



An adaptive shell element for explicit dynamic analysis of failure in laminated composites Part 1: Adaptive kinematics and numerical

Downloaded from: <https://research.chalmers.se>, 2026-04-04 13:14 UTC

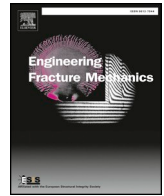
Citation for the original published paper (version of record):

Främby, J., Fagerström, M., Karlsson, J. (2020). An adaptive shell element for explicit dynamic analysis of failure in laminated composites Part 1: Adaptive kinematics and numerical implementation. *Engineering Fracture Mechanics*, 240. <http://dx.doi.org/10.1016/j.engfracmech.2020.107288>

N.B. When citing this work, cite the original published paper.

Contents lists available at [ScienceDirect](https://www.sciencedirect.com)

Engineering Fracture Mechanics

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

An adaptive shell element for explicit dynamic analysis of failure in laminated composites Part 1: Adaptive kinematics and numerical implementation

Johannes Främby^{a,*}, Martin Fagerström^a, Jesper Karlsson^b

^a Department of Industrial and Materials Science, Division on Material and Computational Mechanics, Chalmers University of Technology, Gothenburg, Sweden

^b DYNAMore Nordic AB, Gothenburg, Sweden

ARTICLE INFO

Keywords:

Adaptivity
Shell elements
Crash simulations
Composites
LS-DYNA

ABSTRACT

To introduce more fibre-reinforced polymers in cars, the automotive industry is strongly dependent on efficient modelling tools to predict the correct energy absorption in crash simulations. In this context, an adaptive modelling technique shows great potential. However, as the critical energy absorption in a crash occurs over a very short period of time, and since the deformation behaviour is very complex, car crash simulations are usually performed using explicit dynamic finite element solvers. Therefore, any practical adaptive technique must be adapted to an explicit setting in a software available to the automotive companies. In this paper, we propose an adaptive method for explicit finite element analysis and describe its implementation in the commercial finite element solver LS-DYNA. The method allows for both so-called weak discontinuities (discontinuities in strain), which are crucial for accurate stress and intralaminar damage predictions, and strong discontinuities (discontinuities in displacements), needed for a proper representation of growing delamination cracks. In particular, we detail the implementation of the proposed method into LS-DYNA and also how we propose to remedy the non-physical oscillations arising from the implementation of the adaptive scheme in a explicit dynamic setting. The paper is concluded with numerical examples where we demonstrate the potential for the adaptive approach and also perform a detailed study on its accuracy and stability.

1. Introduction

In order to meet the increasing demands from regulatory bodies on CO₂ emissions from automotive vehicles, the automotive industry is currently very active in reducing vehicle weight. One significant step is to increase the amount of laminated Fibre-Reinforced Polymers (FRP) due to their superior specific properties (e.g. specific strength and specific energy absorption in axial crushing) compared to conventional metals, cf. e.g. Carruthers et al. [1], Jacob et al. [2] or Park et al. [3]. The introduction of laminated FRP in the automotive industry is strongly dependent on accurate and efficient modelling tools to predict the correct energy absorption in crash simulations [4]. From experimental observations during axial crushing (see e.g. Hull [5] and Grauers et al. [6]), it is obvious that in order to obtain good predictability in the simulations, delamination growth needs to be accounted for. Common practice to model delamination initiation and propagation is to model each ply by separate elements, interconnected by cohesive interface (Cohesive Zone, CZ) elements. However, due to restrictions on the simulation time, such high-fidelity layerwise (LW) models are not feasible to use in industrial applications such as full vehicle crash simulations.

* Corresponding author.

E-mail address: johannes.framby@chalmers.se (J. Främby).

<https://doi.org/10.1016/j.engfracmech.2020.107288>

Received 6 February 2020; Received in revised form 13 August 2020; Accepted 24 August 2020

Available online 13 September 2020

0013-7944/ © 2020 The Authors. Published by Elsevier Ltd. This is an open access article under the CC BY license (<http://creativecommons.org/licenses/by/4.0/>).

Nomenclature			
A	area	\mathbf{v}	velocity
\mathbf{a}	acceleration	V	volume
C	damping	\mathbf{X}	reference coordinates
$C_{d,ref}$	weak refinement damping (peak) value	\mathbf{x}	current coordinates
c	nodal mass relation	α	relative position of refinement with respect to bottom and top surfaces
E	elastic modulus	Δt_{ref}	duration of weak refinement damping
\mathbf{f}	internal forces	Δt_{CZ}	duration of strong refinements introduction
f	failure index	$\Delta \theta_{del}$	interface pitch angle threshold for allowing refinements
$f_{CZ,lim}$	interlaminar failure threshold for activation of strong refinements	ζ	damping ratio
G	shear modulus	κ	weight function
\mathcal{G}_c	critical fracture energy	ν	Poisson's ratio
k	cohesive penalty stiffness	ρ	density
\mathbf{M}	mass matrix	σ	Cauchy stress
m	nodal mass	Ω	oscillations
\mathbf{P}	external forces	ω_n	dominating angular frequency
\mathcal{P}	projection matrix for subelemental coordinates and velocities	AFEM	Augmented Finite Element Method
S_{CZ}	interlaminar shear strength	CZ	Cohesive Zone
T_{CZ}	interlaminar normal strength	DOF	Degrees Of Freedom
t	time	ESL	Equivalent Single-Layer
t_{ref}	weak refinement time	FE	Finite Element
\mathbf{u}	displacement	FRP	Fibre-Reinforced Polymers
		LW	LayerWise

A solution to reduce computational cost while maintaining the same level of accuracy as high-fidelity models is to resort to an adaptive modelling technique, where an initially coarse model can be locally refined during the simulation. This way, the costs associated with LW models are limited to the areas where it is needed. We have previously presented an adaptive enrichment method for modelling of multiple and arbitrarily located delamination cracks using an equivalent single-layer (ESL) shell model [7]. During a simulation, ESL shell elements are adaptively refined through the thickness and CZ elements are added. Thus, locally the element transforms to a LW model capable of describing the delamination initiation and propagation.

