Finite Elements in Analysis and Design 113 (2016) 1–13

Contents lists available at ScienceDirect

Finite Elements in Analysis and Design

journal homepage: www.elsevier.com/locate/finel

Full Length Article

A new methodology for element partition and integration procedures for XFEM

M.T. Cao-Rial [a,d], C. Moreno [b], P. Quintela [c,d,*]

[a] Department of Mathematics, Universidade da Coruña, A Coruña 15001, Spain
[b] Department of Statistics, OR and Numerical Analysis, Universidad Nacional de Educación a Distancia, Madrid 28040, Spain
[c] Department of Applied Mathematics, Campus Vida, Universidade de Santiago de Compostela, Santiago de Compostela 15782, Spain
[d] ITMATI (Technological Institute for Industrial Mathematics), Campus Vida. Santiago de Compostela 15782, Spain

ARTICLE INFO

Article history:
Received 16 June 2015
Received in revised form
18 December 2015
Accepted 21 December 2015
Available online 14 January 2016

Keywords:
XFEM
Level sets
Barycentric coordinates
Quadrature
Interface
Enrichment
Element partitioning

ABSTRACT

An overview of the particularities of the extended finite element method implementation in contrast with the classical FEM is presented. The most relevant difficulty lies in the integration over elements containing jumps or singularities, since a classical quadrature rule cannot be applied. We present an algorithm which, avoiding a casuistic analysis, automatically partitions an enriched element and constructs a new quadrature formula for those elements that preserves the integration order from the original one.

1. Introduction

When the solution of a problem involves jumps or singularities, a solution obtained by using the standard finite element method might not accurately approximate the actual solution, or the method might be very expensive, computationally speaking, either due to the use of very refined meshes or the need to remesh every now and then. We can find a large number of examples in real world with non-smooth solutions, both in solid and fluid mechanics. Models for cracks, shear bands, dislocations, inclusions or voids are some examples in solid mechanics and shocks, bubbles, or boundary layers in fluid mechanics.

The eXtended Finite Element Method (XFEM), developed by Moës, Dolbow and Belytschko [7,12], introduces a new methodology that, by means of a local enrichment, not only allows a more accurate approximation of non-smooth solutions avoiding the need to remesh and/or refine, which is an obvious computational advantage, but more importantly, it enables discontinuities across the interface. The local enrichment is achieved by adding new shape functions to the base of the approximation space. These functions are supposed to replicate the singular behaviour of the solution. There are some references that deal with the implementation of the XFEM like [2], where an open source architecture for the method is presented, including several functions in an Object-Oriented framework, [14] where a numerical implementation of the XFEM to analyze crack propagation in a structure under dynamic loading is presented, [17] where the implementation for modelling 2-d cracks in isotropic and bimaterial media is described or [9] where the implementation of the XFEM into the commercial FE software Abaqus is discussed. However, despite the extensive literature about XFEM, to the authors knowledge, all element partition procedures found there use a casuistic procedure to find out which element edges are actually cut [16,2]. In this work, we will present a technique that allows the partition of any element, avoiding a casuistic analysis, by means of an array related to a barycentric representation of the interface [5]. This technique is presented for two and three dimensional elements.

In this paper we will study in detail the implementation of the XFEM in a general framework, showing its special features and difficulties. The main contribution of this work is the introduction of partition matrices to describe the interface position with respect to the mesh elements, which are obtained by solving simple linear systems (as many as the number of faces and edges of one element) of at most three unknowns. These matrices are computed

* Corresponding author at: Department of Applied Mathematics, Campus Vida, Universidade de Santiago de Compostela Santiago de Compostela 15782, Spain.
E-mail addresses: teresa.cao@udc.es (M.T. Cao-Rial), cmoreno@ccia.uned.es (C. Moreno), peregrina.quintela@usc.es (P. Quintela).

http://dx.doi.org/10.1016/j.finel.2015.12.012
0168-874X/

only once if the interface does not evolve, and even with an evolving interface, like time dependent or inverse problems, this procedure is much less computationally expensive than remeshing. The proposed methodology is compatible with level sets description of the interface or a description based on a polygonal given by its vertex coordinates. It is also remarkable that these partition matrices will make automatic the process of integrating over an enriched element, being unnecessary to determine precisely how the element is cut by the interface. Indeed, they allow us to replicate any quadrature formula on the subelements of the virtual partition, keeping the original order of integration. Besides we include in this paper how these partition matrices also allow the projection of discontinuous functions to the XFEM space.

