Advances in Engineering Software 109 (2017) 1–14

Contents lists available at ScienceDirect

Advances in Engineering Software

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

Research paper

# A microstructure modeling scheme for unidirectional composites using signed distance function based boundary smoothing and element trimming

Jae Hyuk Lim[a], Hobeom Kim[b], Sun-Won Kim[c], Dongwoo Sohn[d,*]

[a] Department of Mechanical Engineering, College of Engineering, Chonbuk National University, 567 Baekje-daero, Deokjin-gu, Jeonju, Jeonbuk 54896, Republic of Korea
[b] Department of Mechanical Engineering, Korea Advanced Institute of Science and Technology (KAIST), 291 Daehak-ro, Yuseong-gu, Daejeon 34141, Republic of Korea
[c] Satellite Mechanical Department, Korea Aerospace Research Institute, 169-84 Gwahak-ro, Yuseong-gu, Daejeon 34133, Republic of Korea
[d] Division of Mechanical Engineering, College of Engineering, Korea Maritime and Ocean University, 727 Taejong-ro, Yeongdo-gu, Busan 49112, Republic of Korea

ARTICLE INFO

*Article history:*
Received 11 August 2016
Revised 19 January 2017
Accepted 12 February 2017
Available online 22 March 2017

*Keywords:*
Trimmed meshes
Image processing
Signed distance function
Unidirectional composites

ABSTRACT

A simple and accurate scheme for modeling microstructures is proposed with the help of element trimming combined with signed distance function based boundary smoothing. To accommodate randomly distributed fibers in unidirectional composites, digital image processing is used. The interfaces of multi-materials are identified by introducing a signed distance function, and then, square background elements crossing the interfaces are simply trimmed and divided to represent a single material behavior by a single element. After element trimming, the elements that are polygon-shaped in the two-dimensional domain are split into conventional three-node triangle elements (six-node prism elements in the three-dimensional domain) available in many commercial software packages. The present modeling scheme was verified through benchmark examples in terms of the accuracy and efficiency and then applied to the modeling of unidirectional composites based on real microscopic images to evaluate the equivalent elastic properties.



## 1. Introduction

A variety of composite materials (reinforced plastics, metal composites, ceramic composites and so forth) have been widely adopted in various applications in the automotive, offshore plant, construction, sporting, and aerospace industries due to their high stiffness and strength-to-weight ratio. Unidirectional (UD) fiber-reinforced polymer composites containing microstructures have been routinely modeled by two constituents (the fiber and matrix) or by more than two constituents (fiber, matrix, void and interphase/interface). The microstructure conditions at the small-scale level (constituent-level), such as dimensions, shapes, spatial distributions, material properties of the constituents, strongly affect the interactions between the constituents and thus, have a critical role in the performance at the large-scale level (lamina-, laminate-, and structure-levels) [1,2]. Therefore, there have been numerous efforts to understand the link of the mechanical response between the small-scale and large-scale levels of UD composites.

As part of such efforts, to reflect the microstructure effects, many researchers have developed finite element (FE) model generators [3,4] to consider the statistical distribution of the equivalent elastic properties of the lamina and investigated the effect of artificially generated microstructures. They thus succeeded in a more accurate prediction higher than the FE solutions of simple hexagonal, square fiber arrays or analytic solutions using the Mori-Tanaka method with the assumption of transverse isotropy [5]. However, in a practical sense, manufacturing composite materials that have statistical fiber distributions is not a straightforward process because the microstructure configurations and material properties of composites are quite affected by the manufacturing conditions such as the curing cycle, tool-part interaction during the curing process, level of applied vacuum pressure, temperature control and so forth [5]. Therefore, artificially generated fibers in numerical modeling are somewhat different from real fibers not to mention that their shapes are not purely circular and their diameters are not constant. To accurately investigate the mechanical behavior of UD composites, more realistic FE models that reflect the

* Corresponding author.
*E-mail address:* dsohn@kmou.ac.kr (D. Sohn).

http://dx.doi.org/10.1016/j.advengsoft.2017.02.014

microscope images of SEM or TEM are required as presented by many researchers [6–13]. To accommodate a real microscope image directly as it is, voxel and voxel-like methods have been introduced [6]. Because these methods treat one pixel to be one square FE or four triangle FEs, a large number of nodes have to be used to obtain accurate results for complex objects. It is thus quite difficult to cover a large-size domain due to limited computer resources. In addition, generally, they generate jagged edges that can cause an undesirable stress concentration in contrast to smooth boundaries in real materials. Furthermore, continuing efforts to generate FE meshes that accommodate microstructures through image processing have been reported [7–13]. On the other hand, as a conventional method, it is possible to create a FE model directly when the CAD file of the microstructure is already arranged. However, creating CAD files of complex objects could still be a cumbersome task even though there have been many dramatic enhancements in commercial CAD software, and a few successes in converting images to CAD files have been reported [14–16].

