Engineering Fracture Mechanics 128 (2014) 37–68

Contents lists available at ScienceDirect

Engineering Fracture Mechanics

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

# Discontinuous Galerkin/extrinsic cohesive zone modeling: Implementation caveats and applications in computational fracture mechanics

Vinh Phu Nguyen

*School of Engineering, Institute of Mechanics and Advanced Materials, Cardiff University, Queen's Buildings, The Parade, Cardiff CF24 3AA, United Kingdom*

ARTICLE INFO

*Article history:*
Received 28 January 2014
Received in revised form 6 July 2014
Accepted 7 July 2014
Available online 18 July 2014

*Keywords:*
Cohesive zone model (CZM)
Discontinuous Galerkin
Cohesive interface elements
Intrinsic/extrinsic traction separation laws (TSLs)
Fracture mechanics

ABSTRACT

Implementation of the hybrid discontinuous Galerkin/cohesive zone model method, which is a (volumetric) locking free single field yet highly scalable method for nonlinear fracture mechanics problems, is presented. In this method cohesive interface elements are placed at element surfaces prior to the simulation of which artificial compliance is removed by a discontinuous Galerkin formulation. The formulation is switched to a standard extrinsic cohesive crack model upon satisfaction of a failure criterion. We provide details on the computation of the internal force vector and the tangent stiffness matrix. Examples that consist of (in) compressible elasticity, microcracking of fiber reinforced composite materials and dynamic fracture are provided. This paper is addressed to researchers who would like to have a quick working implementation of the discontinuous Galerkin/cohesive zone model method.



## 1. Introduction

Cohesive zone models (CZMs) which were pioneered in [1,2] is a continuation of linear elastic fracture mechanics with which the unrealistic stress singularity ahead the crack tip is avoided. Application of CZMs as fracture models used in the context of the finite element method however appeared substantially later in [3]. From a numerical point of view, CZMs have been incorporated in a finite element (FE) context using zero-thickness interface elements, elements with embedded discontinuities [4–6] and elements with discontinuous enrichment via the extended/generalized finite element method (XFEM/GFEM) [7,8]. A comparative study on the modeling of quasi-static discontinuous fracture using these techniques was given in [9]; of dynamic fracture in [10] and a review of computational methods for fracture in quasi-brittle solids has recently been reported in [11]. It should be emphasized that the term "cohesive elements" or "cohesive zone elements" usually used to refer to cohesive interface elements is misleading since elements with embedded cohesive cracks or XFEM with cohesive cracks are also cohesive elements. Therefore, in this paper, the name "cohesive interface elements" (in subsequent discussion interface elements are used for brevity) to indicate interface elements equipped with a cohesive law. A cohesive law provides a relation between the cohesive traction and the displacement jump across the crack surfaces.

Despite of the fact that XFEM/GFEM has been gaining more popularity within the fracture mechanics community, interface elements still constitute a very good candidate for modeling a number of fracture mechanics problems including interfacial fracture (delamination in composite laminates [12–16], failure of adhesive layers [17], failure of layered composites

*E-mail address:* nguyenpv@cardiff.ac.uk

http://dx.doi.org/10.1016/j.engfracmech.2014.07.003
0013-7944/© 2014 Elsevier Ltd. All rights reserved.

**Nomenclature**

| | |
|---|---|
| $\beta$ | fracture switch operator |
| $\boldsymbol{\epsilon}$ | strain tensor |
| $\boldsymbol{\sigma}$ | Cauchy stress tensor |
| $\gamma$ | average coefficient |
| $[\![\mathbf{u}]\!]$ | displacement jump vector |
| $\lambda,\ \mu$ | Lamé constants |
| $\theta$ | stabilization parameter |
| $\theta_{DG}$ | dG switch operator |
| $\mathbf{C}$ | constitutive tensor |
| $\mathbf{f}^{ext}$ | external force vector |
| $\mathbf{f}^{int}$ | internal force vector |
| $\mathbf{R}$ | rotation matrix |
| $\mathbf{T}$ | cohesive material tangent |
| $\mathbf{t}$ | cohesive traction |
| $\mathbf{u}$ | displacement field |
| $\{\cdot\}$ | average operator |
| $c_R$ | Rayleigh wave speed |
| $f_{12}$ | shear strength |
| $f_{2t}$ | tensile strength |
| $G_{Ic}$ | mode I fracture toughness |
| $G_{IIc}$ | mode II fracture toughness |
| $h_e$ | element size |
| $K$ | dummy stiffness |

such as fiber-metal laminates [18,19] matrix/interface debonding in fiber reinforced composites [20–22]) and problems with complex crack patterns (fragmentation, dynamic crack branching [23–28], failure of heterogeneous materials [29–34]). Cohesive interface elements possess the following advantages: (i) easy incorporation into any FE codes (the volumetric element formulation is unaltered),[1] (ii) robustness, (iii) straightforward parallelization and (iv) suitable for modeling pervasive fracture and fragmentation. Other computational methods that are capable of dealing with complex fracture mechanics problems include meshfree methods, e.g., [36–39], phase-field approaches [40,41], lattice models [42], and peridynamics see [43,44] among others.

Interface elements can be divided into two categories–intrinsic interface elements and extrinsic interface elements. The former are inserted into the volumetric mesh (along the element edges or faces) prior to the simulation and they employ the so-called intrinsic traction-separation laws (TSL) or cohesive laws [23]. The latter are adaptively added to the mesh, at places where a failure condition is met, during the analysis and they use extrinsic TSLs [25]. The intrinsic and extrinsic TSLs are graphically illustrated in Fig. 1. While parallelizing intrinsic interface element codes is quite straightforward the artificial compliance introduced by the elastic branch of intrinsic TSLs affects the propagation of elastic stress waves in the un-cracked continuum and causes spurious crack tip speed (the lift-off issue). The only remedy to reduce the effect of artificial compliance is to increase, if possible, the initial elastic slope of the TSL, which results in severe stable time step restrictions, which render the intrinsic approach unsuitable for explicit dynamic calculations, or in ill-conditioning of the tangent stiffness matrices in static or implicit dynamics analyses. Being absent in the un-cracked solid, extrinsic interface elements overcome naturally the artificial compliance and the lift-off issue. However, unless advanced data structure is used to store the FE discretization, parallelization of extrinsic interface element codes is not an easy task due to the change of the mesh topology as the cracks advance. Only recently such (topology-based) data structure was emerged [45,27] and a scalable parallel implementation was presented in [46]. Note that even a serial implementation of the extrinsic approach is not trivial especially in three dimensions.

In an attempt to remove the disadvantages of intrinsic interface elements a hybrid discontinuous Galerkin (dG) and extrinsic TSL approach was presented in [47] for two dimensional interfacial fracture problems. The interface elements are inserted prior to the analysis but in the elastic regime they are kept not to be separated, in a weak sense, by a dG formulation. When a failure criterion is met, the dG formulation is replaced by a (extrinsic) cohesive formulation. Recent developments of this method are given in [48,49,34,50] for three dimensional fracture/fragmentation problems and crack propagation of fiber reinforced composites with excellent scalability up to 4000 processors; and thin shell fracture [51]. The term "discontinuous Galerkin" seems to firstly appear in [52] and has its origin from the Nitsche's method [53]. We refer to Ref. [54] for a unified analysis of dG methods for elliptic problems. In a somewhat similar work a mixed interface element

[1] Based on the author's own experiences in implementing XFEM, [35], and cohesive interface elements it can be concluded without any doubt the latter is much more easier to implement.