-static mechanical problem is carried out on a base mesh with 16 hexahedral elements of order $p = 4$ which are dynamically refined twice towards both the solid-liquid interface and the elastic-plastic interfaces. During the simulation the liquid part of the bar is treated as a fictitious domain as explained in Section 2.2 by multiplying E with $\alpha = 10^{-8}$. Extended plane strain conditions are applied along y- and z-directions such that ε_{yy} and ε_{zz} are constant. This is achieved by constraining all degree of

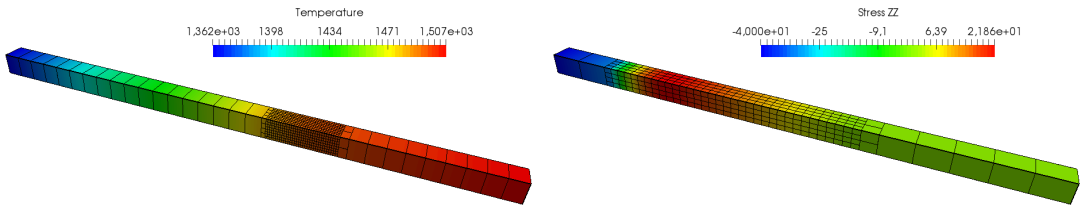
Journal publication: https://doi.org/10.1016/j.camwa.2018.10.027



(a) t = 1 second



(b) t = 5.1 seconds



(c) t = 13.2 seconds

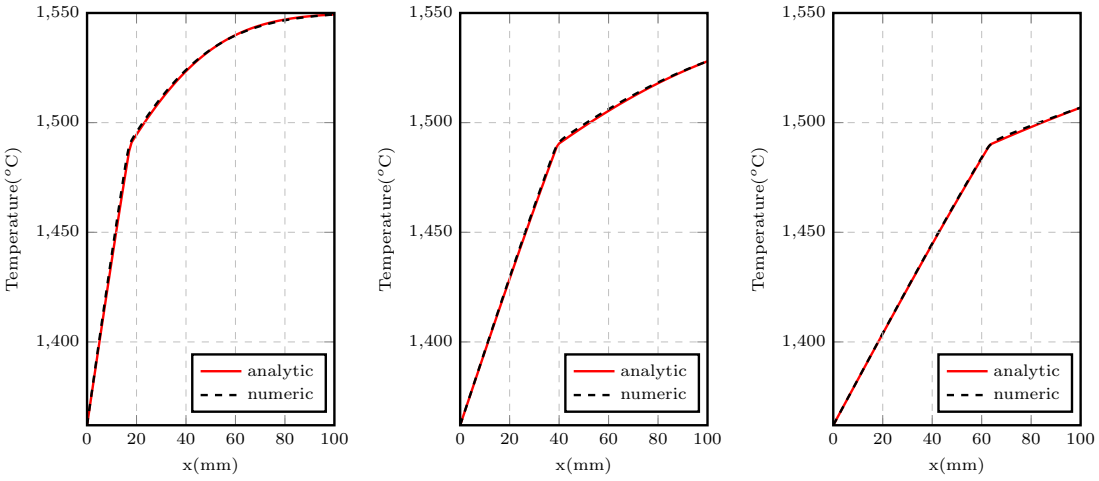
Figure 13: Temperature and stress distributions at various time states on dynamically refined thermal and mechanical meshes, respectively

freedoms corresponding to displacements in y- and z-directions to be equal via Lagrange multipliers. As for the x-direction, the bar is fixed at $x = 0$ and left free at $x = 100$.

Figure 13 presents the results of the thermal and the mechanical problems for the time states $t = \{1, 5.1, 13.2\}$ seconds. For the thermal problem the evolution of the temperature field is provided over the dynamically adapted thermal mesh. As for the mechanical problem the stress component in z-direction (σ_{zz}) is given over the dynamically adapted mechanical mesh. The sequence of figures demonstrates how elements are refined and coarsened during the simulation for thermal and mechanical problems independently.

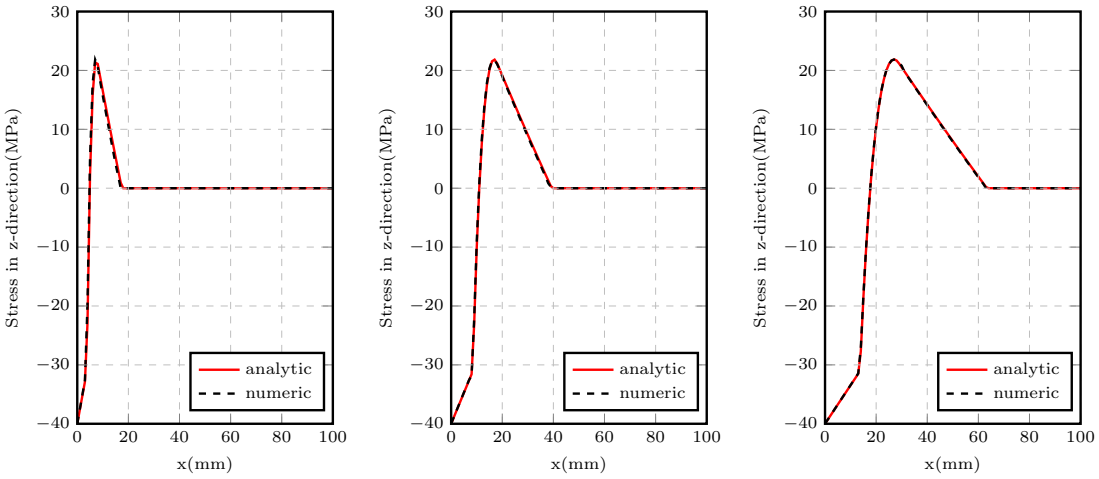
Figure 14 compares the temperature distribution along the bar to the analytical solution obtained by the *Neumanns's method*. The kink in the temperature field at melting temperature, which is caused by the latent heat release during the phase change from liquid to solid, is well captured in all time states as well as the general temperature profile along the bar. The stress component in z-direction (σ_{zz}) along the bar is depicted in Figure 15 along with the analytical solution provided by Weiner and Boley. The kinks at the elastio-plastic interfaces and the solid-liquid interface are well represented due to refinements. Moreover, it can be seen that the numerical results closely match their analytical counterparts along the bar at all time

Journal publication: https://doi.org/10.1016/j.camwa.2018.10.027



(a) t = 1 second (b) t = 5.1 seconds (c) t = 13.2 seconds

Figure 14: Comparison of the numerical and the analytical temperature distribution along x-axis.



(a) t = 1 second (b) t = 5.1 seconds (c) t = 13.2 seconds

Figure 15: Comparison of the numerical and the analytical stress (σ_{zz}) distribution along x-axis.

steps of the computation.

3.3. Applications to metal deposition

In this section the performance of the numerical method introduced in Section 2 in simulating metal deposition processes is investigated against the benchmark problem of a single bead laid down on the top surface of a plate as published in [38]. This benchmark problem is produced by the European Network on Neutron Techniques Standardization for Structural Integrity (NeT), whose mission is to develop experimental

Journal publication: https://doi.org/10.1016/j.camwa.2018.10.027

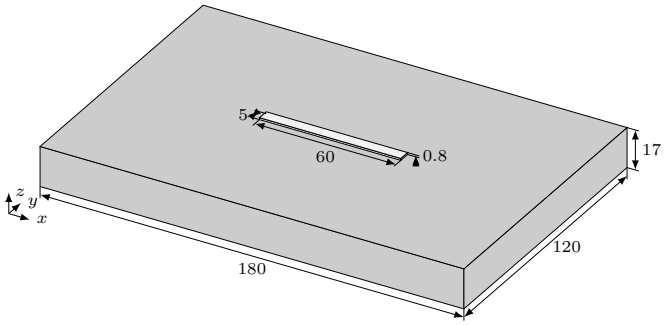


Figure 16: Setup of welding

and numerical techniques and standards for the reliable characterisation of residual stresses in structural welds. Partners from industry and academia participated in this benchmark by constructing an experiment and measuring the residual stresses on the plate [39, 40, 41, 42, 43] or predicting the residual stresses through numerical methods [44, 39, 45, 46, 40, 47]. Smith et. al. compares the residual stress measurements and predictions of all these groups in [48].

Figure 16 illustrates the setup of the benchmark problem where the plate has dimensions 180 mm × 120 mm × 17 mm in length, width and thickness, respectively. Additionally, the bead that is laid down on the top surface of the plate has a width of 5 mm, thickness of 0.8 mm and a length of 60 mm.

According to the benchmark protocol, the plate and the bead material is AISI Type 316L stainless steel. NeT recommends using slightly different temperature dependent material properties for the parent and weld metal. However, in this work the weld metal is not distinguished from the parent metal. Table 1 provides the temperature dependent specific heat, conductivity, thermal expansion coefficient and Young's modulus. Over the melting temperature of 1400 °C these values are assumed to be constant. In addition to that the density and Poisson's ratio are assumed to be constant values of 7966 kg m^{-3} and 0.294, respectively and latent heat is taken as 260 kJ/kg. Finally, the monotonic tensile test results, which show the true stress corresponding to the percentage of plastic strain, is given in Table 2 by NeT. The isotropic hardening behaviour of the material is based on this table.

The heat input from the welding torch, which travels with the speed of 2.27 mm/s, is defined by NeT as 633 J/mm. The moving heat source is based on the Goldak's [49] ellipsoidal model, which is widely used in welding simulations. In this model the body load is given as

$$q(x,y,z) = \frac{6\sqrt{3}\,Q}{\pi\sqrt{\pi}\,r_x\,r_y\,r_z}\, e^{-3\left(\frac{x-x_c}{r_x}\right)^2} e^{-3\left(\frac{y-y_c}{r_y}\right)^2} e^{-3\left(\frac{z-z_c}{r_z}\right)^2}, \tag{29}$$

where x_c, y_c and z_c define the position of the welding torch and r_x, r_y and r_z are the semi-axes of the ellipsoid. Shan et. al.[44] uses the weld bead profile measurements to decide the semi-axes of the ellipsoid, which is followed in this work. According to these measurements r_x, r_y and r_z are assumed to be 1.9 mm, 3.2 mm and 2.8 mm, respectively. Moreover, in equation (29) Q is the power of the heat source, which is computed from the heat input and the welding speed as 1437 Watt.

The heat loss from the system is modeled with radiation and convection boundary conditions on the surface of the weld bead and the plate. During the process, the surface is updated as new weld bead is deposited. The convection coefficient and the emissivity are taken as 10 W/m^2K and 0.75, respectively and they are assumed to be independent of the temperature. Additionally, the ambient temperature is also assumed to be constant at 20 °C.

Due to symmetry, only half of the plate and the weld bead is considered as shown in Figure 17 for the numerical model with the appropriate symmetry boundary conditions. The plate and the weld bead are

Journal publication: https://doi.org/10.1016/j.camwa.2018.10.027

Table 1: Material properties of AISI Type 316L stainless steel [44].

Temperature [°C]	Specific heat [kJ/kg/°C]	Conductivity [W/m/°C]	Thermal expansion [$\times 10^6$ mm/mm/°C]	Young's modulus [GPa]
20	0.492	14.12	14.56	195.6
100	0.502	15.26	15.39	191.2
200	0.514	16.69	16.21	185.4
300	0.526	18.11	16.86	179.6
400	0.538	19.54	17.37	172.6
500	0.550	20.96	17.78	164.5
600	0.562	22.38	18.12	155.0
700	0.575	23.81	18.43	144.1
800	0.587	25.23	18.72	131.4
900	0.599	26.66	18.99	116.8
1000	0.611	28.08	19.27	100.0
1100	0.623	29.50	19.53	80.0
1200	0.635	30.93	19.79	57.0
1300	0.647	32.35	20.02	30.0
1400	0.659	33.78	20.21	2.0

Table 2: True stress values [MPa] of AISI Type 316L stainless steel at various true plastic strain [40].

Temperature [°C]	0%	0.2%	1%	2%	5%	10%	20%	30%	40%
23	210	238	292	325	393	494	648	775	880
275	150	173.7	217	249	325	424	544	575	
550	112	142.3	178	211	286	380	480	500	
750	95	114.7	147	167	195	216	231	236	
800	88	112	120	129	150	169	183		
900	69	70	71	73	76	81			
1100	22.4								
1400	2.7								

embedded in a Cartesian grid of size $30 \times 12 \times 7$ elements with polynomial order 3 for both the thermal and mechanical problems. The dimensions of the computational domain are 180 mm, 60 mm and 18.6 mm in length, width and thickness, respectively. It should be noted that the thickness of the computational domain is greater than the total thickness of the plate and the weld bead, which is 17.8 mm. The first six rows of elements in thickness are used to discretize the plate, while the weld bead is embedded in the last row of elements.

Figure 18 shows a snapshot of the dynamic meshes of the thermal and the mechanical problems with material states, where red, blue and white (hollow) denote weld bead, plate and air, respectively. It should be noted that only the mechanical mesh is refined once along the welding line before deposition starts, which is not coarsened afterwards. Whereas, both meshes are dynamically refined twice towards the weld torch when material is deposited. The weld torch is set to travel 1.5 mm in each time step. Therefore, the newly deposited domain near the weld torch can be captured by the sub-element boundaries and the multi-level grid that keeps track of the physical domain. As the weld torch moves further away, these sub-elements are coarsened while the multi-level grid is used to remember the deposited domain as explained in Section 2.4. Figure 19 depicts the dynamic meshes with temperature and longitudinal stress (σ_{xx}) distributions during the deposition period.