The use of CZ interfaces can sometimes lead to an overly compliant response since the CZ penalty stiffness cannot be set too high. However, by employing an adaptive method where CZ elements are only added when and where needed, the extent of this problem can be reduced.

We want to emphasize that our work is not the only contribution on adaptive refinement or remeshing techniques. Just in the application of FRP there have been several similar approaches presented, e.g. the work by Remmers et al. [8], van der Meer and Sluys [9], Brouzoulis and Fagerström [10,11], Reiner et al. [12], Shor and Vaziri [13], McElroy [14], Lu et al. [15], Zhi and Tay [16] and Adams et al. [17]. We will relate our method to these contributions in the discussion but already here we want to point out that, even though different techniques are used, all except [13] use an implicit solver. This was also the case for our work in [7].

However, due to the extreme level of complexity, industrial full vehicle crash simulations are usually performed using commercial explicit dynamic solvers. Besides being easily available for industrial use, any adaptive method must therefore also be suitable for this type of solver. This creates challenges since dynamic solvers are very sensitive to modifications like remeshing during the simulation. Consequently, besides yielding as accurate results as high-fidelity models while maintaining computational efficiency, the implementation must be numerically robust.

To facilitate adaptive modelling in industrial crash simulations, a modified version of our method [7], previously implemented in the open source software OOFEM [18], has therefore been implemented as a user element in the commercial finite element (FE) solver LS-DYNA [19], a well-accepted industry tool for performing automotive crash simulations. The numerical implementation described in this paper includes remedies to the non-physical oscillations that can occur when refining elements during a simulation in an explicit dynamic solver. We will show that we can reproduce similar results as reference high-fidelity models while saving computational effort.

The current paper is the first part of two closely related papers (Part I and II). In Part II [20] we will utilise the adaptive method and further focus on issues regarding fracture modelling of laminated FRP and validation of the method. We believe that the work we present in these two papers has the potential of enabling large scale predictive simulations of progressive crash failure in FRP, using reasonable amount of computational effort. As mentioned above, this is crucial for structural composites to have a widespread use in future cars.

1.1. Outline of paper

First we set out to describe the implemented method. This includes explaining the element structure and the calculation of internal forces. Next, we will focus on the adaptivity and the challenges related to this. To verify the model we present some numerical examples where we show the performance of the implementation. We end the paper by summarising our work and try to set it into context of other contributions in the field.

2. Implementation of the adaptive method

In this work, we have extended and implemented a modified version of the method from [7] as a user element in the commercial FE solver LS-DYNA. In summary, the method consists of the following steps:

- i The laminated structure is initially represented by a single layer of solid shell elements through the thickness.
- ii Refinement indicators are used to locate areas which need to be refined.
- iii The shell elements are refined through the thickness by enriching the element kinematics to account for material interfaces (weak discontinuities), cf. Section 2.3.1.
- iv At interfaces prone to delaminate, CZ elements are inserted such that delaminations (strong discontinuities) can initiate and propagate, cf. Section 2.3.2.

This means that every adaptive element in the model can be in one of three stages: a) Unrefined; b) Refined in one or several material interfaces; c) Having one or several refined interfaces separated and CZ elements inserted. This is illustrated in Fig. 1. In the following we will refer to the first refinement step, where material interfaces are added through the thickness, as *weak refinements*. The secondary refinement step, where CZ elements are inserted, will be referred to as *strong refinements*.

Remark: In [7] refinements could be limited to a local in-plane patch of elements, which could be expanded when necessary. Due to the choice of enrichment method in the current LS-DYNA implementation (detailed below) it is not as straightforward to guarantee continuity at the refinement boundary as it was in the OOFEM implementation. Therefore, the refinement is made for all elements of the laminate in question. The work in this paper is therefore to be considered as proof of concept for the adaptive method and local patch refinement and expansion will be implemented ahead.

In order to limit the computational costs, the method only models interlaminar fracture with separate Degrees of Freedom (DOF), *i.e.* no in-plane refinements in order to model intralaminar fracture are made. Instead, this is modelled using a smeared-crack intralaminar material model. Nevertheless, the through-the-thickness refinements will improve the prediction of the stress state in the element and thus also improve the prediction of intralaminar fracture.

In the following sections we will describe the different parts of the method. As we have made the implementation as an LS-DYNA user element, most of the functions are performed locally in each element. The implementation is based on the work performed in [21], where one delamination enrichment was hard coded in the middle of a solid shell element (eight nodes with only translational DOF). Furthermore, the refinements are represented using internal subelements, *i.e.* using a so-called Augmented Finite Element Method (AFEM) [22] approach. However, as we stated already in [7] the adaptive method is not dependent on the choice of either element formulation or enrichment technique, instead the key idea is that the delamination DOF are introduced only when and where they are needed during the simulation.

The main part of the implementation consists of a solid shell user element, which can be refined into subelements by utilising so-called extra nodes with three translational DOF each. In the following subsections we will describe the implemented method more in detail by starting with the element connectivity.

Remark: Currently LS-DYNA does not allow for the extra nodes to be dynamically allocated during the simulation, *i.e.* all (including the non-active) extra nodes used to define the adaptive refinements are updated by the LS-DYNA solver during the entire simulation. At the start of the simulation all field variables associated with the extra nodes are set to zero. These will remain zero until the node is activated and included in the internal force calculation of the elements. At this instant, the current position and velocity

