ARTICLE IN PRESS

Finite Elements in Analysis and Design ∎ (∎∎∎∎) ∎∎∎–∎∎∎

Contents lists available at ScienceDirect

Finite Elements in Analysis and Design

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

Full Length Article

Semi-implicit representation of sharp features with level sets

H. Asadi Kalameh [a,b], O. Pierard [a,*], C. Friebel [a], E. Béchet [b]

[a] Morfeo Team, Cenaero, Rue des frères Wright, 29, B-6041 Gosselies, Belgium
[b] Department of Aerospace and Mechanical Engineering, Université de Liège, Quartier Polytech 1, Allée de la Découverte 9, B-4000 Liège, Belgium

ARTICLE INFO

Article history:
Received 28 January 2016
Received in revised form
15 April 2016
Accepted 17 April 2016

Keywords:
Level set method
Sharp feature
Implicit representation
Boolean operation
Finite elements

ABSTRACT

The present contribution enriches the nowadays "classical" level set implicit representation of geometries with topological information in order to correctly represent sharp features. For this, sharp features are classified according to their positions within elements of the level set support. Based on this additional information, sub-elements and interface-mesh used in a finite element context for integration and application of boundary conditions are modified to match exactly to the sharp features. In order to analyze evolving geometries, Boolean operations on these semi-implicit representations are derived so that the minimal additional information to represent correctly the new geometry is stored. This approach has been successfully applied to complex two-dimensional geometries. It computes in a robust way numerous Boolean operations and guarantees the precision and the convergence rate of the numerical simulations.

1. Introduction

The level set method (LSM) was originally introduced by Osher and Sethian [1–3] as a robust technique to represent implicitly the evolution of interfaces which have a smooth geometry in two or three dimensions. The representation of the interface, or more generally any boundary, is obtained by the iso-zero of the level set function, classically a distance function to the boundary. This function is defined on a grid so that it is suitable for using it within a finite element context. A major advantage is that the simulation mesh does not need to match the boundary anymore. In case of a moving interface in the normal direction, a Hamilton–Jacobi equation – also called the level set equation – is solved to track the interface.

This method has been successfully employed in a wide variety of applications such as the solidification process [4], crystal growth [5], crack representation [6], image processing [7] or multi-phases flows [8,9]. Another key topic for which implicit representation with level set is helpful is topology optimization [10–12].

Implicit representation of smooth interfaces is particularly efficient with the level set method. However, when the boundary has small curvature radius, sharp features like corners or small items with respect to the characteristic length of the grid, a smoothing effect is observed. In some cases, as for the implicit representation of a CAD model, this might be unacceptable. Several improvements have been proposed over the years in order to circumvent these limitations. A first approach is to use higher-order level sets instead of the classical first-order interpolation [13]. Accurate integration requires a particular attention [14]. Such an approach has been successfully used in several applications, including magneto-mechanical problems [15]. Another approach is to dissociate the computation mesh from the grid on which the level set is defined as adopted by Legrain et al. [16]. Typically, a finer grid might be used in regions where sharp geometrical features are located. This latter approach has been used by Legrain et al. [17] to represent implicitly CAD thin structures. Even if these two approaches improve the implicit representation by reducing the geometrical error, both are unable to represent exactly sharp features.

Using several level sets to represent accurately sharp features as corners or edges has been set up by Moumnassi et al. [18]. Typically, each level set represents one basic geometric feature like a plane and Boolean operations are performed between them to capture an intersecting edge. This approach is also coupled to level set definition on a sub-grid to improve representation of curvatures. In Tran et al. [19], several level sets are also used for the representation of complex microstructures, each one representing an inclusion.

In the present paper, it is proposed to use a single level set for the implicit representation of the structure. For a correct representation of sharp features, information from the geometry is added to the level set so that the representation is not purely implicit anymore but semi-implicit.

The paper is structured as follows. In Section 2, the definition of level set to represent implicitly a boundary is recalled as well as the way to use it in a finite element context. The effect of

* Corresponding author at: Computational Multiphysics Software Development Team, Cenaero, Belgium.

http://dx.doi.org/10.1016/j.finel.2016.04.004
0168-874X/

Please cite this article as: H. Asadi Kalameh, et al., Semi-implicit representation of sharp features with level sets, Finite Elem. Anal. Des. (2016), http://dx.doi.org/10.1016/j.finel.2016.04.004

smoothing corners is highlighted. In Section 3, the concept of *level set plus* is introduced. It enriches the classical level set with geometrical information to correctly represent the sharp features. In Section 4, Boolean operations on level set plus are examined. Finally, in Section 5, numerical validations are presented for semi-implicit geometry representation, Boolean operations and finite element computations. In what follows "corner" is used along with of "sharp feature" with the same meaning.

2. Interface representation with level set

In this section, basic notations and methodologies used for representation of interfaces with classical level set in the context of a finite element problem are recalled. Problems related to the capture of corners are highlighted.