Journal publication: https://doi.org/10.1016/j.camwa.2018.10.027

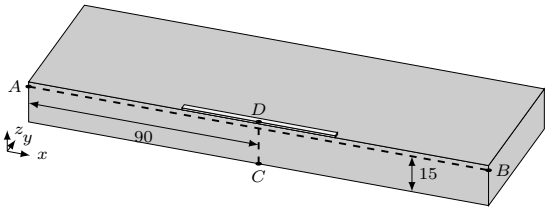
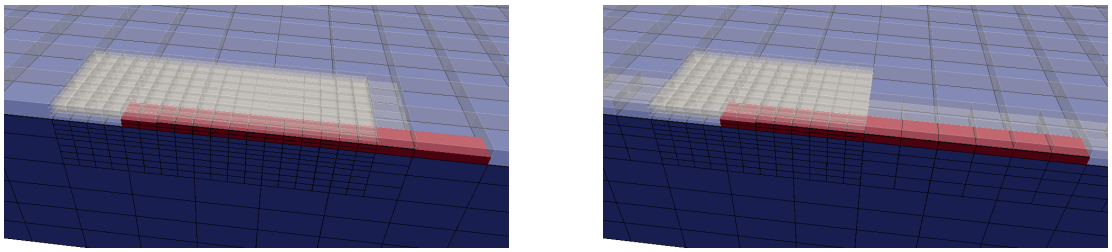


Figure 17: Setup of welding



(a) Mesh of the thermal problem

(b) Mesh of the mechanical problem

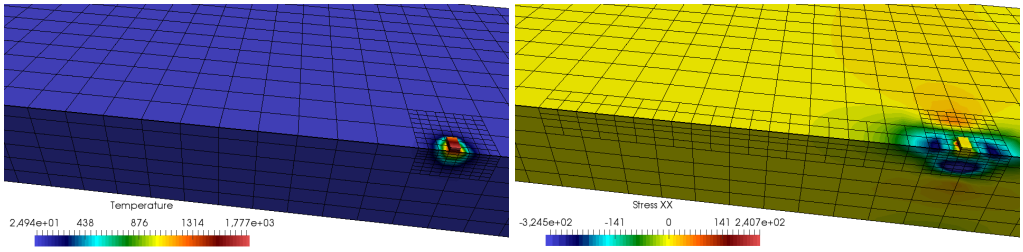
Figure 18: Meshes used for numerical simulation with material states, where red, blue and white (hollow) denote weld bead, plate and air, respectively.

Residual stress distributions after the parts are cooled down to the room temperature are given in Figure 20 over the physical domain. Gilles et. al. [45] measures the longitudinal and transverse residual stresses with a neutron diffraction technique over the lines AB and CD. These lines can be seen in Figure 17 . Figure 21 and Figure 22 compare the numerically predicted residual stresses to the measurements from experiment by Gilles et. al. along the lines AB and CD, respectively. It can be seen from these figures that the measured and the numerically computed stress profiles are in good agreement. However, utilizing the complete isotropic hardening values in Table 2 results in higher residual stress especially for the longitudinal stress component, which is also reported by Gilles et. al. [45]. Their explanation for effect is that the isotropic hardening law which does not consider stress relaxation due to viscoplastic effects or hardening recovery is not good enough to simulate cyclic loads which occur during the welding process. Therefore, the isotropic hardening model is modified in this work such that over one percent of plastic strain the material is assumed to behave elastic perfectly plastic. The results with this modified model match the experimental values as illustrated in Figure 21.

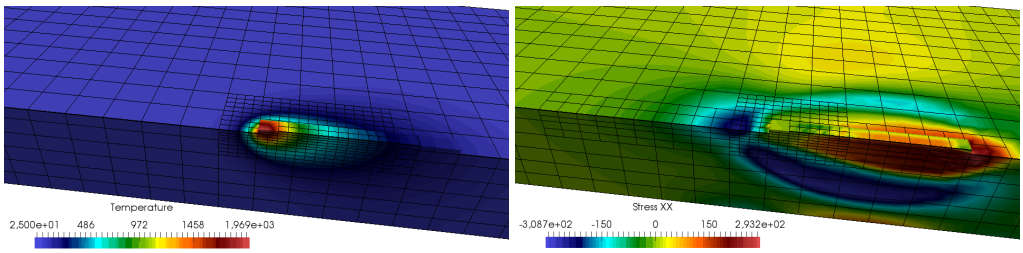
4. Summary and conclusions

The article at hand evaluates if high-order finite elements may be used in metal deposition modeling. To this end, the finite cell method, an embedded domain method of high order, was used. It was demonstrated that this method, combined with the multi-level *hp*-method, leads to high-order convergence rates even if neither, the physical boundaries nor the plastic front is resolved by the boundaries of the computational mesh. Therefore, high-order embedded domain modeling is a valid computational option. Very accurate results were also achieved in the thermo-elasto-plastic setting. Therein, the thermal and the elasto-plastic field variables were computed on their own high-order discetizations and coupled by means of a classical staggered scheme. In all of the computations the state variables were decoupled from the finite cells in an FCM sense by managing them on another computational grid. This allows for their separate coarsening and refinement and a simulation where material can be submerged into the computation on a sub-element

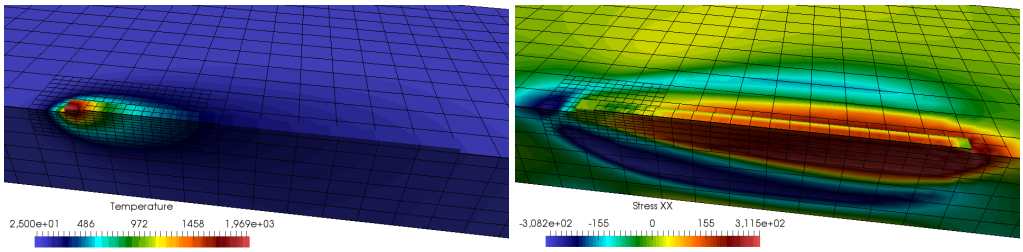
Journal publication: https://doi.org/10.1016/j.camwa.2018.10.027



(a) t = 1 second

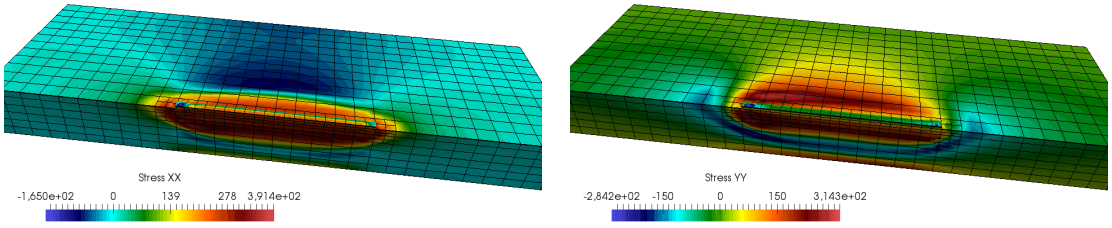


(b) t = 5.1 seconds



(c) t = 13.2 seconds

Figure 19: Evolution of the temperature and longitudinal stress longitudinal stress (σ_{xx}) distributions during the welding process on dynamic meshes at various time states.

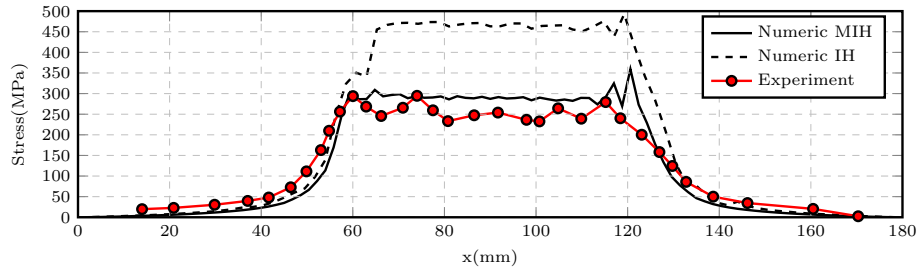


(a) Longitudinal stresses (σ_{xx})

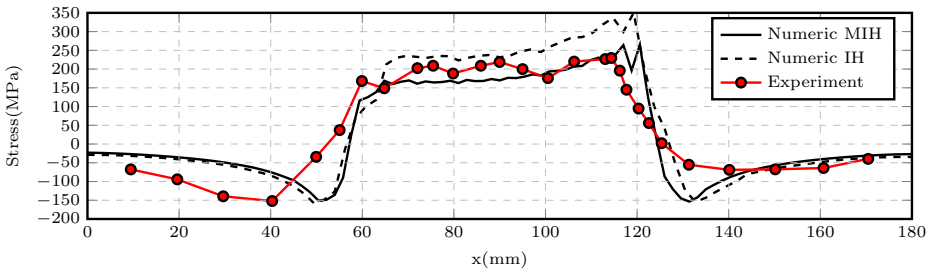
(b) Transverse stresses (σ_{yy})

Figure 20: Residual stress distributions after cooling down to room temperature.

Journal publication: https://doi.org/10.1016/j.camwa.2018.10.027

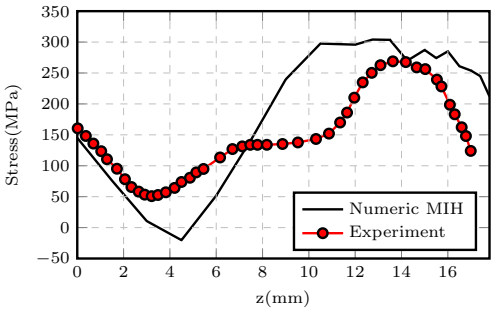


(a) Longitudinal stresses (σ_{xx})

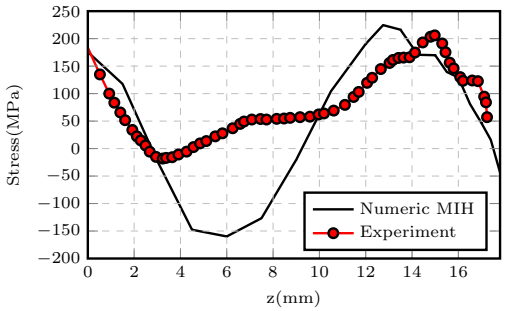


(b) Transverse stresses (σ_{yy})

Figure 21: Numerical and experimental ([45]) results over line AB (IH: Isotropic hardening, MIH: Modified isotropic hardening)



(a) Longitudinal stresses (σ_{xx})



(b) Transverse stresses (σ_{yy})

Figure 22: Numerical and experimental ([45]) results over line CD (MIH: Modified isotropic hardening)

Journal publication: https://doi.org/10.1016/j.camwa.2018.10.027

level i.e. the addition of material is not equal to the addition of finite cells. The methods were thoroughly verified in semi-analytical benchmarks and then applied to an experimental benchmark of metal deposition. A good agreement with the experimentally measured residual strains/stresses was found.

While the overall computational methodology leads to very accurate results open research questions remain. For example, the presented de-refinements in the mechanical part of the simulation are clearly a valid option to accurately compute the stress states in the vicinity of a plastic front. However, the stress state in a manufactured part does not only consist of one plastic front and rather smooth stress states to the side of it. Instead, very complex, local stress states may be present. A straightforward coarsening procedure of the grid in these regions will cause smoothing effects for variables which are not diffusive by nature. It is beyond the scope of this paper to study the influence of strong, local coarsening in complex stress states and their effect in more involved computations. The investigation of this challenge remains open. Since the local plastic strains are not connected to the finite cells but stored in an extra grid which acts like a computational database, one option would be not to coarsen them at all. The computational grid, carrying the degrees of freedom could be coarsened independently. Such a procedure would allow for a constant number of degrees of freedom throughout the computation while detailed information on the local plastic strains would still be available if needed even if the finite cells were coarsened.

Next, it is worthwile to investigate the wall-clock time advantages of the computational approach w.r.t. conventional modeling techniques in detail. This is not easy to some extent because a fair comparison requires an efficient implementation of both techniques which is not available at present. Clearly, the modeling flexibility gained by the presented embedded domain method of high order using transient refinements does introduce a certain implementational as well as a computational overhead as compared to a classical fixed-grid approaches. The break-even point of refining and de-refining regarding wall-clock time further depends on the scale of the computations. However, to refine everywhere to the finest level required locally is also not an option. Thus, refinements and de-refinements must be used for efficient computations. How this is possible was investigated in the article at hand, but its comparison to other refinement techniques remains an open question. The current computations hint that the type of microscopic computations in the metal depositing process presented in the last example pay off in a multi-layer process whereby the refinement and de-refinement is not carried out at each time increment.

In summary, the article presents the first verified and validated steps of a novel computational methodology for the analysis of metal deposition modeling in an embedded domain sense using locally refined, transient discretizations of *hp*-type. As such, it demonstrates that this computational methodology is a valid option for the analysis of metal deposition.

Acknowledgements

The authors gratefully acknowledge the financial support of the German Research Foundation (DFG) under Grant RA 624/27-2.

Journal publication: https://doi.org/10.1016/j.camwa.2018.10.027

References

[1] J. A. Goldak, M. Akhlaghi, Computational Welding Mechanics, Springer, New York, 2005.

[2] L. E. Lindgren, L.E. Lindgren-Computational Welding Mechanics Thermomechanical and Microstructural Simulations, Woodhead Publishing Limited and CRC Press LLC, 2007.

[3] M. Tanaka, An introduction to physical phenomena in arc welding processes, Welding International 18 (11) (2004) 845–851. doi:10.1533/wint.2004.3342.

[4] O. Zinovieva, A. Zinoviev, V. Ploshikhin, Three-dimensional modeling of the microstructure evolution during metal additive manufacturing, Computational Materials Science 141 (2018) 207–220. doi:10.1016/j.commatsci.2017.09.018.

[5] A. Basak, S. Das, Epitaxy and Microstructure Evolution in Metal Additive Manufacturing, Annual Review of Materials Research 46 (1) (2016) 125–149. doi:10.1146/annurev-matsci-070115-031728.

[6] C. Körner, Additive manufacturing of metallic components by selective electron beam melting — a review, International Materials Reviews 61 (5) (2016) 361–377. doi:10.1080/09506608.2016.1176289.

[7] L. E. Lindgren, Numerical modelling of welding, Computer Methods in Applied Mechanics and Engineering 195 (48–49) (2006) 6710–6736, 00127. doi:10.1016/j.cma.2005.08.018.

