Available online at www.sciencedirect.com
ScienceDirect
Comput. Methods Appl. Mech. Engrg. 328 (2018) 75–98
Computer methods in applied mechanics and engineering
www.elsevier.com/locate/cma

Higher-order conformal decomposition FEM (CDFEM)

T.P. Fries

Institute of Structural Analysis, Graz University of Technology, Lessingstr. 25, 8010 Graz, Austria

Received 22 May 2017; received in revised form 28 August 2017; accepted 29 August 2017
Available online 7 September 2017

Highlights

- The proposed CDFEM is a hybrid between classical p-FEM and fictitious domain methods.
- The CDFEM is useful for embedded domains and interface problems.
- Adaptive, automatic, higher-order mesh generation of 2D and 3D domains defined by level-set data.
- Manipulation of background mesh ensures regularity of the generated, conforming elements.
- Optimal convergence rates are achieved.

Abstract

A higher-order accurate finite element method is proposed which uses automatically generated meshes based on implicit level-set data for the description of boundaries and interfaces in two and three dimensions. The method is an alternative for fictitious domain and extended finite element methods. The domain of interest is immersed in a background mesh composed by higher-order elements. The zero-level sets are identified and meshed followed by a decomposition of the cut background elements into conforming sub-elements. Adaptivity is a crucial ingredient of the method to guarantee the success of the mesh generation. It ensures the successful decomposition of cut elements and enables improved geometry descriptions and approximations. It is confirmed that higher-order accurate results with optimal convergence rates are achieved with the proposed conformal decomposition finite element method (CDFEM).

Keywords: Higher-order FEM; Level-set method; Fictitious domain method; Embedded domain method; XFEM; Interface capturing

1. Introduction

The p-version of the finite element method (p-FEM) enables a higher-order accurate and efficient approximation of boundary value problems (BVPs) in engineering, natural sciences, and related fields [1–5]. Two crucial requirements are needed for the successful application of the p-FEM: (i) The geometry must be accurately represented by a mesh composed of higher-order elements and (ii) the solution of the BVP should be sufficiently smooth. Both requirements are not easily met. For (i), *curved* boundaries and interfaces in the domain of interest may render the mesh generation

E-mail address: fries@tugraz.at.
URL: http://www.ifb.tugraz.at.

http://dx.doi.org/10.1016/j.cma.2017.08.046
0045-7825/

difficult, in particular in three dimensions and with elements of higher orders. Even more so when frequent mesh manipulations are desired, for instance, in the context of moving interfaces (interface tracking) or mesh refinements in adaptivity and convergence studies. The original geometry is often generated based on Computer Aided Geometric Design (CAGD or CAD) and the interplay with the analysis tool, i.e., the p-FEM, is not easily established and hardly automated. Concerning (ii), the smoothness of the involved fields in the BVP, it is noted that discontinuities (e.g., in the material parameters) and singularities (e.g., in the stress field of a structure) are frequently present. For the successful application of the p-FEM in these cases, it is again crucial to provide suitable meshes, i.e., those which conform to the discontinuities and are refined at the singularities. It is thus seen that a lot of effort is associated to generating higher-order accurate meshes as properties of the geometry and the approximated solutions must both be considered.

Herein, the focus is on the *automatic*, higher-order accurate generation of conforming meshes based on implicitly defined geometries. The domain of interest is completely immersed in a background mesh. The boundary of the domain and interfaces therein, for example, between different materials, are defined by (several) level-set functions [6–8]. For each level-set function, the elements cut by the zero-level set are decomposed into conforming, higher-order sub-elements. Therefore, the zero-level set is first identified and meshed by interface elements (reconstruction) and then customized mappings generate the sub-elements (decomposition). This follows previous works of the author in [9–11] where the resulting meshes are used in the context of integration and interpolation in implicitly defined domains. However, in elements where the decomposition fails, e.g., due to very complex level-set data, (isolated) recursive refinements were suggested and hanging nodes are a natural consequence. Herein, we wish to use the generated meshes in the context of approximating BVPs and hanging nodes shall be avoided. The quality of the generated sub-elements becomes an important issue in this context. Following [12], node manipulations in the background mesh are suggested to ensure suitable, shape-regular elements. One may also possibly use stabilizations similar to those suggested in [14–17].

Adaptive refinements of the background mesh are suggested in order to (i) refine elements where the decomposition failed, (ii) improve the geometry description driven by the curvature of the involved level-set functions near the zero-level sets, (iii) improve the approximation of the BVP, for example, based on error indicators. Because “good” meshes must consider both, the geometry *and* the involved (sought) fields of the BVP, adaptivity is a natural ingredient for *automatic* mesh generation without any user intervention. Hence, the suggested procedure follows the isogeometric paradigm [18,19] to fully integrate design and analysis, however, for implicit geometries rather than based on NURBS as in CAGD.

The fully automatic generation of meshes based on implicit level-set data is gaining increasing attention. We emphasize the work of [20] in a low-order context for moving interfaces which coined the name CDFEM. A higher-order extension of this work in two dimensions is found in [21] without adaptive refinements and measures to avoid ill-shaped elements. This is the first work where the CDFEM is extended to higher-order consistently in two and three dimensions, including adaptivity and node manipulations to ensure the regularity of the resulting elements. The resulting method is stable and efficient.

The decomposition of elements is frequently employed in the context of “fictitious domain methods” (FDMs) such as the unfitted or cut finite element method [14–17], finite cell method [22–26], Cartesian grid method [27,28], immersed interface method [29], virtual boundary method [30], embedded domain method [31,32] etc. The important difference between the CDFEM and FDMs is that the first uses the shape functions of the decomposed elements in the conforming mesh as the approximation basis whereas the second employs the shape functions of the original background mesh and uses the sub-elements for integration purposes only. Integration in cut elements using element decompositions is suggested in [33–35] using *polygonal* sub-cells together with recursive refinements. Curved sub-cells based on higher-order elements are, e.g., used in [9,11,36,37] and typically lead to much less integration points. The integration schemes based on element decompositions are also frequently employed in the context of the extended or generalized finite element methods (XFEM/GFEM), see e.g., [38–40] for the XFEM and [41,42] for the GFEM. They consider for inner-element jumps and kinks by adding enrichment functions based on the partition of unity concept [43–45]. Again, in these methods the decomposed sub-elements are only used as integration cells without using the implied shape functions for the approximation of the BVP. We also mention the approach in [46] providing a concept for higher-order accurate integration in the presence of one level-set function by deforming the mesh in the vicinity of the zero-level set.

It is emphasized that the proposed higher-order accurate CDFEM may be seen as an alternative for FDMs where *boundaries* are defined implicitly *and* the XFEM where *interfaces* are defined implicitly. The numerical results show