Engineering Fracture Mechanics 153 (2016) 171–174

Contents lists available at ScienceDirect

Engineering Fracture Mechanics

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

Technical Note

# On interface element insertion into three-dimensional meshes

Timothy J. Truster *

*Department of Civil and Environmental Engineering, University of Tennessee, Knoxville, 318 John D. Tickle Engineering Building, Knoxville, TN 37921, United States*

ARTICLE INFO

*Article history:*
Received 17 December 2015
Accepted 19 December 2015
Available online 29 December 2015

*Keywords:*
Interface elements
Cohesive zone models
Discontinuous Galerkin method
Open source code

ABSTRACT

An algorithm is summarized for inserting zero-thickness interface elements or "couplers" into finite element meshes in two and three dimensions. Using only element connectivity, couplers are inserted according to regions within the analysis domain, a geometrically intuitive means to designate their locations. A wide class of volume elements and interface couplers are treated. A three-dimensional test case verifies that the algorithm generates meshes passing the patch test.



## 1. Introduction

Discontinuous finite element formulations have been applied to a variety of fracture mechanics problems, including fragmentation [1], delamination [2], and grain boundary cracking [3]. Additionally, solutions fields for solid mechanics problems containing sharp gradients can be efficiently modeled using the Discontinuous Galerkin (DG) method [4,5]. These methods are commonly implemented using so-called zero-thickness interface finite elements, which involves duplicating the nodes along the potential surface of discontinuity prior to conducting the finite element simulation. Unfortunately, since mesh generators do not currently exist for zero-thickness elements, the user must create the modified mesh connectivity, which can be highly non-trivial for complex meshes in three dimensions. A few algorithms have been proposed for inserting interface elements either one at a time during fragmentation analyses [6] or along restricted types of interfaces within the mesh [7]. Nonetheless, an intuitive method is desirable for the user to easily designate the location of the interface elements.

A general-purpose algorithm was recently developed in [8] for inserting zero-thickness interface elements, termed therein as "couplers", into conforming two and three dimensional meshes along specified regions. The node duplication is accomplished by identifying sectors of elements attached to individuals nodes, which ensures that the desired degree of continuity and discontinuity is maintained in vicinity of the interface. With minimal user input, couplers are inserted along specific interfaces or within specific regions. An open-source version of the algorithm in MATLAB© is provided at https://bitbucket.org/trusterresearchgroup/deiprogram. A three dimensional example on a polycrystalline geometry confirms that the algorithm generates analysis suitable meshes.

## 2. Review of coupler insertion algorithm

We briefly summarize the algorithm from [8], which applies to conforming *domains* of finite *elements* in two (2D) or three (3D) dimensional space. Each of the topological entities *nodes*, *elements*, *edges*, and *faces* are defined in the usual sense of the

* Tel.: +1 (865) 974 1913; fax: +1 (865) 974 2669.
*E-mail address:* ttruster@utk.edu

http://dx.doi.org/10.1016/j.engfracmech.2015.12.019
0013-7944/

finite element method. The term *facet* refers either to an *edge* in 2D or a *face* in 3D. Additionally, a *region* is defined as a unique, contiguous set of *elements* within the *domain*, which in general may form nonconvex subdomains. The *facets* of all the *elements* in the *domain* are separated into three disjoint sets: those which lie on the *domain* boundary, those which lie between *elements* of two different *regions* (termed as *interfaces*), and those which lie between *elements* of the same *region* (termed as *intrafaces*). Finally, a *coupler* is defined as a topological unit consisting of duplicate *nodes* from exactly two *elements* that are adjacent across either an *interface* or *intraface*. Such *couplers*, otherwise known as "zero-thickness elements" or "interface elements", are used for modeling PDEs with discontinuous formulations. For example, intrinsic cohesive zone models can be introduced as *interface couplers* between the fiber and matrix constituents of composite materials, thereby capturing the initiation and progression of fracture at the interface through a traction–separation relation. The treatment of the theoretical and computational aspects of such formulations is beyond the scope of this work; the reader is referred to [1,2,4,5,7] for aspects of theory and implementation.