[8] L. E. Lindgren, H. A. Häggblad, J. M. J. McDill, A. S. Oddy, Automatic remeshing for three-dimensional finite element simulation of welding, Computer Methods in Applied Mechanics and Engineering 147 (3–4) (1997) 401–409, 00052. doi:10.1016/S0045-7825(97)00025-X.

[9] B. Schoinochoritis, D. Chantzis, K. Salonitis, Simulation of metallic powder bed additive manufacturing processes with the finite element method: A critical review, Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture 231 (1) (2017) 96–117. doi:10.1177/0954405414567522.

[10] V. Nübel, A. Düster, E. Rank, An rp-adaptive finite element method for the deformation theory of plasticity, Computational Mechanics 39 (5) (2007) 557–574. doi:10.1007/s00466-006-0111-4.

[11] M. Olesky, W. Cecot, Application of the fully automatic hp-adaptive FEM to elastic-plastic problems, Computer Methods in Materials Science 15 (1) (2015) 204–2012.

[12] T. Elguedj, T. Hughes, Isogeometric analysis of nearly incompressible large strain plasticity, Computer Methods in Applied Mechanics and Engineering 268 (2014) 388–416. doi:10.1016/j.cma.2013.09.024.

[13] A. Abedian, J. Parvizian, A. Düster, E. Rank, Finite cell method compared to h-version finite element method for elasto-plastic problems, Applied Mathematics and Mechanics 35 (10) (2014) 1239–1248, 00002. doi:10.1007/s10483-014-1861-9.

[14] A. Taghipour, Parvizian, J., S. Heinze, A. Düster, The finite cell method for nearly incompressible finite strain plasticity problems with complex geometries, Computers & Mathematics with Applications (2018) accepted.

[15] A. Düster, J. Parvizian, Z. Yang, E. Rank, The finite cell method for three-dimensional problems of solid mechanics, Computer Methods in Applied Mechanics and Engineering 197 (45–48) (2008) 3768–3782. doi:10.1016/j.cma.2008.02.036.

[16] N. Zander, T. Bog, M. Elhaddad, F. Frischmann, S. Kollmannsberger, E. Rank, The multi-level hp-method for three-dimensional problems: Dynamically changing high-order mesh refinement with arbitrary hanging nodes, Computer Methods in Applied Mechanics and Engineering 310 (2016) 252–277. doi:10.1016/j.cma.2016.07.007.

[17] N. Zander, T. Bog, S. Kollmannsberger, D. Schillinger, E. Rank, Multi-level hp-adaptivity: High-order mesh adaptivity without the difficulties of constraining hanging nodes, Computational Mechanics 55 (3) (2015) 499–517. doi:10.1007/s00466-014-1118-x.

[18] J. Parvizian, A. Düster, E. Rank, Finite cell method, Computational Mechanics 41 (1) (2007) 121–133. doi:10.1007/s00466-007-0173-y.

[19] A. Düster, E. Rank, B. A. Szabó, The p-version of the finite element method and finite cell methods, in: E. Stein, R. Borst, T. J. R. Hughes (Eds.), Encyclopedia of Computational Mechanics, Vol. 2, John Wiley & Sons, Chichester, West Sussex, 2017, pp. 1–35.

[20] A. Abedian, J. Parvizian, A. Düster, E. Rank, The finite cell method for the J2 flow theory of plasticity, Finite Elements in Analysis and Design 69 (2013) 37–47, 00000. doi:10.1016/j.finel.2013.01.006.

[21] E. de Souza Neto, P. Perić, P. Owen, Computational Methods for Plasticity: Theory and Applications, John Wiley & Sons, 2009.

[22] D. Perić, C. Hochard, M. Dutko, D. R. J. Owen, Transfer operators for evolving meshes in small strain elasto-placticity, Computer Methods in Applied Mechanics and Engineering 137 (3) (1996) 331–344. doi:10.1016/S0045-7825(96)01070-5.

[23] M. Ortiz, J. J. Quigley, Adaptive mesh refinement in strain localization problems, Computer Methods in Applied Mechanics and Engineering 90 (1) (1991) 781–804. doi:10.1016/0045-7825(91)90184-8.

[24] N.-S. Lee, K.-J. Bathe, Error indicators and adaptive remeshing in large deformation finite element analysis, Finite Elements in Analysis and Design 16 (2) (1994) 99–139. doi:10.1016/0168-874X(94)90044-2.

[25] O. C. Zienkiewicz, J. Z. Zhu, The superconvergent patch recovery and a posteriori error estimates. Part 2: Error estimates and adaptivity, International Journal for Numerical Methods in Engineering 33 (7) (1992) 1365–1382. doi:10.1002/nme.1620330703.

[26] O. Zienkiewicz, J. Zhu, The superconvergent patch recovery (SPR) and adaptive finite element refinement, Computer Methods in Applied Mechanics and Engineering 101 (1-3) (1992) 207–224. doi:10.1016/0045-7825(92)90023-D.

[27] B. Boroomand, O. C. Zienkiewicz, Recovery procedures in error estimation and adaptivity. Part II: Adaptivity in nonlinear problems of elasto-plasticity behaviour, Computer Methods in Applied Mechanics and Engineering 176 (1) (1999) 127–146. doi:10.1016/S0045-7825(98)00333-8.

[28] A. R. Khoei, S. A. Gharehbaghi, Three-dimensional data transfer operators in large plasticity deformations using modified-SPR technique, Applied Mathematical Modelling 33 (7) (2009) 3269–3285. doi:10.1016/j.apm.2008.10.033.

Journal publication: https://doi.org/10.1016/j.camwa.2018.10.027

[29] A. Düster, H. Bröker, 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 (7) (2001) 673–703.
[30] D. Celentano, E. Oñate, S. Oller, A temperature-based formulation for finite element analysis of generalized phase-change problems, International Journal for Numerical Methods in Engineering 37 (20) (1994) 3441–3465.
[31] S. Kollmannsberger, A. I. Özcan, E. Rank, Computational modeling of the SLM process: A hierarchical treatment of the transient heat equation with phase changes in preparation.
[32] R. Hill, The Mathematical Theory of Plasticity, Oxford Classic Texts in the Physical Sciences, Oxford University Press, Oxford, New York, 1998.
[33] L. Kudela, N. Zander, S. Kollmannsberger, E. Rank, Smart octrees: Accurately integrating discontinuous functions in 3D, Computer Methods in Applied Mechanics and Engineering 306 (Supplement C) (2016) 406–426, 00021. doi:10.1016/j.cma.2016.04.006.
[34] H. Weber, B. Riemann, Die Partiellen Differential-Gleichungen Der Mathematischen Physik: Nach Riemann's Vorlesungen in 5. Aufl. Bearb, no. v. 1, F. Vieweg und Sohn, 1912.
[35] H. Hu, S. A. Argyropoulos, Mathematical modelling of solidification and melting: A review, Modelling and Simulation in Materials Science and Engineering 4 (4) (1996) 371.
[36] D. W. Hahn, Heat Conduction, 3rd Edition, Wiley, Hoboken, N.J, 2012.
[37] J. H. Weiner, B. A. Boley, Elasto-plastic thermal stresses in a solidifying body, Journal of the Mechanics and Physics of Solids 11 (3) (1963) 145–154, 00124.
[38] C. E. Truman, M. C. Smith, The NeT residual stress measurement and modelling round robin on a single weld bead-on-plate specimen, International Journal of Pressure Vessels and Piping 86 (1) (2009) 1–2. doi:10.1016/j.ijpvp.2008.11.018.
[39] C. Ohms, R. C. Wimpory, D. E. Katsareas, A. G. Youtsos, NET TG1: Residual stress assessment by neutron diffraction and finite element modeling on a single bead weld on a steel plate, International Journal of Pressure Vessels and Piping 86 (1) (2009) 63–72. doi:10.1016/j.ijpvp.2008.11.009.
[40] X. Ficquet, D. J. Smith, C. E. Truman, E. J. Kingston, R. J. Dennis, Measurement and prediction of residual stress in a bead-on-plate weld benchmark specimen, International Journal of Pressure Vessels and Piping 86 (1) (2009) 20–30. doi:10.1016/j.ijpvp.2008.11.008.
[41] M. Hofmann, R. C. Wimpory, NET TG1: Residual stress analysis on a single bead weld on a steel plate using neutron diffraction at the new engineering instrument 'STRESS-SPEC', International Journal of Pressure Vessels and Piping 86 (1) (2009) 122–125. doi:10.1016/j.ijpvp.2008.11.007.
[42] S. Pratihar, M. Turski, L. Edwards, P. J. Bouchard, Neutron diffraction residual stress measurements in a 316L stainless steel bead-on-plate weld specimen, International Journal of Pressure Vessels and Piping 86 (1) (2009) 13–19. doi:10.1016/j.ijpvp.2008.11.010.
[43] M. Turski, L. Edwards, Residual stress measurement of a 316l stainless steel bead-on-plate specimen utilising the contour method, International Journal of Pressure Vessels and Piping 86 (1) (2009) 126–131. doi:10.1016/j.ijpvp.2008.11.020.
[44] X. Shan, C. M. Davies, T. Wangsdan, N. P. O'Dowd, K. M. Nikbin, Thermo-mechanical modelling of a single-bead-on-plate weld using the finite element method, International Journal of Pressure Vessels and Piping 86 (1) (2009) 110–121. doi:10.1016/j.ijpvp.2008.11.005.
[45] P. Gilles, W. El-Ahmar, J.-F. Jullien, Robustness analyses of numerical simulation of fusion welding NeT-TG1 application: "Single weld-bead-on-plate", International Journal of Pressure Vessels and Piping 86 (1) (2009) 3–12. doi:10.1016/j.ijpvp.2008.11.012.
[46] S. K. Bate, R. Charles, A. Warren, Finite element analysis of a single bead-on-plate specimen using SYSWELD, International Journal of Pressure Vessels and Piping 86 (1) (2009) 73–78. doi:10.1016/j.ijpvp.2008.11.006.
[47] P. J. Bouchard, The NeT bead-on-plate benchmark for weld residual stress simulation, International Journal of Pressure Vessels and Piping 86 (1) (2009) 31–42. doi:10.1016/j.ijpvp.2008.11.019.
[48] M. C. Smith, A. C. Smith, NeT bead-on-plate round robin: Comparison of transient thermal predictions and measurements, International Journal of Pressure Vessels and Piping 86 (1) (2009) 96–109. doi:10.1016/j.ijpvp.2008.11.016.
[49] J. Goldak, A. Chakravarti, M. Bibby, A new finite element model for welding heat sources, Metallurgical Transactions B 15 (2) (1984) 299–305. doi:10.1007/BF02667333.

Journal publication: https://doi.org/10.1016/j.camwa.2018.10.027

4.4.3 Forecasting melt pool geometries

It was possible to carry out a thorough validation of the transient thermal-solidification model. This validation is discussed in detail in [82] and presented next in form of that publication. The quantities of interest were the width, depth, and length of the melt pool of a SLM process, as well as the associated cooling rates. The measurements were performed for a series of benchmarks which were defined and measured very recently at the National Institute of Standards and Technology in USA [83]. Their goal is to establish industrial benchmarks for SLM processes precisely to validate physical models of the process.

Before the corresponding journal publication is presented, it is worth to consider some remarks on the process of modeling itself in computational analysis. All models have to be calibrated against a fixed reference case. Of course, it is good practice to carry out a model calibration using only parameters that cannot be measured easily. For example, if the shape of the laser beam is not well known, it is a valid choice to use the power distribution of the laser to fit the computational results to the measured results in one reference case. Such a calibration is only accurate up to a certain deviation from the experiment. It is important to note that this first calibration step only proves that the model is actually able to replicate the obtained experimental results. The more interesting task is to determine the *range of validity* of the considered model i.e. how robust the predictions are w.r.t. variations of the input parameters. To this end, the calibrated computational model then has to be executed for different laser speeds and different laser powers, which again correspond to a set of benchmark experiments. However, in the sense of a double-blind comparison, the experimental results should not be available to the computational scientist. This was the case in the computations of the melt pool geometries presented in section 4.4.3. This double-blind comparison revealed that the physical model could actually only predict the melt pool shapes up to 5.9% in length, 19.3% in width, and 18.6% in depth. This large deviation revealed that the physical model required an update to take the convection inside the melt pool into account. This convection was first neglected due to the expected large increase in computational complexity for convection problems. Additionally, even if a corresponding fluid-dynamics model were incorporated, it would only come along with even more unknown parameters to describe the physical process. What, for example, would be the viscosity of the molten metal at these high temperatures? Which turbulence model is valid in these physically extreme situations inside the melt pool?

Therefore, a much simpler approach was chosen. It could then be demonstrated that a simple and valid remedy to physically model the convection inside the melt pool was to introduce anisotropic conductivities at the meso-scale. This model update then delivered deviations of a maximum of only 0.84% for the calibrated case – but it also only led to deviations by a maximum of 6.5% for all other investigated quantities and cases. It has to be added that this physical model is computationally only marginally more expensive than the original one. It is a condensation of local convective effects into just one three-dimensional parameter which could also be obtained by a micro-scale model. Here, it is demonstrated nicely that insight into the developing field of simulations of SLM processes is best gained by an interplay between experiments, computational analysis, and a careful validation of the involved models.