From the viewpoint of computational mechanics, to consider the detailed configuration of microstructures efficiently, many multi-material modeling schemes have been proposed by several researchers using the extended finite element method (X-FEM) [17,18], mesh-free methods [19,20], generalized finite element method (G-FEM) [21], and the element carving/trimming scheme [22]. In all of them, basically, a background mesh with square elements is used to cover the entire domain of problems. To identify each material phase on the background mesh, the level-set method [23] has been frequently used. In the level-set method, object boundaries are represented implicitly using the minimum signed distance function (SDF) values as nodal variables. Especially, when the level-set method is combined with the aforementioned multi-material modeling scheme such as the X-FEM, it provides an efficient modeling tool for dealing with various singularities including cracks, multi-material interfaces, dislocations and so forth without remeshing. It also allows one element to cover multi-material constituents with a subdomain (distinguished by the signed distance)-wise Gauss integration. In contrast, in the element carving/trimming scheme, the elements crossing multi-material interfaces are locally trimmed so that each element cover only one material. In other words, this scheme splits the square elements in the background mesh to several polygon-shaped pieces. In particular, when the trimmed square elements, which are polygonal-shaped, is further divided into three-node triangle elements in the two-dimensional (2D) domain (six-node prism elements in the three-dimensional (3D) domain), FE models obtained by this scheme can be directly used in FE analysis by means of any commercial FE software such as ABAQUS.

The goal of this work was to propose a simple and accurate multi-material modeling scheme using the SDF based boundary smoothing and element trimming techniques. The remainder of this paper is organized as follows. In Section 2, we explain how to construct trimmed FE meshes to fit multi-material interfaces represented by the SDF. This is followed by an element merging scheme to enhance the mesh quality. Some distorted and small-area elements, which can be generated during the element trimming, should be merged into large elements to avoid the ill-conditioning or nearly zero values of the stiffness matrix. This manipulation can thus provide accurate solutions comparable to those of conventional FE meshes with CAD files. Next, in Section 3, to verify the efficiency and accuracy of the proposed scheme compared with other numerical schemes, we solve benchmark problems regarding plates containing a hole or an inclusion. In Section 4, we show the effectiveness of the proposed scheme in the FE modeling of UD composites with complex fiber configurations, which are acquired from image processing. Finally, we finish the paper with concluding remarks in Section 5.

## 2. Multi-material representation using the SDF based boundary smoothing and element trimming

### 2.1. SDF based boundary smoothing algorithm

Consider an analysis domain covered by square-background elements. Fig. 1 shows two material regions, $\Omega_1$ and $\Omega_2$, and their interface, $\partial\Omega=\Omega_1\cap\Omega_2$. If a few objects are located in the domain, the material interface is represented using the minimum SDF. For any material point $\mathbf{x}$ in the domain at time $t$, the minimum SDF $\phi(\mathbf{x},t)$ is defined as the minimum distance between the nodes of the square elements and the material interfaces which is a scalar variable independently defined at the nodes. The zero SDF value ($\phi(\mathbf{x},t)=0$) corresponds to the interface of the two materials. For the case of multi-materials ($n$-materials), the independent variables, for which the number is $(n-1)$, have to be considered at the nodes. With this concept, the X-FEM has reported remarkable successes for many singularity problems [17,18] without additional local remeshing because it describes the material interfaces implicitly using the SDF. However, in our approach, we explicitly split the square elements crossing multi-material boundaries into several polygon-shaped pieces (see Fig. 2(a)) which are represented by the combination of simple three-node triangle elements in the 2D domain (six-node prism elements in the 3D domain) shown in Fig. 2(b). As a result, using element trimming based on the SDF values, all of the material boundaries have a smoothed representation.

Before calculating the SDF values, a high-quality image should be first prepared. To obtain an image with local contrast enhancement and noise reduction, we conduct a preprocessing using two well-known image filters; a median filter taking the median value to remove speckles/dots on an image, and a Gaussian high-pass filter to sharpen an image. The two filters are supported by MATLAB image processing toolbox [24]. Subsequently, for a given black-and-white patterned image passed through the filters, we consider the following two techniques to define the minimum SDF for detecting material boundaries.

The first way of obtaining SDF values from raw images is to solve iteratively a level-set equation given by Eq. (1) on square-type background meshes, until the solution of the level-set equation is converged to constant values within a tolerance limit [23,25]:

$$\frac{\partial\phi(\mathbf{x},t)}{\partial t}+\vec{v}\,\nabla\phi=0, \tag{1}$$

where the minimum SDF $\phi$ should satisfy the following properties;

$$\phi(\mathbf{x},t)<0 \text{ for } \mathbf{x}\in\Omega_1, \tag{2}$$

$$\phi(\mathbf{x},t)>0 \text{ for } \mathbf{x}\in\Omega_2, \tag{3}$$

$$\phi(\mathbf{x},t)=0 \text{ for } \mathbf{x}\in\partial\Omega. \tag{4}$$

When the normal component $v_N$ of velocity $\vec{v}$ is taken, Eq. (1) becomes the following partial differential equation:

$$\frac{\partial\phi(x,y)}{\partial t}+v_N|\nabla\phi|=0. \tag{5}$$

Considering the curvature-based level-set evolution, or mean curvature flow, with a curvature-based force that smoothes the curve, $v_N=-b\kappa$, Eq. (5) can be rewritten as [25]

$$\frac{\partial\phi(x,y)}{\partial t}=b\kappa|\nabla\phi|, \tag{6}$$

where $\kappa$ is the curvature, and $b$ is a weighting parameter for the curvature-based force. Eq. (6) can be discretized using central differencing with a uniform Cartesian grid. In this work, we use an