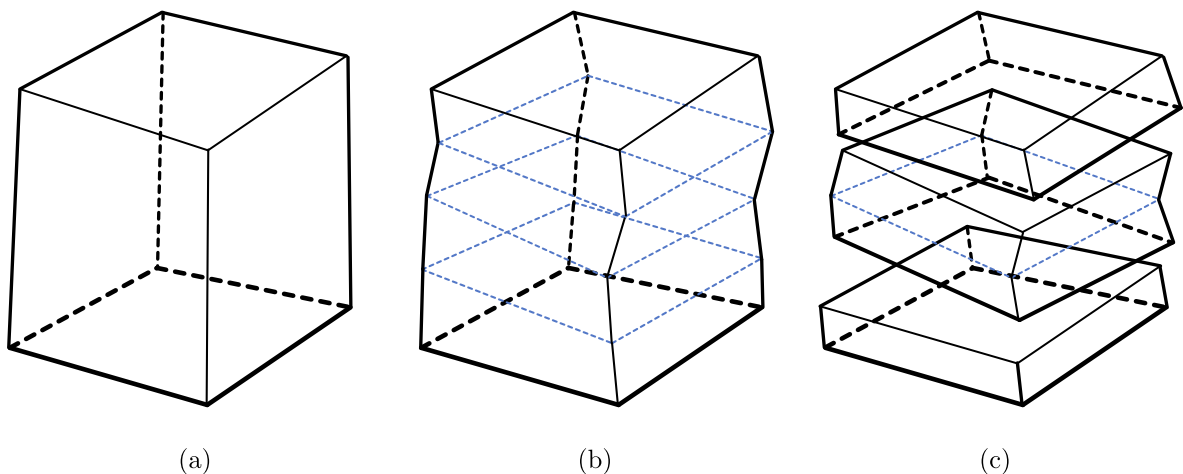


Fig. 1. The different stages of the adaptive elements: In the unrefined stage the entire laminate is represented by one element through the thickness (a); the element can be through-the-thickness refined in order to model weak discontinuities, here exemplified with three refinements (b); the refined interfaces from the previous stage can be separated (and have CZ elements added) in order to model strong discontinuities (c).

will be updated to the appropriate values as described in Section 2.3. Thus, any improved computational efficiency is currently only related to the internal computations of the user element.

2.1. Element connectivity

The base element consists of 8 vertex nodes. These base nodes are each associated with N so-called extra (or phantom) nodes, making a total of $8(N + 1)$ nodes, where the number N is the number of active refinements in the element.¹ Please note that the extra nodes are not element local. Instead, they are global like their parent base nodes and thereby shared between elements. We want to make it clear that unless the analysis input specifies otherwise, no refinements are present from the start. These are activated from within the elements when certain conditions are met during the simulation, cf. Section 2.3.

The internal numbering of nodes in an element starts from the bottom surface going upwards counter-clockwise around each respective surface. That is, the nodal numbering will depend on the number of (active) refinements, as is illustrated in Fig. 2. Let \mathbf{X}_k be the reference coordinates at time $t = 0$ for node k . Then, eight base nodes will represent the bottom and top surface of the element as

$$\begin{aligned} \mathbf{X}_1 &= \hat{\mathbf{X}}_1 & \mathbf{X}_{8N+5} &= \hat{\mathbf{X}}_5 \\ \mathbf{X}_2 &= \hat{\mathbf{X}}_2 & \mathbf{X}_{8N+6} &= \hat{\mathbf{X}}_6 \\ \mathbf{X}_3 &= \hat{\mathbf{X}}_3 & \mathbf{X}_{8N+7} &= \hat{\mathbf{X}}_7 \\ \mathbf{X}_4 &= \hat{\mathbf{X}}_4 & \mathbf{X}_{8N+8} &= \hat{\mathbf{X}}_8 \end{aligned} \quad \text{and} \quad \mathbf{X}_{8i+5} = \hat{\mathbf{X}}_5, \quad \mathbf{X}_{8i+6} = \hat{\mathbf{X}}_6, \quad \mathbf{X}_{8i+7} = \hat{\mathbf{X}}_7, \quad \mathbf{X}_{8i+8} = \hat{\mathbf{X}}_8, \quad i = 1 \dots N, \quad (1)$$

where $\hat{\mathbf{x}}$ denotes base nodes of the element.

In the presence of internal interfaces, the extra nodes will represent the surfaces of the subelements. Their reference positions are defined to be an interpolation of the bottom and top base nodes as

$$\begin{aligned} \mathbf{X}_{8(i-1)+5} &= \mathbf{X}_{8i+1} = \hat{\mathbf{X}}_1 + \alpha_i (\hat{\mathbf{X}}_5 - \hat{\mathbf{X}}_1) \\ \mathbf{X}_{8(i-1)+6} &= \mathbf{X}_{8i+2} = \hat{\mathbf{X}}_2 + \alpha_i (\hat{\mathbf{X}}_6 - \hat{\mathbf{X}}_2) \\ \mathbf{X}_{8(i-1)+7} &= \mathbf{X}_{8i+3} = \hat{\mathbf{X}}_3 + \alpha_i (\hat{\mathbf{X}}_7 - \hat{\mathbf{X}}_3) \\ \mathbf{X}_{8(i-1)+8} &= \mathbf{X}_{8i+4} = \hat{\mathbf{X}}_4 + \alpha_i (\hat{\mathbf{X}}_8 - \hat{\mathbf{X}}_4) \end{aligned} \quad i = 1 \dots N, \quad (2)$$

where $\|\alpha_i\| \in]0,1[$ is a parameter which specifies the relative position of an interface refinement i with respect to the bottom and top surfaces. This means that, if a refinement is activated during the simulation its associated reference nodal positions will be calculated from the undeformed element. From Eq. 2 it can also be seen that it is assumed that each internal interface consists of two coinciding surfaces, defined by eight extra nodes. However, during the first weak refinement stage only one set of four extra nodes is active while an additional set of four is reserved for later use. If the interface is then strongly refined the reserved set of nodes is activated. As an illustration, in Fig. 2 the nodal numbering for two active refinements is shown.

Let \mathbf{x}_k be the current coordinates of node k . Then

$$\mathbf{x}_k = \mathbf{X}_k + \mathbf{u}_k \quad k = 1 \dots 8(N + 1), \quad (3)$$

where \mathbf{u}_k is the displacement of the node. The displacements are calculated by LS-DYNA integrating the equations of motions using central-difference time integration [23]. In order to perform the time integration, forces (and masses) associated to all the DOF must be assembled by the user-element routine, which will be described in the next sections.

2.2. Internal force calculation

In each time step ${}^n t$ the user element is fed with current nodal coordinates ${}^n \mathbf{x}$ and velocities ${}^{n-1/2} \mathbf{v}$ from the LS-DYNA solver and must return the element internal forces ${}^n \mathbf{f}$.