Journal Publication

title:	Accurate prediction of melt pool shapes in laser powder bed fusion by the non-linear temperature equation including phase changes
authors:	S. Kollmannsberger, M. Carraturo, A. Reali, F. Auricchio
published at:	*Integrating Materials and Manufacturing Innovation*
publisher:	Springer
year:	2019
volume:	8
pages:	167–177
doi:	https://doi.org/10.1007/s40192-019-00132-9

myjournal manuscript No.
(will be inserted by the editor)

Accurate prediction of melt pool shapes in laser powder bed fusion by the non-linear temperature equation including phase changes

Model validity: isotropic versus anisotropic conductivity to capture AM Benchmark Test AMB2018-02

Stefan Kollmannsberger · Massimo Carraturo · Alessandro Reali · Ferdinando Auricchio

Received: February 15, 2019/ Accepted: date

Abstract In this contribution, we validate a physical model based on a transient temperature equation (including latent heat) w.r.t. the experimental set AMB2018-02 provided within the additive manufacturing benchmark series, established at the National Institute of Standards and Technology, USA. We aim at predicting the following quantities of interest: width, depth, and length of the melt pool by numerical simulation and report also on the obtainable numerical results of the cooling rate.
We first assume the laser to posses a double ellipsoidal shape and demonstrate that a well calibrated, purely thermal model based on isotropic thermal conductivity is able to predict all the quantities of interest, up to a deviation of maximum 7.3% from the experimentally measured values. However, it is interesting to observe that if we directly introduce, whenever available, the measured laser profile in the model (instead of the double ellipsoidal shape) the investigated model returns a deviation of 19.3% from the experimental values. This motivates a model update by introducing anisotropic conductivity, which is intended to be a simplistic model for heat material convection inside the melt pool. Such an anisotropic model enables the prediction of all quantities of interest mentioned above with a maximum deviation from the experimental values of 6.5%. We note that, although more predictive, the anisotropic model induces only a marginal increase in computational complexity.

Keywords melt pool size · validation · model calibration · laser powder-bed fusion · heat transfer analysis · SLM · laser bed power fusion · metal additive manufacturing

PACS HEAT TRANSFER: 44.05.+e Analytical and numerical techniques · 44.10.+i Heat conduction ·

Mathematics Subject Classification (2010) MSC 65: numerical analysis

Stefan Kollmannsberger and Massimo Carraturo
Technical University of Munich, Germany
Tel.: +49-89-25021
Fax: +49-89-25051
E-mail: stefan.kollmannsberger@tum.de, massimo.carraturo@tum.de

Massimo Carraturo, Ferdinando Auricchio and Alessandro Reali
University of Pavia
Tel.:+39-0382-985476
E-mail: massimo.carraturo01@universitadipavia.it, auricchio@unipv.it, alessandro.reali@unipv.it

1 Introduction

The shape and thermal history of the melt pool are key ingredient to determine the physical properties of an artifact generated through a welding process. Therefore, the prediction of weld pool dynamics has been a subject of intensive research in the last decades in both the experimental and the numerical modeling community of welding; e.g., recent reviews of this subject are provided in [8, 23]. Furthermore, it is fundamental to observe that process-structure-property relationships are also tightly interlinked and strongly determined by the characteristics of the weld pool in laser powder bed fusion (LPBF) additive manufacturing technologies [22]. Therefore, an accurate thermal analysis is a key ingredient in the numerical simulations and predictions of LPBF processes as well.

https://doi.org/10.1007/s40192-019-00132-9

To this end, many physical models have been proposed to obtain accurate and reliable numerical approximations of melt pools. Although different in scale, the basic phenomena in LPBF are similar to those in arc welding processes, see [11, 15, 24] for an overview. Recent summaries more specific to LPBF processes are published in [18, 21, 22]. While particle based models [12] as well as Lattice Boltzmann type approaches [14] exist, most common are continuum models based on the conservation of mass, momentum and energy [25]. Continuum approaches allow for modeling the transient evolution of primal variables (temperatures, pressures, and velocities) taking into account a large number of effects, such as the convection inside the melt pool, also including the one caused by a gradient in the surface tension (Marangoni effect as well as capillary effects), vaporization, momentum losses in mushy zones due to porous media effects, etc.

All these models may deliver very accurate results, but the more effects they include, the more computational power they require. Additionally, the abundance of models comes along with a wealth of parameters: these may be material viscosity, density, thermal conductivity and capacity, latent heat, etc., most of which show a non-negligible temperature dependence, such that their accurate, experimental determination may be both crucial and critical. Further modeling parameters, such as emissivity or absorptivity or even the geometry of powder particles, may come into play and they can be introduced in the model as boundary or initial conditions. However, accurate measurements of many of the listed parameters are not publicly available. This is even true for the most basic parameters, such as heat capacity or thermal conductivity, that are not published for the temperatures involved in metal based LPBF processes. All this drought of information results in the fact that even the evaluation of several parameters may itself often rely on models which, in turn, need to be calibrated against further observations.

The dilemma of choosing a correct model for the case of limited data is an important issue in statistics. As an example, George Box [3] stated, somewhat drastically in his well known aphorism, that all models are wrong and that, therefore, the most complicated model is not necessarily the best. Instead, it is recommended to follow the lines of William of Occam, in which an economical description of the observations is sought which 'is as simple as possible, but not simpler'.

Following this line of thought, the purpose of the present paper is to built and validate an economical model able to replicate the results obtained by the benchmark measurements of a single line laser stroke on a bear metal plate of IN 625 published in [1]. As stated in the chapter CHAL-AMB2018-02-MP of the previous reference, the quantities of interest are the width, depth, and length of the melt pool. Additionally, we also monitor the cooling rate as defined in CHAL-AMB2018-02-CR, even if this quantity is not of primary concern here. To this end, we employ a heat transfer model which considers the different phases of the material as a homogeneous media. This approach is well established in literature and has proven to be effective also in the thermal numerical analysis of large scale LPBF processes [5, 7, 20]. Other successful attempts in this direction, however with a focus on the scale of the melt pool, include the very recent publication of Zhang [26], which provides a summary of previous approaches but most importantly also incorporates anisotropic conductivities, as discussed in the paper at hand. The model proposed in [26] is more elaborate than ours as it also incorporates a spatially variable laser absorptivity and, in this sense, it is not minimal w.r.t. the data set we face. Further, interesting efforts to construct a valid, yet minimal model, are published in [17] where the linking of thermal models to experiments is carried out via surrogate modeling based on multivariate Gaussian processes.

We begin by introducing the widely used physical model based on the transient heat equation including phase changes in section 2. We shortly remark on the verification of this model in section 3, before we move to model validation in section 4. The section on model validation is the main section and commences with reciting the main results of the benchmark cases obtained on two machines, a commercial machine (CBM) and the additive manufacturing metrology testbed (AMMT), both located at the National Institute of Standards and Technology, USA. In an effort to obtain a minimal set of modeling parameters, we evaluate the sensitivities of the quantities of interest to the modeling parameters given by the physical model. We then select only the most relevant modeling parameters and use them to calibrate the physical model towards a similar benchmark already published as case 7 in [9]. We then proceed with the evaluation of our model against the benchmark results on the CBM machine. We observe that the more accurate measurements of the laser profile on the AMMT render the model calibrated to the CBM machine using a double-ellipsoidal heat source less accurate in the AMMT case in which accurate measurements of the laser profile exist. This observation necessitates an update of the model. The model update is presented in subsection 4.3 by incorporating anisotropic conductivity, which is thought to be a simple way to model the convection inside the melt pool. Finally, in section 5 we conclude that given accurate

https://doi.org/10.1007/s40192-019-00132-9

measurements of the profile of the laser, the anisotropic model provides an increase in accuracy over the tested parameter range as compared to the simpler, isotropic physical model.

2 Governing equations

We use a non-linear heat transfer equation as a physical model to describe the evolution of temperature $T = T(t, \mathbf{x})$ as a function of space and time. Given a spatial domain Ω and a time interval $\mathcal{T} = [0, t_{end})$, the heat transfer equation can be written as follows:

$$\rho c \frac{\partial T}{\partial t} + \rho L \frac{\partial f_{pc}}{\partial t} - \nabla \cdot (k \nabla T) = 0 \qquad \text{in} \quad \Omega \times \mathcal{T}. \tag{1}$$

Therein ρ and L describe the density and the latent heat of the material, $c = c(T, \mathbf{x})$ and $k = k(T, \mathbf{x})$ are the temperature dependent heat capacity and thermal conductivity of the material, while $f_{pc} = f_{pc}(T)$ is the phase-change function describing the solid-to-liquid phase transition of the material. Therefore, beside the non linear contribution of the heat capacity and thermal conductivity, the latent heat term of Equation 1 introduces a further nonlinearity into the problem.

Equation 1 is completed by the initial condition at time $t = 0$:

$$T(\mathbf{x}, 0) = T_0 \qquad \text{in} \quad \Omega, \tag{2}$$

as well as Neumann boundary conditions:

$$k \nabla T \cdot \mathbf{n} = q^r + q^l \qquad \text{on} \quad \Gamma_N \times \mathcal{T}. \tag{3}$$

Herein, T_0 is the initial temperature of the body, $\mathbf{n}$ is the unit normal vector, q^l is the heat flux input and q^r is the radiation boundary condition defined as:

$$q^r = \sigma \epsilon \left(T^2 + T_e^2\right)\left(T_e^2 - T^2\right). \tag{4}$$

In Equation 4 σ is the Stefan-Boltzmann constant, ϵ is the emissivity of the material, and T_e is the ambient temperature. In our model, convection boundary conditions are neglected. Further details, specifically the adopted finite element formulation, are provided in [4, 13].

2.1 Phase-change model

For iso-thermal phase changes f_{pc} exhibits a jump at the melting temperature T_m, as the temperature changes the material state from solid to liquid. Since the phase-change for metals is actually non-isothermal, we regularize this sudden jump between two temperatures T_s and T_l, with $T_s < T_l$. We can now define the phase change function f_{pc}, such as:

$$f_{pc}(T) = \frac{1}{2}\left[\left(S \frac{2}{T_l - T_s}\left(T - \frac{T_s + T_l}{2}\right)\right) + 1\right]. \tag{5}$$

The parameter S in Equation 5 is adjusted such that the bulk of the phase change actually occurs between T_s and T_l.

2.2 Heat flux model

In the sequel we consider two variants of the heat flux input q^l. The first variant is the double elliptical model of Goldak [10] described in Figure 1. The front quadrant as is defined by:

$$q^l = \frac{2Q\eta f_f}{\pi a c_f} \exp\left(-2((z^{'} - z_0^{'})/c_f^2 + (x - x_0)/a^2)\right), \tag{6}$$

while in the rear quadrant it takes the form:

$$q^l = \frac{2Q\eta f_r}{\pi a c_r} \exp\left(-2((z^{'} - z_0^{'})/c_r^2 + (x - x_0)/a^2)\right). \tag{7}$$

Herein, Q is the laser power and η is the absorptivity of the material. The geometrical parameters $z_0^{'}$ and x_0 define the center of the laser beam on the upper surface at time t, while f_f and f_r are the fraction of heat deposited in the front and the rear quadrant, respectively.

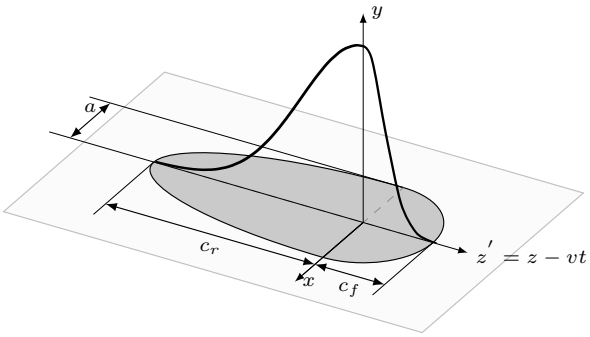


Fig. 1: Goldak model for the heat flux input. The model consists in a double-ellipse on which a gaussian profile is defined.

The second variant the heat source q^l is not a model. In fact, q^l is directly provided to Equation 3 as given by corresponding measurements.

https://doi.org/10.1007/s40192-019-00132-9

3 Model Verification

The computational model was verified against the series of analytical or semi-analytical solutions defined in [13], where a multi-level *hp*-discretization was used. The computational model utilized in the paper at hand is slightly different, as it uses an IGA discretization wherein multi-level Bézier extraction is applied to construct an efficient discretization which is refined locally in the vicinity of the laser beam. This implementation was verified in [6] in two dimensions as well as in three dimensions using the same series of problems which were also used for the verification of the multi-level *hp*-basis [13]. Since the focus of the present contribution is the validation of the model, we will not repeat these extensive verification studies. Instead, in the next section of this paper, we will use the capabilities of the proposed discretization to directly evaluate the validity of the physical model given in section 2.

4 Model Validation

As a preamble to this section we want to highlight the fact that there are situations (e.g. the presence of highly complex phenomena, problem physics still unclear, model uncertainties and difficulties in ascertain its effectiveness, inability to measure all the model parameters) in which model validation must consist of two steps. In the first step (calibration step) the indeterminacy of the physical model is investigated and calibrated against a first set of experimental evidences; in the second step (validation step) the numerical results are compared against a different set of experimental evidences in order to define the range of validity and the robustness of the numerical model. The case under investigation is characterized by the inability to measure all the model parameters, in particular we have limited information on the absorptivity, emissivity, thermal conductivity and heat capacity of the material at high temperature, justifying the choice of the previously defined two-step model validation process.

In subsection 4.1 we will shortly describe and report the experimental benchmarks published in [1]. Following the previously described steps of the validation procedure, first, in subsection 4.2, we calibrate the isotropic model of section 2 using the double ellipsoidal heat source introduced by Goldak [10] and then we validate the isotropic model using the two heat fluxes described in subsection 2.2. For the case in which an accurate measurement of the laser power distribution is given, we observe that the isotropic material assumption has a very limited range of validity. These findings serve as a motivation to extend the physical model by introducing anisotropic conductivities. This extended model is then presented in subsection 4.3 where it will be demonstrated that it predicts weld pool shapes with an improved accuracy.