2.1. The classical level set method

The primary concept of the level set technique is to implicitly describe an interface Γ by a function [3]. The level set function Φ is a signed distance function with respect to Γ such that the following sign convention applies:

$$\begin{cases} \Phi(x) < 0 \Leftrightarrow & x \in \Omega^- \\ \Phi(x) = 0 \Leftrightarrow & x \in \Gamma \\ \Phi(x) > 0 \Leftrightarrow & x \in \Omega^+ \end{cases} \tag{1}$$

where x is a point in the level set support Ω containing the interface. The level set support can be defined on an unstructured or structured non-conforming mesh. Ω^- and Ω^+ are sub-domains of Ω on both sides of the interface such that $\Omega^- \cup \Gamma \cup \Omega^+ = \Omega$ and $\Omega^- \cap \Omega^+ = \varnothing$. When the level set is used to represent implicitly a structure embedded within the level set support, Ω^- is the structure and Γ is the boundary. In this paper, the terms *internal* and *external* regions respectively correspond to Ω^- and Ω^+.

2.2. Usage of level set within a finite element simulation

In the context of this work, when the level set is used within a finite element simulation, Ω is considered to be a 2D triangular first-order mesh. Decomposition to simplicial elements is performed for other element types. Level set values are computed at every node belonging to Ω. For nodes which are very close to the interface (e.g., $|\Phi(x)| < 0.01$ e, where e is the element edge size), the level set value is arbitrarily set to 0 to avoid narrow sub-elements for integration.

In the finite element context, for the imposition of Neumann boundary conditions on the interface, an *Interface-mesh* is reconstructed from the level set. The interface, defined by the iso-0 level set, i.e. a curve on which $\Phi(x) = 0$, is obtained by evaluating a level set field with help of the finite element shape functions. In case of first-order shape functions, the intersection points on edges are determined with a linear interpolation between the level set values calculated at the end-points of each edge.

In order to create bijective relations between mesh entities (vertex, edge and face) of the level set support and intersection points, a tagging system is introduced for linking entities. As illustrated in Fig. 1, an interface-tag TAG-I is defined, so that any side of a relation can be retrieved by knowing the other side and the tag. Connecting these intersection points with line segments results in the construction of the *Interface-mesh* which is a poly-line in 2D. As illustrated in Fig. 1, each of these poly-line segments is also tagged to the corresponding element.

In addition to the *Interface-mesh*, a *Submesh* associated to each element cut by the *Interface-mesh* is constructed for the purpose of integration over the element. Typically, different integration rules can be used on both sides of the interface (i.e. materials interface) or even no integration at all on Ω^+ in case of an implicitly defined volume. It is important to notice that the use of *Submesh* is limited to integration and visualization purposes, without introduction of additional degrees of freedom to the finite element problem.

Creation of the *Submesh* is shown in Fig. 2. Elements crossed by the iso-0 level set are subdivided into sub-elements (triangles) whose edges are conforming to the iso-0 level set. Similar Gaussian quadrature rule is used on each sub-element as for the uncut elements.

The subdivision algorithm is the following. Intersection points which are detected during construction of the *Interface-mesh* are retrieved from support entities using the interface-tag (TAG-I) and then subdivision is performed for each element so that each sub-element is located on either side of the *Interface-mesh*. As shown in Fig. 2, intersection points and the vertices are duplicated and

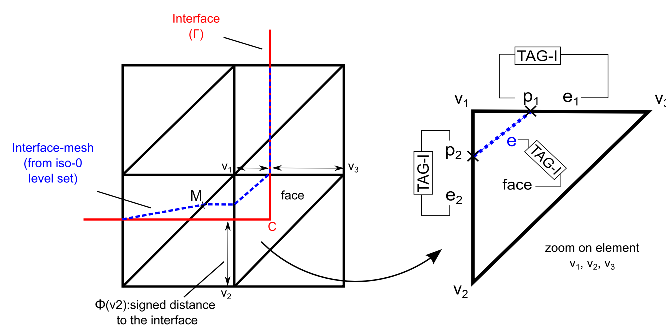

Fig. 1. Classical level set approach: creation of the *Interface-mesh* and associated tagging system on the element containing the corner *C*.

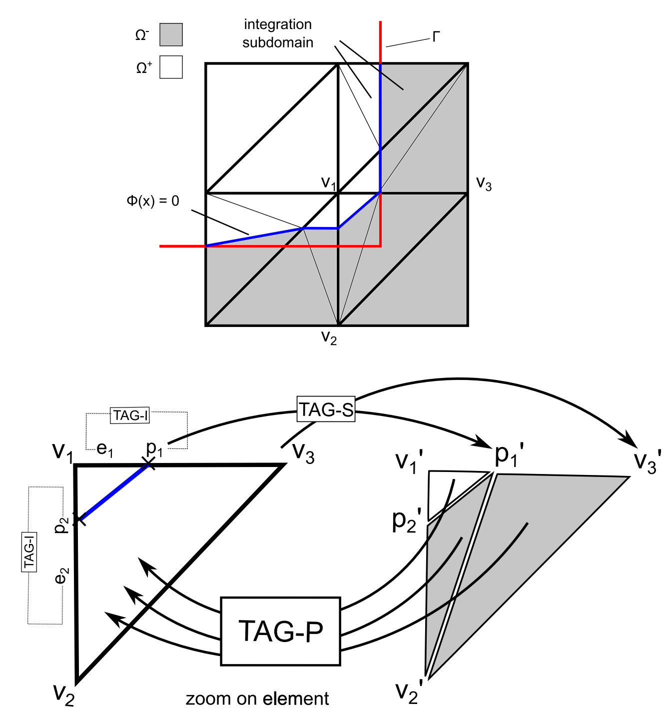

Fig. 2. Classical level set approach: creation of the associated *Submesh* on the element containing the corner.