The size of ${}^n \mathbf{x}$, ${}^{n-1/2} \mathbf{v}$ and ${}^n \mathbf{f}$ as well as the reference coordinates \mathbf{X} will depend on the number of active refinements. For the case of no initial refinements:

$$\mathbf{X} = \begin{bmatrix} \mathbf{X}_1 \\ \mathbf{X}_2 \\ \mathbf{X}_3 \\ \mathbf{X}_4 \\ \mathbf{X}_5 \\ \mathbf{X}_6 \\ \mathbf{X}_7 \\ \mathbf{X}_8 \end{bmatrix}, \quad {}^n \mathbf{x} = \begin{bmatrix} {}^n \mathbf{x}_1 \\ {}^n \mathbf{x}_2 \\ {}^n \mathbf{x}_3 \\ {}^n \mathbf{x}_4 \\ {}^n \mathbf{x}_5 \\ {}^n \mathbf{x}_6 \\ {}^n \mathbf{x}_7 \\ {}^n \mathbf{x}_8 \end{bmatrix}, \quad {}^{n-1/2} \mathbf{v} = \begin{bmatrix} {}^{n-1/2} \mathbf{v}_1 \\ {}^{n-1/2} \mathbf{v}_2 \\ {}^{n-1/2} \mathbf{v}_3 \\ {}^{n-1/2} \mathbf{v}_4 \\ {}^{n-1/2} \mathbf{v}_5 \\ {}^{n-1/2} \mathbf{v}_6 \\ {}^{n-1/2} \mathbf{v}_7 \\ {}^{n-1/2} \mathbf{v}_8 \end{bmatrix}, \quad {}^n \mathbf{f} = \begin{bmatrix} {}^n \mathbf{f}_1 \\ {}^n \mathbf{f}_2 \\ {}^n \mathbf{f}_3 \\ {}^n \mathbf{f}_4 \\ {}^n \mathbf{f}_5 \\ {}^n \mathbf{f}_6 \\ {}^n \mathbf{f}_7 \\ {}^n \mathbf{f}_8 \end{bmatrix}, \quad (4)$$

¹ N is at the moment limited to five in LS-DYNA (hard coded in the software), however, the number of extra nodes could theoretically be infinite.

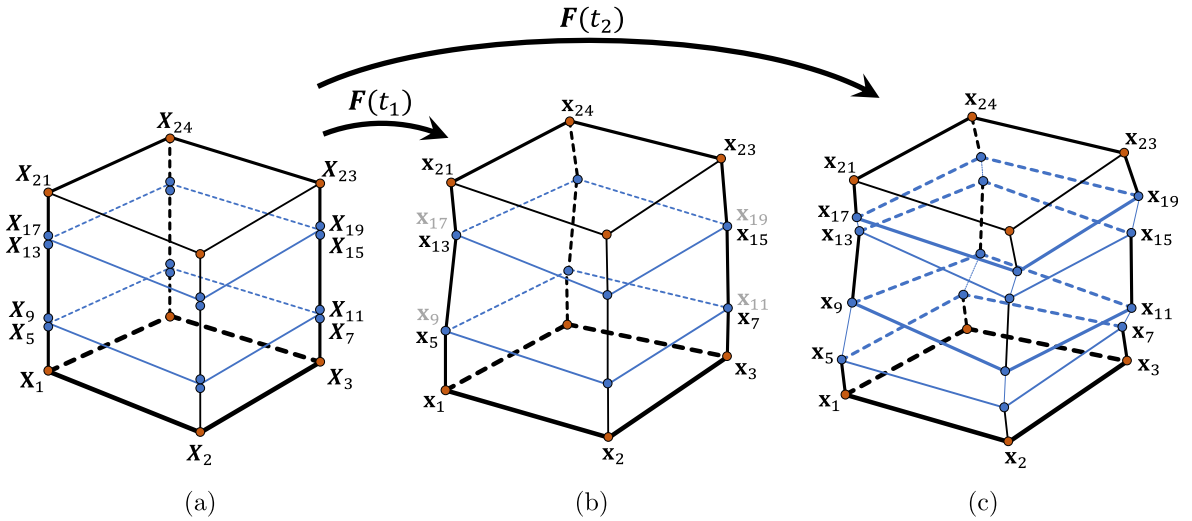


Fig. 2. Element connectivity for an element with two active refinements in an undeformed state (a) and deformed state with weak refinements (b) and with strong refinements (c). Orange nodes indicate base nodes and blue extra nodes, i.e. nodes 1–4 and 21–24 are base nodes and 5–20 are extra nodes. In the state with weak refinements, only one set of extra nodes is active per interface, while additional sets are reserved for later use (indicated in grey). Please note that the interface nodes coincide in the undeformed state but are drawn separately for clarity.

where the subindex indicates internal nodal numbers of the element. Thus, in this case only the base nodes are activate. However, when N refinements are activated the user element is divided into several sublaminates elements and the vectors of Eq. (4) will change to

$$X = \begin{bmatrix} X_1 \\ \vdots \\ X_{8(N+1)} \end{bmatrix}, \quad n\mathbf{x} = \begin{bmatrix} n\mathbf{x}_1 \\ \vdots \\ n\mathbf{x}_{8(N+1)} \end{bmatrix}, \quad n^{-1/2}\mathbf{v} = \begin{bmatrix} n^{-1/2}\mathbf{v}_1 \\ \vdots \\ n^{-1/2}\mathbf{v}_{8(N+1)} \end{bmatrix}, \quad n\mathbf{f} = \begin{bmatrix} n\mathbf{f}_1 \\ \vdots \\ n\mathbf{f}_{8(N+1)} \end{bmatrix}, \quad (5)$$

where the numbering follows the numbering scheme presented in Section 2.1. In the following we will neglect the index for the current time step n .