Parameter values	A	B	C
laser power [W]	150	195	195
laser speed [mm/s]	400	800	1200
laser spot diameter $D4\sigma[\mu m]$	100	100	100

Table 1: CBM machine: parameter values

4.1 Benchmark cases

All benchmark cases are thoroughly defined in the laser additive manufacturing benchmarks published in [1], including a detailed report on the measurements. The benchmarks are obtained through a traveling laser beam on a bear metal plate of nickel-based alloy IN625. The experimental quantities we will use to validate our model are: width, length, and depth of the melt pool as defined in CHAL-AMB2018-02-MP of the above reference. We also report on the cooling rates defined in CHAL-AMB2018-02-CR, although they are not the primal focus in the paper at hand. The benchmarks cited above and reported in [1] were performed on two different machines: a commercial machine (CBM) and the additive manufacturing metrology testbed (AMMT), both located at the National Institute of Standards and Technology, USA. On each machine a set of ten measurements was carried out for three different cases (labeled A, B, and C), i.e. for varying laser power and speed. These cases are specified in Table 1 for the CBM machine and Table 2 for the AMMT machine. The averages of the experimental measurements for the CBM machine are reported in Table 3, while average measurements for the AMMT machine are reported in Table 4. In the first case the cooling rate is defined as:

$$CR = \frac{1290\,[^\circ\mathrm{C}] - 1000\,[^\circ\mathrm{C}]}{\Delta d[mm]} \times v \left[\frac{mm}{sec}\right],$$

with v laser speed and Δd distance in the direction of the laser path, while in the second case as:

$$CR = \frac{1290\,[^\circ\mathrm{C}] - 1190\,[^\circ\mathrm{C}]}{\Delta d[mm]} \times v \left[\frac{mm}{sec}\right].$$

For the exact definition of v and Δd, as well as for further details on the experimental benchmarks we refer to the original website which continues to be updated as further measurements become available [1].

https://doi.org/10.1007/s40192-019-00132-9

Parameter values	A	B	C
laser power [W]	137.9	179.2	179.2
laser speed [mm/s]	400	800	1200
laser spot diameter $D4\sigma[\mu m]$	170	170	170

Table 2: AMMT machine: parameter values

case	length $[\mu m]$	cooling rate $[\frac{^\circ C}{sec}]$
A	659 ± 21	$6.20 \times 10^5 \pm 7.99 \times 10^4$
B	782 ± 21	$9.35 \times 10^5 \pm 1.43 \times 10^5$
C	754 ± 46	$1.28 \times 10^6 \pm 3.94 \times 10^5$

Table 3: CBM machine: experimental measurements according to [1], CHAL-AMB2018-02-MP

case	length $[\mu m]$	width $[\mu m]$	depth $[\mu m]$	cooling rate $[\frac{^\circ C}{sec}]$
A	300	147.9	42.5	1.16×10^6
B	359	123.5	36	1.08×10^6
C	370	106	29.5	1.90×10^6

Table 4: AMMT machine: experimental measurements according to [1], CHAL-AMB2018-02-MP

4.2 Isotropic conductivity model

The calibration step of the isotropic model is carried out for case B on the CBM machine, as given in Table 1, which is exactly the same configuration as case 7 in [9]. The validation step is obtained comparing the calibrated model to the cases A, B and C of Table 3 and Table 4. For all the numerical simulations the IN625 material parameters are taken from literature [2, 19] and are reported in Table 5, Figure 2, and Figure 3.

Table 5: Material and process constant parameters

density	8.44e-6 [kg/mm^3]
latent heat	2.8e5 [J/kg]
melting temperature interval	1290 - 1350 [°C]

It is noteworthy that material and process parameters, necessary to run the numerical simulation, are not experimentally available for the effective temperatures occurring in LPBF processes. For example, the measurement of the thermal conductivity k in Figure 2 is only available up to 871°C, but the melting range for IN625 is 1290-1350°C. Likewise, the melting temperature interval the value of k can only be extrapolated. It is important to note that this extrapolation itself represents a physical model which, in turn, needs

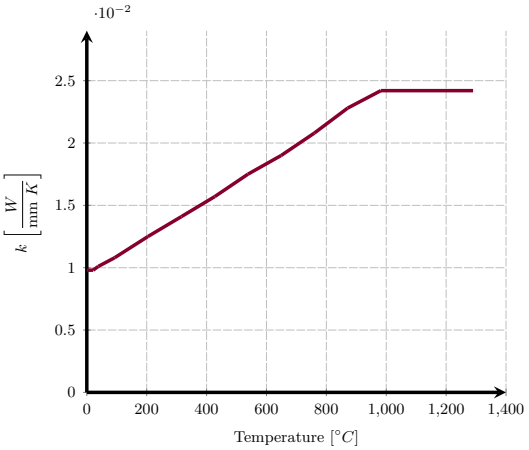


Fig. 2: Conductivity vs. Temperature

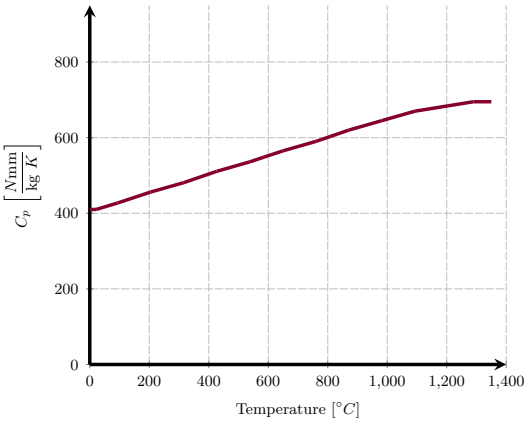


Fig. 3: Heat capacity vs. Temperature

to be calibrated. This circumstance is used in subsection 4.3 to better describe the conductivity of the material and, consequently, improve the accuracy of the predicted melt pool geometry. Further coefficients, whose measurements are only available up to a certain temperature, are the absorptivity η and the emissivity ϵ. The latter necessary to define the radiation boundary condition given in Equation 3.

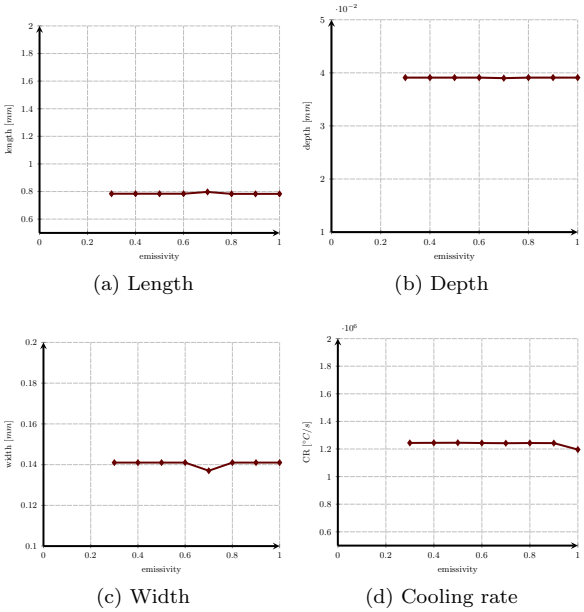
In case of the CBM machine, the benchmark defines the laser spot radius equal to $50\mu m$. We utilize this value for both parameters c_f and a of the double elliptical model (see Figure 1). However, our model also contains the radius ratio c_r/c_f as a model parameter as well as the power fraction f_f/f_r, which has the side condition that $f_r + f_f = 2$. Both these parameters are additional, potential candidates to calibrate the physical model.

https://doi.org/10.1007/s40192-019-00132-9

Model Calibration
Model calibration first requires to identify the sensitivities of the quantities of interest, i.e., length, width, depth, and cooling rate at the wake of the melt pool w.r.t. the modeling parameters η, ϵ, f_f/f_r and c_r/c_f given by the physical model presented in section 2. To this end, four studies were carried out, and for each study a single parameter is varied while the others stay fixed. In the following we only present the sensitivity studies w.r.t. the absorptivity η and the emissivity ϵ of the material due to editorial space limitations of this paper. Figure 4 presents the variation of the length, width, depth, and cooling rate w.r.t. the emissivity values. The study clearly suggests that there is practically no influence of the emissivity on the quantities of interest. At first sight this result comes as a surprise because the model of the boundary conditions suggests an influence of fourth order in the temperature, something that for sure can not be neglected. Indeed, numerous authors explicitly include this boundary condition to obtain good results, see for example [16] and the references cited therein. However, the investigation at hand considers the temperature directly under or in close vicinity to the laser and, therefore, the contribution of the radiation boundary condition is marginal. To illustrate this effect, we consider the flux caused by radiation at the melting temperature $T_m = 1290°\text{C} = 1563.15\text{K}$, with an ambient temperature of $T_e = 20°\text{C} = 293.15\text{K}$. The corresponding power loss is

$$5.67 \times 10^{-8} \times 0.47 \left(T_m^2 + T_e^2\right)\left(T_e^2 - T_m^2\right) = 1.59 \times 10^5 \left[\frac{\text{W}}{m^2}\right] = 0.16 \left[\frac{\text{W}}{mm^2}\right], \quad (8)$$

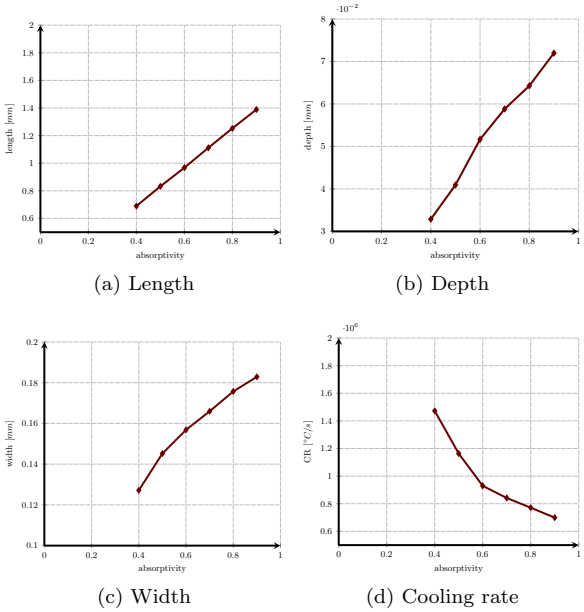
which represents a negligible quantity compared to the peak power density of $q^l = 2.33 \times 10^4 \left[W/mm^2\right]$ in the center of the laser beam. Clearly, under these conditions, radiation itself may be neglected for studies of temperature fields in close proximity to the laser source. To the contrary, the absorptivity has a large influence (see Figure 5), as do the power fraction and the radius ratio[1]. An iterative calibration delivers the final choice of the parameters: $\varepsilon = 0.47$, $\eta = 0.38$, $f_f/f_r = 0.053$ and $c_r/c_f = 0.167$.



(a) Length (b) Depth

(c) Width (d) Cooling rate

Fig. 4: CBM machine. Sensitivity studies w.r.t. the emissivity of the material ϵ



(a) Length (b) Depth

(c) Width (d) Cooling rate

Fig. 5: CBM machine. Sensitivity studies w.r.t. the absorptivity of the material η

Numerical results for calibration
Numerically computed temperature curves along the laser path are depicted in Figure 6. The figure reports also the experimentally measured temperature, carried out using *in-situ* thermography, as described in [9]. The

[1] The latter two are not depicted due to limitations of space in this article

https://doi.org/10.1007/s40192-019-00132-9

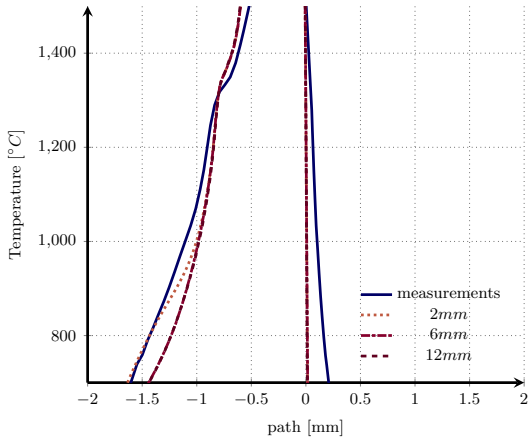


Fig. 6: Computation of the temperature profile calibrated to case 7 in [9]

different curves labeled $2mm$, $6mm$ and $12mm$ indicate at which position the zero of the abscissa of the plot coincides with the laser path. A steady state is reached already after only $2mm$. We specifically note that the calibration was carried out to best capture the temperature range around the melting temperature. Larger, even unphysical deviations, are tolerated outside this region. This kind of calibration towards a process window is justifiable not only due to the fact that merely the region of interest needs to be captured with accuracy by the computations, but also because the camera itself delivers its most accurate measurements in that range. The plot also directly shows where the numerical model is not valid, namely directly inside the melt pool. Here, the temperature drastically overshoots to unrealistically high values. The very good agreement of the computation in the range of the melting zone is further confirmed in Figure 7. This figure overlays the image of the cross section of the track taken by an ex-situ measurement of a confocal laser scanning microscope (CLSM) with the calibrated computation. Both Figure 6 and Figure 7 demonstrate that it is possible to obtain an excellent agreement with the experiment using the simple physical model presented in section 2, if η, f_f/f_r, and c_r/c_f serve as model calibration parameters.

Model validation for the CBM machine

The calibrated model delivers the results depicted in Table 6. It can readily be concluded that the model is able to predict the length of the weld pool up to at least 7.3% accuracy in the parameter range covered by cases A to C. The prediction of cooling rates is approximately one order less accurate. It is interesting to note

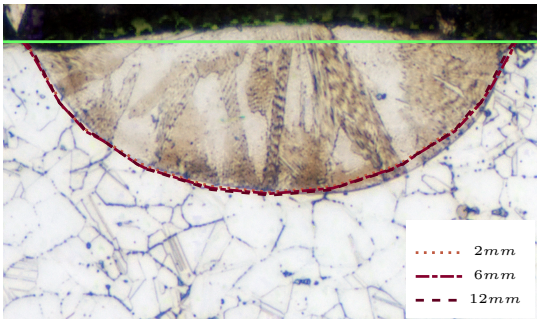


Fig. 7: Computation of cross section (red line) calibrated to case 7 in [9]

case	meas. l[μm]	num. l[μm]	Δ [%]	meas. CR[$\frac{°C}{sec}$]	num. CR [$\frac{°C}{sec}$]	Δ [%]
A	659	707	7.3	6.20×10^5	8.79×10^5	41.8
B	782	812	3.8	9.35×10^5	1.35×10^6	44.3
C	754	772	2.4	1.28×10^6	2.09×10^6	63.3

Table 6: CBM machine: obtained weld pool length l and cooling rates cr

that the measured length in [9], i.e., the length towards which the model was calibrated was $813[\mu m]$, provided with a tolerance of $\pm 79[\mu m]$. However, measurements performed in [1] for exactly the same case (case B) were more accurate and are given as $782 \pm 21[\mu m]$, see also Table 6. Thus, a re-calibration of the model to case B will likely deliver more accurate predictions for the cases A and C. However, this was not carried out because even more accurate measurements are available for the AMMT machine which lead to the development of the extended physical model presented in subsection 4.3.