According to [8], the set of *couplers* to insert and the set of *nodes* to duplicate within an initially conforming mesh are determined based upon the topology of *regions* in the mesh. Crucially, the method relies upon the concept of *sectors* of *elements* surrounding a focus *node*. For a given *node*, a *sector* consists of *elements* that are connected by *facets* which are not designated for *couplers*, indicating a "solid" portion of the mesh. A *sector* may contain *elements* from multiple *regions*, since the user may desire for *couplers* to be inserted only along selected *region interfaces*. In Fig. 1(a), which is reproduced from [8], the *sector* is indicated by a single shading pattern. Therefore, the *node* duplication phase of the algorithm must ensure that a single nodal coefficient is assigned to all *elements* in a common *sector*, so that the solution field remains locally continuous. Meanwhile, *elements* in distinct *sectors* are assigned separate duplicates of the *node* so that the resulting interpolation functions include the desired level of discontinuity. We remark that use of *sectors* rather than *regions* for assigning nodal duplicates is critical when *couplers* are inserted only along selected *region interfaces*. Otherwise, the definition of *sectors* becomes equivalent to *regions* when all *interface facets* are designated for *couplers*. This subtle point is elucidated through Fig. 1(b), where an improper degree of discontinuity results when *couplers* are not inserted along the *interface* between *regions B* and *C* while duplicates of *node 5* are still assigned to each *region*.

The *coupler* insertion algorithm consists of six phases that are listed below. Further details on the algorithm phases, as well as pseudo-code, are contained in [8].

1. Construct the set of *elements* attached to each *node*.
2. Categorize all *facets* in the mesh as *boundary*, *interface*, or *intraface*.
3. According to *regions*, designate all *interface facets* where *couplers* are to be inserted.
4. According to *sectors*, duplicate *nodes* along *interfaces* and update element connectivity.
5. Duplicate *nodes* for all *intraface facets* where *couplers* are to be inserted.
6. Construct the connectivity of *nodes* to *couplers* according to specified templates.

Templates for triangular and quadrilateral *elements* in 2D and tetrahedral, wedge, and hexahedral *elements* in 3D along with their *couplers* are provided in [8]. A realization of the proposed algorithm written in MATLAB© is provided at https://bitbucket.org/trusterresearchgroup/deiprogram; the code has also been successfully tested in the GNU interpreted language Octave. Several 2D and 3D examples are provided with the program, including the 2D mesh shown in Fig. 1 as well as the 3D mesh described in Section 3.

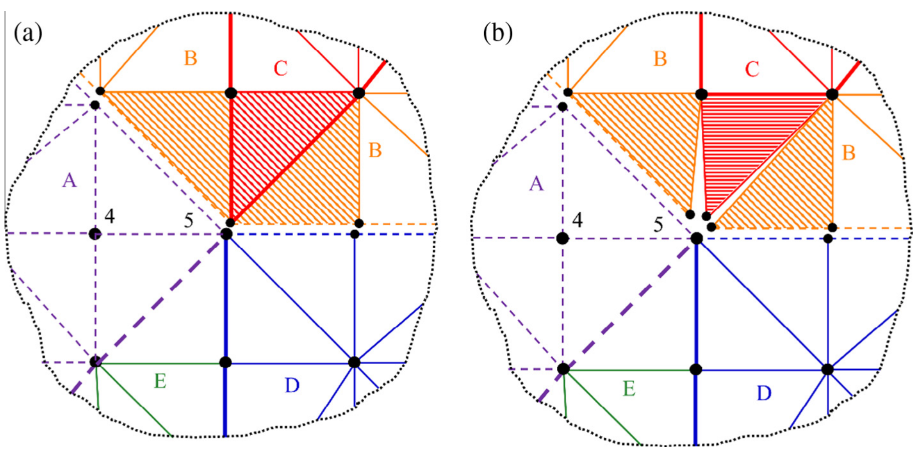


**Fig. 1.** Node duplication: (a) correct continuity obtained by using sectors; (b) incorrect discontinuity obtained by using regions.