As mentioned above, the element can be in one of three stages. In the third state some of the interfaces can be weak refinements while other are strong refinements, representing strong discontinuities with CZ elements. This means that the full internal force vector \mathbf{f} of the element will have contributions both from the currently active internal sublaminates \mathbf{f}^{lam} and from CZ elements \mathbf{f}^{coh}

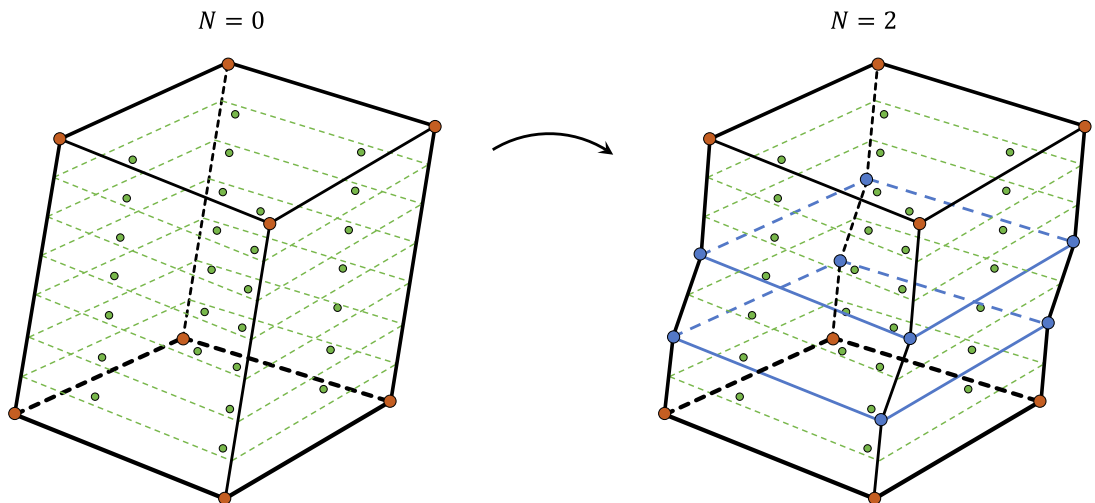


Fig. 3. Example of how an unrefined element with seven plies (left) is divided into sublaminates when the element is refined in interface two and four (right). The bottom two plies will be associated with the bottom subelement, ply three and four with the second subelement and the top three plies with a third and topmost subelement. The corresponding interface position array will be $\alpha = [2/7 \ 4/7]^T$. The element base nodes are indicated with orange and the extra nodes with blue. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

as:

$$\mathbf{f} = \mathbf{f}^{\text{lam}} - \mathbf{f}^{\text{coh}}. \quad (6)$$

The force vectors \mathbf{f}^{lam} and \mathbf{f}^{coh} will be defined in the next Subsections. Due to how the refinement is implemented, we will in the following refer to the sublaminates elements as *subelements*.

In our model, each laminate ply is represented by one integration point through the thickness. Thus, for an unrefined element ($N = 0$) all plies are included in one single (sub) element representing the entire laminate. When the first refinement is activated ($N = 1$), extra nodes are assigned at the appropriate interlaminar position. The plies below the refinement are assigned to a lower subelement and those above to an upper subelement. Further refinement of a subelement ($N = 2$) would divide the plies to new subelements below and above the refinement interface. This division of the plies is exemplified in Fig. 3.

Remark: Since each ply is represented by only one integration point through the thickness, a subelement consisting of one ply would become under integrated. Besides potentially resulting in hourglass deformation, bending cannot be modelled. In order to prevent this, refinements are not allowed such that a subelement consist of a single ply - there must be at least two. However, a more robust solution would be to increase the number of through-the-thickness integration points when a subelement happens to include only one ply. Alternatively, hourglass control must be enabled.

The challenge of calculating Eq. (6) is to gather the correct nodal positions and velocities and ply data for each subelement and then assemble the force contributions to the right places in \mathbf{f} . We will describe this procedure next.

2.2.1. Processing the laminate subelements

From \mathbf{X} , \mathbf{x} and \mathbf{v} in Eq. (5) subvectors associated with subelement i (placed directly beneath interface refinement i) can be extracted as

$$\mathbf{X}^{\text{sublam},i} = \begin{bmatrix} \mathbf{X}_{8(i-2)+5} \\ \mathbf{X}_{8(i-2)+6} \\ \mathbf{X}_{8(i-2)+7} \\ \mathbf{X}_{8(i-2)+8} \\ \mathbf{X}_{8(i-1)+5} \\ \mathbf{X}_{8(i-1)+6} \\ \mathbf{X}_{8(i-1)+7} \\ \mathbf{X}_{8(i-1)+8} \end{bmatrix}, \quad \mathbf{x}^{\text{sublam},i} = \begin{bmatrix} \mathbf{x}_{8(i-2)+5} \\ \mathbf{x}_{8(i-2)+6} \\ \mathbf{x}_{8(i-2)+7} \\ \mathbf{x}_{8(i-2)+8} \\ \mathbf{x}_{8(i-1)+5} \\ \mathbf{x}_{8(i-1)+6} \\ \mathbf{x}_{8(i-1)+7} \\ \mathbf{x}_{8(i-1)+8} \end{bmatrix}, \quad \mathbf{v}^{\text{sublam},i} = \begin{bmatrix} \mathbf{v}_{8(i-2)+5} \\ \mathbf{v}_{8(i-2)+6} \\ \mathbf{v}_{8(i-2)+7} \\ \mathbf{v}_{8(i-2)+8} \\ \mathbf{v}_{8(i-1)+5} \\ \mathbf{v}_{8(i-1)+6} \\ \mathbf{v}_{8(i-1)+7} \\ \mathbf{v}_{8(i-1)+8} \end{bmatrix} \quad (7)$$