Model validation for the AMMT machine

Surprisingly, very different experimental results were obtained with the same scan parameters at the AMMT machine as compared to the CBM machine. Due to this reason, more thorough studies were carried out on the AMMT machine. These include measurements of the actual laser profile itself. These measurements, now published in [1], enable their direct application as the Neumann boundary condition q^l in Equation 3. Thus, the physical model presented in section 2 is more tightly defined. This generates an interesting situation from the perspective of model validation because two (influential) calibration parameters, the power fraction f_f/f_r and the radius ratio c_r/c_f are now fixed and, therefore, can not be used for calibration. Given that the emissivity ϵ has practically no influence, the absorp-

https://doi.org/10.1007/s40192-019-00132-9

case	length $[\mu m]$	width $[\mu m]$	depth $[\mu m]$	cooling rate $[\frac{°C}{sec}]$
A	301	119	52	0.91×10^6
B	360	103	42	1.33×10^6
C	348	91	32	2.18×10^6

Table 7: AMMT machine: computed values

case	length $\Delta[\%]$	width $\Delta[\%]$	depth $\Delta[\%]$	cooling rate $\Delta[\%]$
A	0.47	19.3	18.6	21.6
B	0.11	16.4	15.8	23.1
C	5.9	14.2	10.1	14.7

Table 8: AMMT machine: deviations from experimental values

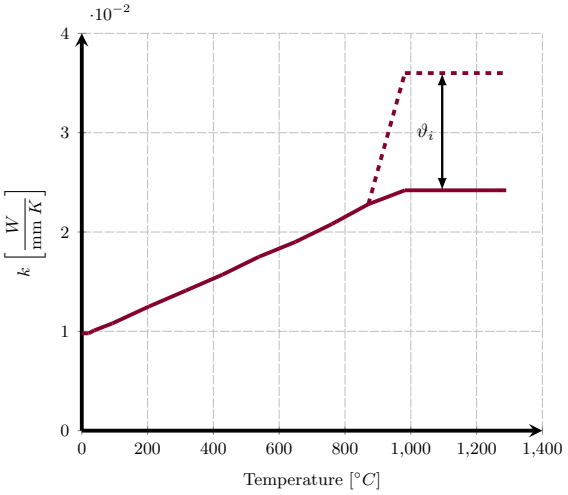


Fig. 8: Scaling of the conductivity in direction $i = x, y, z$

tivity η is the only parameter left for a re-calibration. For $\eta = 0.086$ we obtain the numerical results provided in Table 7. The corresponding deviations are provided in Table 8. While the deviations in the length are still at a maximum of approx. 6%, width and depth are only predicted to an accuracy of 20%. No further calibration is possible as there is only one parameter to calibrate but three values of interest to fit (excluding the cooling rate). This clearly shows the boundaries of validity of the model presented in section 2 and motivates the development of the model discussed in the next section.

4.3 Anisotropic conductivity model

Two possible modifications are readily imaginable: a) a definition of an absorptivity field instead of a scalar value η and b) the definition of an anisotropic conductivity. The former could be motivated by the fact that the melt pool surface will surely cause the absorption of the laser energy to be non-constant. However, to the authors opinion a good model should be as simple as possible, yet replicate the observed effects as accurately as possible. With this objective in mind, the definition of an anisotropic conductivity is a more attractive choice. The only change necessary is that the scalar value k in Equation 1 changes to $\mathbf{k}$, a diagonal matrix with the entries $diag(k_x,\ k_y,\ k_z)$. Further, we set $\epsilon = 0$. The physical motivation for this model is that the (transient) diffusion equation given by Equation 1 does by no means include the effects caused by convective heat transfer inside the weld pool. This flaw has already inspired other authors e.g., [16] to use a strongly increased conductivity k inside the melt pool to model convective effects. We now extend this idea by choosing anisotropic values. For simplicity, we introduce the scaling factor ϑ_i where $i = \{x, y, z\}$ such that $\mathbf{k} = diag(k\vartheta_x, k\vartheta_y, k\vartheta_z)$. The values for ϑ_i deviate from 1 only after the last obtainable measurement (at $T = 871°C$) of the conductivity as depicted in Figure 8. After calibration to the AMMT machine B we obtain the set $\vartheta_x = 1.0, \vartheta_y = 1.4, \vartheta_z = 0.9$. This delivers very well matching weld pool geometries. While the effect of the scaling of k is marginal in a temperature plot along the length (because here $\vartheta_x = 1.0$), its effect in the cross-section is quite pronounced (see Figure 9 for a direct overlay of the melt pool geometry over the cross section). In the validation step, we keep the calibration parameters fixed, i.e., $\vartheta_x = 1.0$, $\vartheta_y = 1.4$, and $\vartheta_z = 0.9$ and we compute cases A and C of the AMMT machine. The computed values are provided in Table 9 and the corresponding deviations are provided in Table 10. We observe that for the anisotropic model the maximum deviation of length, width and depth is 6.49% at worst while, for the isotropic conductivity model it was merely 19.3%. Even the forecast of the cooling rates has improved sightly.

case	length $[\mu m]$	width $[\mu m]$	depth $[\mu m]$	cooling rate $[\frac{°C}{sec}]$
A	304	146.4	44.6	0.82×10^6
B	362	123.7	36.1	1.23×10^6
C	346	105.1	27.3	1.88×10^6

Table 9: Anisotropic conductivity model: computed values

https://doi.org/10.1007/s40192-019-00132-9

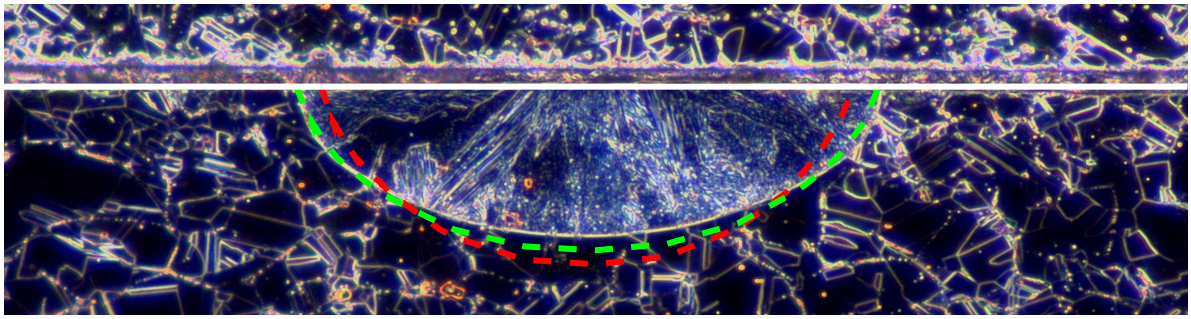


Fig. 9: Melt pool cross section micrograph image 50×DF (from `https://phasedata.nist.gov/rest/blob?id=5b102edd4407e700870ff13e`) over computed cross sections using isotropic (dashed red line) and anisotropic conductivity (dashed green line)

case	length Δ[%]	width Δ[%]	depth Δ[%]	cooling rate Δ[%]
A	1.33	1.0	2.5	29.3
B	0.84	0.02	0.2	13.9
C	6.49	0.8	5.1	1.3

Table 10: Anisotropic conductivity model: deviations of computed values from experimental values

5 Summary and Conclusions

In this contribution we used the standard heat diffusion model to predict the length, width and depth of the melt pool in the laser additive manufacturing benchmarks CHAL-AMB2018-02-MP published in [1]. The physical model included a latent heat term as published e.g., in [4] along with a radiation boundary condition. Within this model we found the radiation boundary condition to have little to no influence upon the quantities of interest. This is due to the fact that in close proximity of the laser beam impact region, the power lost by radiation is much lower than the applied laser energy itself.

As a first approach we assumed the laser source to possess the well known double elliptical shape as proposed for welding by Goldak [10]. We demonstrated that this model is well suited to predict the shape of the weld pool as it delivered a maximum deviation from the measurements of 7.3%. However, in case the shape of the laser source is given by a measurement, the standard, transient heat diffusion model only provides accuracies of 19.1% for the investigated benchmark cases. This renders it practically invalid.

We then extended the isotropic thermal model by introducing anisotropic conductivities. Their physical interpretation is to model anisotropic convection inside the melt pool. This slight extension enabled the model to deliver at worst 6.49% deviations in length, width and depth of the melt pool. Therefore, we conclude that the introduction of an anisotropic conductivity is a simple, yet effective way to improve the physical model based on transient heat equation including phase changes and remark that the added computational effort for this extension is marginal.

Acknowledgements The first author gratefully acknowledges the financial support of the German Research Foundation (DFG) under grant RA 624/27-2. This work was partially supported by Regione Lombardia through the project "TPro.SL - Tech Profiles for Smart Living" (No. 379384) within the Smart Living program, and through the project "MADE4LO - Metal ADditivE for LOmbardy" (No. 240963) within the POR FESR 2014-2020 program. Massimo Carraturo and Alessandro Reali have been partially supported by Fondazione Cariplo - Regione Lombardia through the project "Verso nuovi strumenti di simulazione super veloci ed accurati basati sull'analisi isogeometrica", within the program RST - rafforzamento.

References

1. AM Bench benchmark challenge CHAL-AMB2018-02-MP. https://www.nist.gov/ambench/amb2018-02-description
2. Special metals corporation. http://www.specialmetals.com
3. Box, G.: Science and Statistics. Journal of the American Statistical Association **71**(356), 791–799 (1976). DOI 10.2307/2286841
4. Celentano, D., Oñate, E., Oller, S.: A temperature-based formulation for finite element analysis of generalized phase-change problems. International Journal for Numerical Methods in Engineering **37**(20), 3441–3465 (1994)
5. Chiumenti, M., Neiva, E., Salsi, E., Cervera, M., Badia, S., Moya, J., Chen, Z., Lee, C., Davies, C.:

Numerical modelling and experimental validation in Selective Laser Melting. Additive Manufacturing **18**, 171–185 (2017). DOI 10.1016/j.addma.2017.09.002

6. D'Angella, D., Kollmannsberger, S., Rank, E., Reali, A.: Multi-level Bézier extraction for hierarchical local refinement of Isogeometric Analysis. Computer Methods in Applied Mechanics and Engineering **328**, 147–174 (2018). DOI 10.1016/j.cma.2017.08.017
7. Denlinger, E.R., Jagdale, V., Srinivasan, G., El-Wardany, T., Michaleris, P.: Thermal modeling of Inconel 718 processed with powder bed fusion and experimental validation using in situ measurements. Additive Manufacturing **11**, 7–15 (2016). DOI 10.1016/j.addma.2016.03.003. 00005
8. Fotovvati, B., Wayne, S.F., Lewis, G., Asadi, E.: A Review on Melt-Pool Characteristics in Laser Welding of Metals. Advances in Materials Science and Engineering **2018**, 1–18 (2018). DOI 10.1155/2018/4920718
9. Ghosh, S., Ma, L., Levine, L.E., Ricker, R.E., Stoudt, M.R., Heigel, J.C., Guyer, J.E.: Single-Track Melt-Pool Measurements and Microstructures in Inconel 625. JOM (2018). DOI 10.1007/s11837-018-2771-x
10. Goldak, J., Chakravarti, A., Bibby, M.: A new finite element model for welding heat sources. Metallurgical Transactions B **15**(2), 299–305 (1984). DOI 10.1007/BF02667333
11. Goldak, J.A., Akhlaghi, M.: Computational Welding Mechanics. Springer, New York (2005)
12. Khairallah, S.A., Anderson, A.T., Rubenchik, A., King, W.E.: Laser powder-bed fusion additive manufacturing: Physics of complex melt flow and formation mechanisms of pores, spatter, and denudation zones. Acta Materialia **108**, 36–45 (2016). DOI 10.1016/j.actamat.2016.02.014
13. Kollmannsberger, S., Özcan, A., Carraturo, M., Zander, N., Rank, E.: A hierarchical computational model for moving thermal loads and phase changes with applications to selective laser melting. Computers & Mathematics with Applications **75**(5), 1483–1497 (2018). DOI 10.1016/j.camwa.2017.11.014
14. Körner, C., Attar, E., Heinl, P.: Mesoscopic simulation of selective beam melting processes. Journal of Materials Processing Technology **211**(6), 978–987 (2011). DOI 10.1016/j.jmatprotec.2010.12.016
15. Lindgren, L.E.: L.E. Lindgren-Computational Welding Mechanics Thermomechanical and Microstructural Simulations. Woodhead Publishing Limited and CRC Press LLC (2007)
16. Lu, X., Lin, X., Chiumenti, M., Cervera, M., Li, J., Ma, L., Wei, L., Hu, Y., Huang, W.: Finite element analysis and experimental validation of the thermomechanical behavior in laser solid forming of Ti-6Al-4V. Additive Manufacturing **21**, 30–40 (2018). DOI 10.1016/j.addma.2018.02.003
17. Mahmoudi, M., Tapia, G., Karayagiz, K., Franco, B., Ma, J., Arroyave, R., Karaman, I., Elwany, A.: Multivariate Calibration and Experimental Validation of a 3D Finite Element Thermal Model for Laser Powder Bed Fusion Metal Additive Manufacturing. Integrating Materials and Manufacturing Innovation **7**(3), 116–135 (2018). DOI 10.1007/s40192-018-0113-z
18. Megahed, M., Mindt, H.W., N'Dri, N., Duan, H., Desmaison, O.: Metal additive-manufacturing process and residual stress modeling. Integrating Materials and Manufacturing Innovation **5**(1) (2016). DOI 10.1186/s40192-016-0047-2
19. Mills, K.C.: Recommended Values of Thermophysical Properties for Selected Commercial Alloys. Woodhead, Cambridge (2002). OCLC: ocm47677674
20. Riedlbauer, D., Scharowsky, T., Singer, R.F., Steinmann, P., Körner, C., Mergheim, J.: Macroscopic simulation and experimental measurement of melt pool characteristics in selective electron beam melting of Ti-6Al-4V. The International Journal of Advanced Manufacturing Technology **88**(5-8), 1309–1317 (2017). DOI 10.1007/s00170-016-8819-6
21. Schoinochoritis, B., Chantzis, D., Salonitis, K.: Simulation of metallic powder bed additive manufacturing processes with the finite element method: A critical review. Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture **231**(1), 96–117 (2017). DOI 10.1177/0954405414567522
22. Smith, J., Xiong, W., Yan, W., Lin, S., Cheng, P., Kafka, O.L., Wagner, G.J., Cao, J., Liu, W.K.: Linking process, structure, property, and performance for metal-based additive manufacturing: Computational approaches with experimental support. Computational Mechanics **57**(4), 583–610 (2016). DOI 10.1007/s00466-015-1240-4
23. Svenungsson, J., Choquet, I., Kaplan, A.F.: Laser Welding Process – A Review of Keyhole Welding Modelling. Physics Procedia **78**, 182–191 (2015). DOI 10.1016/j.phpro.2015.11.042
24. Tanaka, M.: An introduction to physical phenomena in arc welding processes. Welding International **18**(11), 845–851 (2004). DOI 10.1533/wint.2004.3342