The outline of the paper is the following. First, we will review the basic concepts of the XFEM and the implications of the presence of an interface on the mesh. In particular, in Section 2 we will focus on the classification of elements and nodes depending on the kind of shape functions associated to them while in Section 3 we set a general framework where we introduce the XFEM approximation space. Since the main feature of the XFEM is that the shape functions might be discontinuous or have singularities on some elements, which need to be split in order to apply any classical quadrature rule, our next goal will be the integration on enriched elements. Therefore, in Section 4 we will present several theoretical results that prove some interesting properties of the quadrature rules, in particular, given any partition of a standard element, they will allow us to adapt any quadrature rule automatically to every subelement. Sections 5 and 6 are devoted to the obtention of a general technique to split a reference element automatically without any additional computation. The key steps are two, the barycentric description of the interface, and the use of such description to construct the embedding matrices that will define the subelements. In order to relate the procedure presented on this paper with the actual implementation, a flow chart of a generic problem implementation will be shown, and each section will be related to some step on the chart. In Section 7 we will show the results obtained applying the techniques presented in this work to benchmark problems, consisting of a cracked elastic plate subjected to fracture modes I and II, corresponding to opening and in-plane shear forces and, finally, in Section 8 we will review the main contributions of this work. Just for the sake of completeness we will also include two annexes where the obtention of the barycentric description of an interface given as a polygonal line is presented as well as a list of notations used throughout this paper.

2. Sets of mesh nodes and elements

In this section we will introduce the notation used to refer to the different elements and nodes in a mesh and some characteristic subsets where different integration techniques are needed.

Let us consider a domain $\Omega \subset \mathbb{R}^n$, $n=2$ or 3, discretized by n_{he} elements. Although the procedure here described can be applied also to other elements and higher order functions with minor changes, for the sake of simplicity in the exposition, throughout this paper we will consider that the elements are triangles or tetrahedra and the standard shape functions considered are linear. We denote by

$$\mathcal{T}_h = \{T^k : k = 1, \ldots, n_e\},$$

the set of all elements in the mesh, h being the discretizacion parameter of the mesh, and by

$$\mathcal{P}_h = \{\boldsymbol{X}_I : I = 1, \ldots, n_n\},$$

the set of all node coordinates of $\mathcal{T}_h$, being n_n the number of mesh nodes. In the following, in order to simplify the notation, the superindex k will be omitted when there is no room for confusion, and therefore, T will denote a generic element of $\mathcal{T}_h$.

We consider also that there is an interface, $\mathcal{S}$, on the domain, not necessarily aligned with the elements edges. The interface $\mathcal{S}$ is assumed to be an oriented curve or surface. In the three dimensional case, the normal vector defining the orientation of the surface $\mathcal{S}$ will be denoted by $\boldsymbol{n}$. When $n=2$, we will consider $\boldsymbol{n}$ as the normal vector to the oriented curve pointing to its right while walking along it in the direction of the positive orientation (see Fig. 1). The domain can therefore be split into two domains, Ω^+, whose outward normal is $\boldsymbol{n}$ and Ω^-, its complement, by extending the interface with enough regularity if necessary as can be seen in Fig. 2 when $n=2$.

Usually, the presence of an interface in the domain leads to discontinuous fields, whether on displacements, pressure or its derivatives. Furthermore, if the interface is open, its ending points might introduce singularities on the solution. That is why several kinds of enrichment functions might be taken into account and the classification of nodes and elements will depend on which enrichment function is used on them.

For the purpose of this paper we will consider two kinds of enrichment functions: H which will be a discontinuous function through $\mathcal{S}$, and a set, Ψ, of n_ψ functions ψ_l, $l=1,\ldots,n_\psi$, whose properties will depend on the local behaviour of the solution. Clearly, how many kinds are used or the number of functions ψ may vary depending on the kind of problem studied. The generalization of the methodology presented here to other enrichment functions carries no difficulty.

2.1. Level sets representation

Let us assume that the interface is given by level sets. If it is a closed interface, one level set function is enough to represent it. If it is an open interface two or even three level set functions might be necessary to describe it, as shown in Fig. 3.

In the following we will consider that the interface is given by two level set functions ϕ and φ (see Fig. 4), since the closed interface case is a particular case and when more than two level set functions are needed the generalization is straightforward. Therefore, the zero level set of the scalar function ϕ corresponds to the interface when $\varphi < 0$ and the points $\boldsymbol{x} \in \Omega \subset \mathbb{R}^n$ verifying $\phi(\boldsymbol{x}) = 0$ and $\varphi(\boldsymbol{x}) = 0$ represent the interface ending points.

In a point lying to the left of the interface ϕ will be assigned a positive value, whereas on the right side it will take negative values, this is

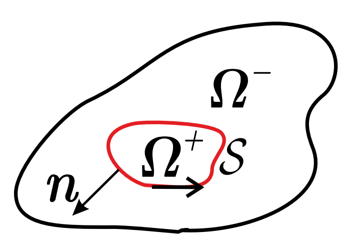

Fig. 1. Domain decomposition for a closed interface.

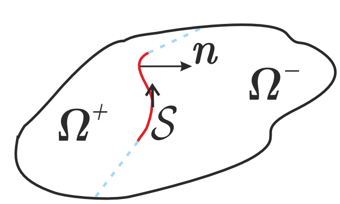

Fig. 2. Domain decomposition when the interface is an open path.