if the interface below is a weak refinement, or

$$\mathbf{X}^{\text{sublam},i} = \begin{bmatrix} \mathbf{X}_{8(i-1)+1} \\ \mathbf{X}_{8(i-1)+2} \\ \mathbf{X}_{8(i-1)+3} \\ \mathbf{X}_{8(i-1)+4} \\ \mathbf{X}_{8(i-1)+5} \\ \mathbf{X}_{8(i-1)+6} \\ \mathbf{X}_{8(i-1)+7} \\ \mathbf{X}_{8(i-1)+8} \end{bmatrix}, \quad \mathbf{x}^{\text{sublam},i} = \begin{bmatrix} \mathbf{x}_{8(i-1)+1} \\ \mathbf{x}_{8(i-1)+2} \\ \mathbf{x}_{8(i-1)+3} \\ \mathbf{x}_{8(i-1)+4} \\ \mathbf{x}_{8(i-1)+5} \\ \mathbf{x}_{8(i-1)+6} \\ \mathbf{x}_{8(i-1)+7} \\ \mathbf{x}_{8(i-1)+8} \end{bmatrix}, \quad \mathbf{v}^{\text{sublam},i} = \begin{bmatrix} \mathbf{v}_{8(i-1)+1} \\ \mathbf{v}_{8(i-1)+2} \\ \mathbf{v}_{8(i-1)+3} \\ \mathbf{v}_{8(i-1)+4} \\ \mathbf{v}_{8(i-1)+5} \\ \mathbf{v}_{8(i-1)+6} \\ \mathbf{v}_{8(i-1)+7} \\ \mathbf{v}_{8(i-1)+8} \end{bmatrix} \quad (8)$$

if the interface below is a strong refinement.² The extractions in Eq. (7) and (Eq. (8)) can conveniently be formulated using projection matrices as

$$\mathbf{X}^{\text{sublam},i} = \mathcal{P}_{\text{weak}}^{\text{sublam},i} \mathbf{X}, \quad \mathbf{x}^{\text{sublam},i} = \mathcal{P}_{\text{weak}}^{\text{sublam},i} \mathbf{x}, \quad \mathbf{v}^{\text{sublam},i} = \mathcal{P}_{\text{weak}}^{\text{sublam},i} \mathbf{v} \quad (9)$$

and

$$\mathbf{X}^{\text{sublam},i} = \mathcal{P}^{\text{sublam},i} \mathbf{X}, \quad \mathbf{x}^{\text{sublam},i} = \mathcal{P}^{\text{sublam},i} \mathbf{x}, \quad \mathbf{v}^{\text{sublam},i} = \mathcal{P}^{\text{sublam},i} \mathbf{v}, \quad (10)$$

respectively.

The position and velocity vectors together with the sublaminates material properties (including material orientations) and integration point history data are then used to call an element subroutine which calculates the subelement internal force vector $\mathbf{f}^{\text{sublam},i}$ of the current time step. In our implementation the element subroutine is basically the same as the LS-DYNA *PART_COMPOSITE_TSHHELL element formulation³ but with the assumption of equally thick plies of the same material. Therefore, we will not give more details on the element formulation and how the forces are integrated, but refer to the LS-DYNA theory manual [23]. Furthermore, the material orientations are defined using an elemental midsurface (LS-DYNA parameter AOPT = 3) [24].

Once the subelement force is calculated it must be assembled in the element force vector. This can be done using the projection

² The bottom surface $i = 0$ is always considered a strong discontinuity, even if it shares nodes with an element below.

³ This is the simplest solid shell element with 3D stress state currently available in LS-DYNA and is therefore appropriate for development of the adaptive method. However, other formulations should be explored in future developments.

matrices above as

$$\mathbf{f}^{\text{lam}} = \sum_{i \in I} \mathcal{P}_{\text{weak}}^{\text{sublam},i\text{T}} \mathbf{f}^{\text{sublam},i} + \sum_{i \in J} \mathcal{P}^{\text{sublam},i\text{T}} \mathbf{f}^{\text{sublam},i}, \quad (11)$$

where I contains those subelements with weak interfaces below and J contains all other (including the bottom subelement).

2.2.2. Processing the cohesive interface elements

The nodal positions and velocities associated with the CZ elements can be extracted in a similar way as for the subelements. That is, for an interface i

$$\mathbf{X}^{\text{coh},i} = \begin{bmatrix} \mathbf{X}_{8(i-1)+5} \\ \mathbf{X}_{8(i-1)+6} \\ \mathbf{X}_{8(i-1)+7} \\ \mathbf{X}_{8(i-1)+8} \\ \mathbf{X}_{8i+1} \\ \mathbf{X}_{8i+2} \\ \mathbf{X}_{8i+3} \\ \mathbf{X}_{8i+4} \end{bmatrix} \quad \mathbf{x}^{\text{coh},i} = \begin{bmatrix} \mathbf{x}_{8(i-1)+5} \\ \mathbf{x}_{8(i-1)+6} \\ \mathbf{x}_{8(i-1)+7} \\ \mathbf{x}_{8(i-1)+8} \\ \mathbf{x}_{8i+1} \\ \mathbf{x}_{8i+2} \\ \mathbf{x}_{8i+3} \\ \mathbf{x}_{8i+4} \end{bmatrix} \quad \mathbf{v}^{\text{coh},i} = \begin{bmatrix} \mathbf{v}_{8(i-1)+5} \\ \mathbf{v}_{8(i-1)+6} \\ \mathbf{v}_{8(i-1)+7} \\ \mathbf{v}_{8(i-1)+8} \\ \mathbf{v}_{8i+1} \\ \mathbf{v}_{8i+2} \\ \mathbf{v}_{8i+3} \\ \mathbf{v}_{8i+4} \end{bmatrix} \quad (12)$$

or

$$\mathbf{X}^{\text{coh},i} = \mathcal{P}^{\text{coh},i} \mathbf{X} \quad \mathbf{x}^{\text{coh},i} = \mathcal{P}^{\text{coh},i} \mathbf{x} \quad \mathbf{v}^{\text{coh},i} = \mathcal{P}^{\text{coh},i} \mathbf{v} \quad (13)$$

where, once again, $\mathcal{P}^{\text{coh},i}$ are projection matrices.

Similar as for the subelements, the vectors in Eq. (12) together with the interface material properties are used to call a CZ element subroutine which calculates the internal force vector $\mathbf{f}^{\text{coh},i}$ of the current time step. To be specific, the CZ element subroutine is currently the same as the LS-DYNA solid element formulation 19 [24], an eight-noded element with four in-plane integration points. When the total cohesive force is obtained it is assembled to the element force vector as

$$\mathbf{f}^{\text{coh}} = \sum_{i \in I} \mathcal{P}^{\text{coh},i\text{T}} \mathbf{f}^{\text{coh},i}, \quad (14)$$

where I contains those refined interfaces which have CZ elements.