25. Yan, Z., Liu, W., Tang, Z., Liu, X., Zhang, N., Li, M., Zhang, H.: Review on thermal analysis in laser-based additive manufacturing. Optics & Laser Technology **106**, 427–441 (2018). DOI 10.1016/j.optlastec.2018.04.034
26. Zhang, Z., Huang, Y., Rani Kasinathan, A., Imani Shahabad, S., Ali, U., Mahmoodkhani, Y., Toyserkani, E.: 3-Dimensional heat transfer modeling for laser powder-bed fusion additive manufacturing with volumetric heat sources based on varied thermal conductivity and absorptivity. Optics & Laser Technology **109**, 297–312 (2019). DOI 10.1016/j.optlastec.2018.08.012

https://doi.org/10.1007/s40192-019-00132-9

5 Summary, outlook and conclusion

The Finite Cell Method is an embedded domain method of higher order. It was the first of its kind in the sense that it preserves the higher order accuracy of the underlying finite element discretization even if the geometry of the physical domain is discretized in a non-boundary conforming fashion. In recent years, the FCM has matured to a versatile tool in computational analysis avoiding the burden of mesh generation. It does not stand alone any more. It has had influence on or even sparked other, rather similar techniques such as the immersogeometric analysis [84, 85] and in the coarse of its development it was influenced by techniques such as CUTfem [86]. Other very closely related variants such as the isogeometric B-Rep analysis [87, 88] also set out to integrate CAD and computational analysis. A very recent treatise by de Prenter [89] nicely relates these close relatives of FCM in the numerical context of preconditioning the resulting equation systems. Even counting the numerous citations given throughout this treatise, this short summary remains incomplete but serves to demonstrate that many scientists continue to unleash the power of embedded domain modeling.

As laid out in this treatise, numerous engineering applications have already proven to benefit in the context of the FCM. Among these are structural dynamics, large scale structural analysis of CT scans for mechanical and biomedical applications, as well as the simulation of production processes such as additive manufacturing. In summary, it may be stated that there seem to be only little bounds left as to what can be analyzed in the framework of FCM concerning a forward solution of linear problems.

Yet, no method is perfect for every application and, thus, many challenges remain. Among these is the treatment of strong non-linearities of the kind that occur in the simulation of stresses caused by finite strains combined with plasticity. Even though recent contributions address this problem [37, 90, 91], instabilities might still occur. In these extreme situations a good preconditioning, the choice of the appropriate continuity of the underlying discretization, a fine tuned balance between penalizing the material of the fictitious domain w.r.t. the physical domain, the appropriate numerical integration of the physical domain, a suitable formulation of the material law in the fictitious domain as well as the cunning combination of all these numerical techniques help but their interplay and boundaries are not yet fully understood.

Further open questions are related to bridging scales. It could be demonstrated that the Finite Cell Method combined with the multi-level *hp*-method is able to spatially resolve very large scales. Several ways exist from homogenization techniques to bruit force computations of very large structures involving astonishing detail. Many of these techniques are further being developed under the roof of the FCM. However, some applications such as the process of additive manufacturing also involve vast temporal scales on expanding domains which can not yet be resolved. To this end, the combination of the FCM with the multi-level *hp*-method in the framework of space-time finite elements promises to offer the possibility to increase the temporal resolution. First, preliminary work on this subject is now taken up to further push the boundaries of the method.

Other fields of engineering application seem promising as well. Very recently, extremely accurate predictions for three-dimensional fracture of solids were possible. It is the hope that this may be useful in geo-sciences to harvest geothermal energy, for example. Yet another interesting application which surely will be pursued further lies in structural heritage where

the fundamental feature of the FCM to connect geometric and computational models is coming of age in the point cloud version of the FCM.

It will be fun to see where else embedded domain modeling in general, and the Finite Cell Method in particular, may engender new insights in understanding, replicating and forecasting the mechanical behavior of artifacts.

References

[1] C. Peskin. *Flow patterns around heart valves: A digital computer method for solving the equations of motion.* PhD thesis, Albert Einstein College of Medicine, 1972. URL: http://www.umi.com/hp/Products/DisExpress.html.

[2] V. K. Saul'ev. A method for automatization of the solution of boundary value problems on high performance computers. *Dokl. Akad. Nauk SSSR 144 (1962), 497-500 (in Russian). English translation in Soviet Math. Dokl.*, 3:763–766, 1963.

[3] V. K. Saul'ev. On solution of some boundary value problems on high performance computers by fictitious domain method. *Siberian Mathematical Journal*, 4:912–925, 1963.

[4] N. Moës, J. Dolbow, and T. Belytschko. A finite element method for crack growth without remeshing. *International Journal for Numerical Methods in Engineering*, 64:131–150, 1999.

[5] I. Babuška and J.M. Melenk. The partition of unity finite element method: Basic theory and applications. *Computer Methods in Applied Mechanics and Engineering*, 139:289–314, 1996.

[6] T. Belytschko and T. Black. Elastic crack growth in finite elements with minimal remeshing. *International Journal for Numerical Methods in Engineering*, 45:601–620, 1999.

[7] J. Dolbow, N. Moës, and T. Belytschko. Discontinuous enrichment in finite elements with a partition of unity method. *Finite Elements in Analysis and Design*, 36(3-4):235–260, 2000.

[8] C. Peskin. The Immersed Boundary Method. *Acta Numerica*, 11:1–39, 2002.

[9] R. Mittal and G. Iaccarino. Immersed Boundary Method. *Annual Review Fluid Mechanics*, 37:239–260, 2005.

[10] 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.

[11] J. Parvizian, A. Düster, and E. Rank. Finite cell method – h- and p-extension for embedded domain problems in solid mechanics. *Computational Mechanics*, 41:121–133, 2007.

[12] 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, 2008.

[13] 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. doi:DOI10.1002/nme.3289.

[14] M. Dauge, A. Düster, and E. Rank. Theoretical and numerical investigation of the finite cell method. *Journal of Scientific Computing*, 65:1039–1064, 2015. doi:DOI:10.1007/s10915-015-9997-3.

[15] E. Rank, S. Kollmannsberger, Ch. Sorger, and A. Düster. Shell Finite Cell Method: A High Order Fictitious Domain Approach for Thin-Walled Structures. *Computer Methods in Applied Mechanics and Engineering*, 200:3200–3209, 2011. doi:10.1016/j.cma.2011.06.005.

[16] C. Sorger, F. Frischmann, S. Kollmannsberger, and E. Rank. TUM.GeoFrame: Automated high-order hexahedral mesh generation for shell-like structures. *Engineering with Computers*, 30(1):41–56, 2014. doi:10.1007/s00366-012-0284-8.

[17] 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.

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

[19] S. Kollmannsberger, A. Düster, Ch. Sorger, and E. Rank. The Shell Finite Cell Method. In *Trends and Challanges In Computational Mechanics*, Padova, Italy, May 2011.

[20] E. Rank, M. Ruess, S. Kollmannsberger, D. Schillinger, and A. Düster. Geometric modeling, Isogeometric Analysis and the Finite Cell Method. *Computer Methods in Applied Mechanics and Engineering*, 249–252:104–115, 2012. doi:10.1016/j.cma.2012.05.022.

[21] R. Schmidt, R. Wüchner, and K.-U. Bletzinger. Isogeometric analysis of trimmed nurbs geometries. *Computer Methods in Applied Mechanics and Engineering*, 241-244:93–111, Oct 2012. doi:10.1016/j.cma.2012.05.021.

[22] B. Wassermann, S. Kollmannsberger, T. Bog, and E. Rank. From geometric design to numerical analysis: A direct approach using the finite cell method on constructive solid geometry. *Computers & Mathematics with Applications*, 74(7):1703–1726, 2017. doi:10.1016/j.camwa.2017.01.027.

[23] Fady Massarwi and Gershon Elber. A B-spline based framework for volumetric object modeling. *Computer-Aided Design*, 78:36–47, September 2016. doi:10.1016/j.cad.2016.05.003.

[24] E. Cohen, T. Martin, R.M. Kirby, T. Lyche, and R.F. Riesenfeld. Analysis-aware modeling: Understanding quality considerations in modeling for isogeometric analysis. *Computer Methods in Applied Mechanics and Engineering*, 199:334–356, 2010.

[25] Wassermann B., Kollmannsberger S., Kudela L., Rank E., and Shuohui Y. Direct simulation of geometrical models using the finite cell method. In *Proceedings of CAD'17*, Okayama, Japan, Aug 2017. doi:10.14733/cadconfP.2017.160-164.

[26] Benjamin Wassermann, Stefan Kollmannsberger, Shuohui Yin, László Kudela, and Ernst Rank. Integrating CAD and numerical analysis: 'Dirty geometry' handling using the Finite Cell Method. *Computer Methods in Applied Mechanics and Engineering*, 351:808–835, Juli 2019. doi:10.1016/j.cma.2019.04.017.

[27] László Kudela, Stefan Kollmannsberger, Umut Almac, and Ernst Rank. Direct structural analysis of domains defined by point clouds. *Computer Methods in Applied Mechanics and Engineering*, accepted for publication, 2019.

[28] M. Elhaddad, N. Zander, T. Bog, L. Kudela, S. Kollmannsberger, J. Kirschke, T. Baum, M. Ruess, and E. Rank. Multi-level hp-finite cell method for embedded interface problems with application in biomechanics. *International Journal for Numerical Methods in Biomedical Engineering*, page online, 2017. doi:10.1002/cnm.2951.

[29] K. Kolev, T. Brox, and D. Cremers. Fast Joint Estimation of Silhouettes and Dense 3D Geometry from Multiple Images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(3):493–505, 2012.

[30] Pierre Moulon, Pascal Monasse, Renaud Marlet, and Others. OpenMVG. An Open Multiple View Geometry library. https://github.com/openMVG/openMVG.

[31] L. Kudela, U. Almac, S. Kollmannsberger, and E. Rank. Direct Numerical Analysis of Historical Structures Represented by Point Clouds. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 11196 LNCS:64–75, 2018. doi:10.1007/978-3-030-01762-0_6.

[32] B.A. Szabó and I. Babuška. *Finite element analysis*. John Wiley & Sons, 1991.

[33] 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. doi:DOI10.1002/nme.3289.

[34] A. Abedian, J. Parvizian, A. Düster, H. Khademyzadeh, and E. Rank. Performance of different integration schemes in facing discontinuities in the finite cell method. *International Journal of Computational Methods*, 10(3):1350002/1–24, 2013. doi:10.1142/S0219876213500023.

[35] 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(5):457–471, 2012. doi:10.1002/nme.4269.

[36] B. Müller, F. Kummer, and M. Oberlack. Highly accurate surface and volume integration on implicit domains by means of moment-fitting. *International Journal for Numerical Methods in Engineering*, 96:512–528, 2013. doi:DOI:10.1002/nme.4569.

[37] S. Hubrich and A. Düster. Numerical integration for nonlinear problems of the finite cell method using an adaptive scheme based on moment fitting. *Computers & Mathematics with Applications*, 77(7):1983–1997, 2019. doi:https://doi.org/10.1016/j.camwa.2018.11.030.

[38] T. Rüberg and F. Cirak. A fixed-grid b-spline finite element technique for fluid-structure interaction. *International Journal for Numerical Methods in Fluids*, in press, 2013.

[39] 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. URL: http://doi.wiley.com/10.1002/nme.4522, doi:10.1002/nme.4522.

[40] László Kudela, Nils Zander, Tino Bog, Stefan Kollmannsberger, and Ernst Rank. Efficient and accurate numerical quadrature for immersed boundary methods. *Advanced Modeling and Simulation in Engineering Sciences*, 2(1):1–22, Juni 2015. doi:10.1186/s40323-015-0031-y.

[41] László Kudela, Nils Zander, Stefan Kollmannsberger, and Ernst Rank. Smart octrees: Accurately integrating discontinuous functions in 3D. *Computer Methods in Applied Mechanics and Engineering*, 306:406–426, Juli 2016. doi:10.1016/j.cma.2016.04.006.