2.3. Adaptivity

So far, we have presented the user element connectivity and internal force calculation with respect to the different element stages shown in Fig. 1. In this section we will propose when and how to go from one stage to the next. In general we will follow the procedure presented in [7] with some modifications and additions. One of the major modifications has already been presented: the separation into a weak refinement stage and the subsequent strong refinement with CZ elements.

A reason for introducing weak refinements is that, although ESL elements are computationally efficient, the kinematic assumptions they are based on lead to low quality in the prediction of the out-of-plane stresses. By relaxing the kinematic assumptions the stress prediction can be improved. So, in the first refinement stage weak discontinuities are introduced through the thickness, which will transform the ESL element into a LW element (weak refinement). This way, better prediction of the out-of-plane stresses is achieved, and failure of individual plies can be modelled more accurately.

If delamination initiation and propagation are to be modelled there is a second refinement stage. In this stage, the weak discontinuities can be separated and CZ elements are inserted (strong refinement). This introduction of strong discontinuities will of course further relax the kinematic constraints.

Any of the adaptive elements in the model can trigger weak and strong refinements. If this occurs it will affect also the surrounding elements. So, if refinement is indicated by some element at a particular time, this will activate a refinement first in the following time step, in order to avoid any interference between elements. This way it is ensured that all elements are in the same time step when a refinement is activated.

2.3.1. Weak refinement

In this paper we are only considering elastic intralaminar behaviour. Then there is no benefit of introducing weak refinement in order to improve the intralaminar failure prediction. This is instead addressed in Part II [20]. In this paper, weak refinements will therefore only be used as stepping stones to strong refinements. Thus, the activations of both refinement types will be based on an interlaminar failure index, cf. Section 2.3.2. Nevertheless, implementation-wise a strong refinement is based on the structure of a weak refinement. For completeness we will therefore present the details of how the extra nodes are treated when they are activated during a weak refinement.

When an interface is to be weakly refined, a set of four extra nodes needs to be positioned to represent the interface. The reference position \mathbf{X} of these nodes will be given by Eq. (2), however, the current positions \mathbf{x} and velocities \mathbf{v} are unknown. We have chosen to assign these in a similar way as the reference positions, i.e. for interface i we interpolate \mathbf{x} and \mathbf{v} as

$$\begin{aligned}
\mathbf{x}_{8i+1} &= \mathbf{x}_{\text{below},1} + \hat{\alpha}_i (\mathbf{x}_{\text{above},1} - \mathbf{x}_{\text{below},1}) \\
\mathbf{x}_{8i+2} &= \mathbf{x}_{\text{below},2} + \hat{\alpha}_i (\mathbf{x}_{\text{above},2} - \mathbf{x}_{\text{below},2}) \\
\mathbf{x}_{8i+3} &= \mathbf{x}_{\text{below},3} + \hat{\alpha}_i (\mathbf{x}_{\text{above},3} - \mathbf{x}_{\text{below},3}) \\
\mathbf{x}_{8i+4} &= \mathbf{x}_{\text{below},4} + \hat{\alpha}_i (\mathbf{x}_{\text{above},4} - \mathbf{x}_{\text{below},4})
\end{aligned} \tag{15}$$

and

$$\begin{aligned}
\mathbf{v}_{8i+1} &= \mathbf{v}_{\text{below},1} + \hat{\alpha}_i (\mathbf{v}_{\text{above},1} - \mathbf{v}_{\text{below},1}) \\
\mathbf{v}_{8i+2} &= \mathbf{v}_{\text{below},2} + \hat{\alpha}_i (\mathbf{v}_{\text{above},2} - \mathbf{v}_{\text{below},2}) \\
\mathbf{v}_{8i+3} &= \mathbf{v}_{\text{below},3} + \hat{\alpha}_i (\mathbf{v}_{\text{above},3} - \mathbf{v}_{\text{below},3}) \\
\mathbf{v}_{8i+4} &= \mathbf{v}_{\text{below},4} + \hat{\alpha}_i (\mathbf{v}_{\text{above},4} - \mathbf{v}_{\text{below},4})
\end{aligned} \tag{16}$$

where \bullet_{below} and \bullet_{above} denotes the closest active nodes below and above the new interface at the time of activation. Please note that these nodes are not necessarily the base nodes, as is the case for the reference positions in Eq. (2). Therefore, the interpolation value $\hat{\alpha}_i$ is the relative position of the newly refined interface with respect to the nearest already active nodes, below and above. For example, if the third interface position in the right side of Fig. 3 is to be refined, the interpolation would be made from the second and fourth interfaces. This would give an interpolation value of

$$\hat{\alpha}_2 = \frac{3-2}{4-2} = \frac{1}{2},$$

where the index $i = 2$ indicates that this will be the second refinement from the bottom.

After activation, the nodes are treated as all other nodes in the model, *i.e.* their velocities and positions are updated using the explicit time integration scheme.

2.3.2. Strong refinement

In order to predict delaminations correctly, primarily two sources for these must be captured: ply cracks and high out-of-plane stresses. The former, which involves the interaction of intra- and interlaminar cracks, will be treated in Part II [20]. Instead focusing on the latter, interlaminar stresses in the laminate are extracted and used to evaluate an interlaminar failure initiation criterion

$$\hat{f}_{CZ} = \left(\frac{\langle \sigma_{33} \rangle}{T_{CZ}} \right)^2 + \left(\frac{\sigma_{13}}{S_{CZ}} \right)^2 + \left(\frac{\sigma_{23}}{S_{CZ}} \right)^2, \tag{17}$$

where $\langle \bullet \rangle$ are the Macauley brackets, σ_{ij} are the interlaminar out-of-plane Cauchy stresses and T_{CZ} and S_{CZ} are the interlaminar normal and shear strengths, respectively. If \hat{f}_{CZ} reaches a limit $f_{CZ,\text{lim}}$, *i.e.* if

$$\hat{f}_{CZ} \geq f_{CZ,\text{lim}} \tag{18}$$

the interface is completely separated into two surfaces connected via a CZ element. To avoid jumps in the stress state, the CZ elements should be introduced before interlaminar failure is initiated, *i.e.* $f_{CZ,\text{lim}}$ should always be chosen less than one.