[42] Simeon Hubrich, Paolo Di Stolfo, László Kudela, Stefan Kollmannsberger, Ernst Rank, Andreas Schröder, and Alexander Düster. Numerical integration of discontinuous functions: Moment fitting and smart octree. *Computational Mechanics*, pages 1–19, Juli 2017. doi:10.1007/s00466-017-1441-0.

[43] I. Babuška. The finite element method with penalty. *Mathematics of Computation*, 27:221–228, 1973.

[44] J. Nitsche. Über ein Variationsprinzip zur Lösung von Dirichlet-Problemen bei Verwendung von Teilräumen, die keinen Randbedingungen unterworfen sind. *Abhandlungen aus dem Mathematischen Seminar der Universität Hamburg*, 36(1):9–15, 1971.

[45] Martin Ruess, David Tal, Nir Trabelsi, Zohar Yosibash, and Ernst Rank. The finite cell method for bone simulations: verification and validation. *Biomechanics and modeling in mechanobiology*, 11:425–437, 2012.

[46] R. Codina and J. Baiges. Approximate imposition of boundary conditions in immersed boundary methods. *International Journal for Numerical Methods in Engineering*, 80:1379–1405, 2009. doi:10.1002/nme.2662.

[47] S. Kollmannsberger, A. Özcan, J. Baiges, M. Ruess, E. Rank, and A. Reali. Parameter-free, weak imposition of Dirichlet boundary conditions and coupling of trimmed and non-conforming patches. *International Journal for Numerical Methods in Engineering*, 101(9):1–30, 2014. doi:10.1002/nme.4817.

[48] D. Schillinger, I. Harari, M.S. Hsu, D. Kamensky, S. Stoter, Z. Yu, and Y. Zhao. The non-symmetric Nitsche method for the parameter-free imposition of weak boundary and coupling conditions in immersed finite elements. *Computer Methods in Applied Mechanics and Engineering*, 309:625–652, 2016.

[49] B. Schottt. *Stabilized Cut Finite Element Methods for Complex Interface Coupled Flow Problems*. PhD thesis, Fakultät für Maschinenwesen, Technische Universität München, 2017.

[50] T.-P. Fries and T. Belytschko. The extended/generalized finite element method: An overview of the method and its applications. *International Journal for Numerical Methods in Engineering*, 84(3):253–304, 2010.

[51] J.M. Melenk and I. Babuška. The partition of unity finite element method: basic theory and applications. *Computer Methods in Applied Mechanics and Engineering*, 139:289–314, 1996.

[52] M. Joulaian and A. Düster. Local enrichment of the finite cell method for problems with material interfaces. *Computational Mechanics*, 52:741–762, 2013. doi:10.1007/s00466-013-0853-8.

[53] S. Kollmannsberger, A. Reali, A. Özcan, M. Ruess, J. Baiges, and E. Rank. Parameter free weak boundary and coupling conditions for IGA. In *Proc. III South-East European Conf. On Computational Mechanics*, Kos, Greece, Juni 2013.

[54] A. Gerstenberger and W. Wall. An embedded Dirichlet formulation for 3D continua. *International Journal for Numerical Methods in Engineering*, 82:537–563, 2010.

[55] Yujie Guo, Jason Heller, Thomas J.R. Hughes, Martin Ruess, and Dominik Schillinger. Variationally consistent isogeometric analysis of trimmed thin shells at finite deformations, based on the STEP exchange format. *Computer Methods in Applied Mechanics and Engineering*, März 2018. doi:10.1016/j.cma.2018.02.027.

[56] T. Bog, N. Zander, S. Kollmannsberger, and E. Rank. Weak imposition of frictionless contact constraints on automatically recovered high-order, embedded interfaces using the finite cell method. *Computational Mechanics*, Aug 2017. doi:10.1007/s00466-017-1464-6.

[57] L. Demkowicz. *Computing with hp-adaptive finite elements: one and two dimensional elliptic and Maxwell problems*, volume 1. Chapman & Hall / CRC Applied Mathematics & Nonlinear Science, 2006.

[58] L. Demkowicz, J. Kurtz, D. Pardo, M. Paszyński, W. Rachowicz, and A. Zdunek. *Computing with hp-adaptive finite elements: three dimensional elliptic and Maxwell problems with applications*, volume 2. Chapman & Hall / CRC Applied Mathematics & Nonlinear Science, 2007.

[59] E. Rank. Adaptive remeshing and h-p domain decomposition. *Computer Methods in Applied Mechanics and Engineering*, 101:299–313, 1992.

[60] Nils Zander, Tino Bog, Mohamed Elhaddad, Felix Frischmann, Stefan Kollmannsberger, and Ernst Rank. The multi-level hp-method for three-dimensional problems: Dynamically changing high-order mesh refinement with arbitrary hanging nodes. *Computer Methods in Applied Mechanics and Engineering*, 310:252–277, Oktober 2016. doi:10.1016/j.cma.2016.07.007.

[61] D. Schillinger, S. Kollmannsberger, R. Mundani, and E. Rank. The Finite Cell Method for Geometrically Nonlinear Problems of Solid Mechanics. In *The World Congress on Computational Mechanics*. Sidney, Australia, 2010. doi:10.1088/1757-899X/10/1/012170.

[62] 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*, 55(3):499–517, 2015. doi:10.1007/s00466-014-1118-x.

[63] Nils Zander, Martin Ruess, Tino Bog, Stefan Kollmannsberger, and Ernst Rank. Multi-level *hp*-adaptivity for cohesive fracture modeling. *International Journal for Numerical Methods in Engineering*, 2016. doi:10.1002/nme.5340.

[64] S. Kollmannsberger, A. Özcan, M. Carraturo, N. Zander, and E. Rank. A hierarchical computational model for moving thermal loads and phase changes with applications to selective laser melting. *Computers & Mathematics with Applications*, 75(5):1483–1497, 2018. doi:https://doi.org/10.1016/j.camwa.2017.11.014.

[65] Paolo Di Stolfo, Andreas Schröder, Nils Zander, and Stefan Kollmannsberger. An easy treatment of hanging nodes in hp-finite elements. *Finite Elements in Analysis and Design*, 121:101–117, November 2016. doi:10.1016/j.finel.2016.07.001.

[66] D. D'Angella, S. Kollmannsberger, E. Rank, and A. Reali. Multi-level bézier extraction for hierarchical local refinement of isogeometric analysis. *Computer Methods in Applied Mechanics and Engineering*, 328:147–174, 2018. doi:10.1016/j.cma.2017.08.017.

[67] A.-V. Vuong, C. Giannelli, B. Jüttler, and B. Simeon. A hierarchical approach to adaptive local refinement in isogeometric analysis. *Computer Methods in Applied Mechanics and Engineering*, 200(49-52):3554–3567, Dezember 2011. doi:10.1016/j.cma.2011.09.004.

[68] M. Ainsworth and J.T. Oden. *A posteriori error estimation in finite element analysis*. John Wiley & Sons, 2000.

[69] D. D'Angella, N. Zander, S. Kollmannsberger, F. Frischmann, A. Schröder, E. Rank, and A. Reali. Multi-Level hp-Adaptivity and Explicit Error Estimation. *Advanced Modeling and Simulation in Engineering Sciences*, 3(1):18, 2016. doi:10.1186/s40323-016-0085-5.

[70] V. Nübel, A. Shadavakhsh, M. Elhaddad, N. Zander, and S. Kollmannsberger. Dynamic analysis of highly loaded components discretized by fictitious domain methods. In *ECCOMAS Congress 2016*, Jun 2016. URL: http://congress.cimne.com/iacm-eccomas2014/admin/files/fileabstract/a2221.pdf.

[71] M. Elhaddad, N. Zander, S. Kollmannsberger, A. Shadavakhsh, V. Nübel, and E. Rank. Finite cell method: high order structural dynamics for complex geometries. *International Journal for Structural Stability and Dynamics*, 15:1–28, 2015. doi:10.1142/S0219455415400180.

[72] Mehran Monavari. The finite cell method for the analysis of die casts based on computed tomography images. Master's thesis, Technische Universität München, 2011.

[73] Z. Yang, S. Kollmannsberger, A. Düster, M. Ruess, E. Garcia, R. Burgkart, and E. Rank. Non-standard bone simulation: interactive numerical analysis by computational steering. *Computing and Visualization in Science*, 14(5):207–216, 2012. doi:10.1007/s00791-012-0175-y.

[74] John Jomo, Nils Zander, Mohamed Elhaddad, Ali Imran Özcan, Stefan Kollmannsberger, R.-P. Mundani, and Ernst Rank. Parallelization of the multi-level hp-adaptive finite cell method. *Computers and Mathematics with Applications*, 74(1):126–142, 2017. doi:doi.org/10.1016/j.camwa.2017.01.004.

[75] J.N. Jomo, F. de Prenter, M. Elhaddad, D. D'Angella, C.V. Verhoosel, S. Kollmannsberger, J.S. Kirschke, V. Nübel, E.H. van Brummelen, and E. Rank. Robust and parallel scalable iterative solutions for large-scale finite cell analyses. *Finite Elements in Analysis and Design*, 163:14–30, Oktober 2019. doi:10.1016/j.finel.2019.01.009.

[76] Martin Ruess, David Tal, Nir Trabelsi, Zohar Yosibash, and Ernst Rank. The finite cell method for bone simulations: Verification and validation. *Biomechanics and modeling in mechanobiology*, 11(3-4):425–37, März 2012. doi:10.1007/s10237-011-0322-2.

[77] Zhengxiong Yang, Martin Ruess, Stefan Kollmannsberger, Alexander Düster, and Ernst Rank. An efficient integration technique for the voxel-based finite cell method. *International Journal for Numerical Methods in Engineering*, 91(5):457–471, August 2012. doi:10.1002/nme.4269.

[78] Zhengxiong Yang, Stefan Kollmannsberger, Alexander Düster, Martin Ruess, Eduardo Grande Garcia, Rainer Burgkart, and Ernst Rank. Non-standard bone simulation: Interactive numerical analysis by computational steering. *Computing and Visualization in Science*, 14(5):207–216, April 2012. doi:10.1007/s00791-012-0175-y.

[79] Mohamed Elhaddad, Nils Zander, Tino Bog, László Kudela, Stefan Kollmannsberger, Jan S. Kirschke, Thomas Baum, Martin Ruess, and Ernst Rank. Multi-level hp-finite cell method for embedded interface problems with application in biomechanics. *International Journal for Numerical Methods in Biomedical Engineering*, 34(4):e2951, 2018. doi:10.1002/cnm.2951.

[80] Christopher B. Williams, Farrokh Mistree, and David W. Rosen. A Functional Classification Framework for the Conceptual Design of Additive Manufacturing Technologies. *Journal of Mechanical Design*, 133(12):121002, 2011. doi:10.1115/1.4005231.

[81] Kianian Babak. Wohlers Report 2017: 3D Printing and Additive Manufacturing State of the Industry, Annual Worldwide Progress Report : Chapters titles: The Middle East, and other countries. Technical report, Wohlers Associates, Inc., 2017.

[82] Stefan Kollmannsberger, Massimo Carraturo, Alessandro Reali, and Ferdinando Auricchio. Accurate Prediction of Melt Pool Shapes in Laser Powder Bed Fusion by the Non-Linear Temperature Equation Including Phase Changes. *Integrating Materials and Manufacturing Innovation*, 8(2):167–177, 2019. doi:10.1007/s40192-019-00132-9.

[83] AM Bench Benchmark Challenge CHAL-AMB2018-02-MP. https://www.nist.gov/ambench/amb2018-02-description.

[84] Songzhe Xu, Fei Xu, Aditya Kommajosula, Ming-Chen Hsu, and Baskar Ganapathysubramanian. Immersogeometric analysis of moving objects in incompressible flows. *Computers & Fluids*, 189:24–33, 2019. doi:https://doi.org/10.1016/j.compfluid.2019.05.018.

[85] David Kamensky, Ming-Chen Hsu, Dominik Schillinger, John A. Evans, Ankush Aggarwal, Yuri Bazilevs, Michael S. Sacks, and Thomas 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, Februar 2015. 00015. doi:10.1016/j.cma.2014.10.040.

[86] Erik Burman, Susanne Claus, Peter Hansbo, Mats G. Larson, and André Massing. CutFEM: Discretizing geometry and partial differential equations. *International Journal for Numerical Methods in Engineering*, 104(7):472–501, November 2015. doi:10.1002/nme.4823.

[87] Thomas Oberbichler, Anna M. Bauer, Ann-Kathrin Goldbach, Roland Wüchner, and Kai-Uwe Bletzinger. CAD-integrierte Analyse im Entwurfsprozess. *Bautechnik*, 96(5):400–408, 2019. arXiv:https://onlinelibrary.wiley.com/doi/pdf/10.1002/bate.201800105, doi:10.1002/bate.201800105.

[88] B. Philipp, M. Breitenberger, I. D'Auria, R. Wüchner, and K.-U. Bletzinger. Integrated design and analysis of structural membranes using the Isogeometric B-Rep Analysis. *Computer Methods in Applied Mechanics and Engineering*, 303:312–340, Mai 2016. doi:10.1016/j.cma.2016.02.003.

[89] Frits de Prenter. *Preconditioned Iterative Solution Techniques for Immersed Finite Element Methods: With Applications in Immersed Isogeometric Analysis for Solid and Fluid Mechanics*. PhD thesis, Eindhoven University of Technology, Eindhoven, Juni 2019.

[90] A. Taghipour, Parvizian, J., S. Heinze, and A. Düster. The finite cell method for nearly incompressible finite strain plasticity problems with complex geometries. *Computers & Mathematics with Applications*, page accepted, 2018.

[91] S. Kollmannsberger, D. D'Angella, E. Rank, W. Garhuom, S. Hubrich, A. Düster, P. Di Stolfo, and A. Schröder. Spline- and hp-basis functions of higher differentiability in the finite cell method. *GAMM-Mitteilungen*, accepted for publication:x, August 2019.