The interlaminar failure initiation criterion in (Eq. (17)) is evaluated using the out-of-plane stresses predicted by the unrefined (or weakly refined) element. Unfortunately, the prediction of the out-of-plane stresses is of low quality in ESL shells. In [7] we showed that a stress recovery technique can be used to improve the prediction of the stress state (of the ESL shell) and thus also of when to introduce CZ elements. Therefore, instead of using the out-of-plane stresses calculated from the shell kinematics, Eq. (17) can preferably be evaluated using the recovered stress state. However, in the numerical examples below there is little benefit of using a stress recovery. This feature has therefore not yet been added in the LS-DYNA implementation.

When the criterion in Eq. (18) is met the weak refinement interface is duplicated by copying the current positions \mathbf{x} and velocities \mathbf{v} from the already activated set of extra nodes to the reserved set. That is, for an interface i at the time of strong refinement this becomes

$$\begin{aligned}
\mathbf{x}_{8i+1} &= \mathbf{x}_{8(i-1)+5} \\
\mathbf{x}_{8i+2} &= \mathbf{x}_{8(i-1)+6} \\
\mathbf{x}_{8i+3} &= \mathbf{x}_{8(i-1)+7} \\
\mathbf{x}_{8i+4} &= \mathbf{x}_{8(i-1)+8}
\end{aligned} \tag{19}$$

and

$$\begin{aligned}
\mathbf{v}_{8i+1} &= \mathbf{v}_{8(i-1)+5} \\
\mathbf{v}_{8i+2} &= \mathbf{v}_{8(i-1)+6} \\
\mathbf{v}_{8i+3} &= \mathbf{v}_{8(i-1)+7} \\
\mathbf{v}_{8i+4} &= \mathbf{v}_{8(i-1)+8}
\end{aligned} \tag{20}$$

The interface is now defined by eight nodes and a CZ element can be inserted in order to model delamination initiation and propagation.

Ideally, the strong refinement criterion in Eq. (18) will introduce CZ elements at already weakly refined interfaces. However, in cases when delaminations are prone to initiate without any preceding intralaminar damage, the adaptive scheme must also handle

the transition into strong refinements directly. In those cases, the criterion in Eq. (18) will activate first a weak refinement followed by a strong refinement in the same time step.

2.4. Challenges related to using an explicit dynamic solver

The introduction of weak and strong refinements can result in non-physical oscillations (in the sense that they are not a result of the problem being solved and would not be present in a model which is highly refined from the start). Since a explicit dynamic solver is used these non-physical oscillations will remain and can lead to an unstable solution. To prevent this problem, we have implemented two types of damping procedures which will be presented below.

2.4.1. Instability from weak refinement

The source of the non-physical oscillations during weak refinement is that the interpolated positions (and velocities) in Eq. (15) (and (16)) are possibly not resulting in balance of linear momentum. The proposed interpolation is only an estimation, which leads to that newly introduced nodes will start to move towards their balance positions. This problem is especially present in a case where the element consists of a laminate of orthotropic plies with different orientations, where the stiffness of the subelements can be very different to each other and compared to that of the parent element.

There has been a lot of research conducted in the field of adaptive FE methods where element are refined in an explicit dynamic solver. For example, there is already an h-adaptive in-plane refinement technique for shells in LS-DYNA, based on the work by Belytschko, Wong and Plaskacz [25]. Neither [25] nor the review on adaptive FE methods by Li and Bettess [26] mention any instability due to refinements. However, it should be mentioned that the LS-DYNA adaptive technique makes a restart (and optionally a repetition of time steps) every time the model is refined. This is in contrast to our refinement method, where the refinements are made without making a restart of the simulation.

To the best of our knowledge there have been little or no work on integrated h-adaptive remeshing techniques in explicit solvers that can refine the model without restarting the simulation. Therefore, it is difficult to find any solutions in the literature to the instability encountered from the refinement of elements.

In order to stabilise the solution, the refinement oscillations are removed by applying a small amount of damping for a short time period directly after the refinement instant:

$$C = \begin{cases} C_{d,ref} & \text{if } t_{ref} \leq t \leq t_{ref} + \Delta t_{ref}, \\ 0 & \text{else} \end{cases}, \quad (21)$$

where $C_{d,ref}$ is the damping value, t_{ref} is the time of refinement and Δt_{ref} is the duration of the damping, see Fig. 5 (left).

In order to determine the level of damping $C_{d,ref}$ and the duration Δt_{ref} the unwanted oscillation is analysed. By finding the oscillation Ω_0 (defined as the peak-to-peak amplitude) and the dominating angular frequency ω_n a suitable damping can be applied. Since only a small amount of damping is to be applied, the system will be underdamped and the oscillation Ω should follow an exponential decay according to

$$\Omega = \Omega_0 e^{-\zeta \omega_n \Delta t_{ref}}, \quad (22)$$

where ζ is the damping ratio. If a certain level of oscillations $\bar{\Omega}$ is acceptable the damping duration is given by how strong damping one wants to allow:

$$\Delta t_{ref} = \frac{1}{\zeta \omega_n} \ln \left(\frac{\Omega_0}{\bar{\Omega}} \right). \quad (23)$$

We will elaborate on this in the numerical examples below.

Remark: In the current implementation we are using the LS-DYNA mass-weighted damping DAMPING_GLOBAL [24]. When active, this damps the entire model including rigid body motions. Therefore, we propose not to use a damping larger than 10% of the critical damping of the model.

2.4.2. Instability from strong refinement

When a strong refinement is activated the interface of the weak refinement is duplicated and a CZ element is inserted. This will cause abrupt changes in internal forces, which can result in non-physical oscillations. Instead of applying a damping force to stabilise the solution also in this case, we follow a method similar to that proposed by Menouillard and Belytschko [27] and apply a correction force to balance the sudden change in internal forces. The purpose of the correction force is to tie the node pairs of the two coincidental interfaces together, i.e. the separate node pairs shall move as they were one node in each time step:

$$\mathbf{x}_I = \mathbf{x}_{II} \quad \mathbf{v}_I = \mathbf{v}_{II} \quad \mathbf{a}_I = \mathbf{a}_{II} \quad (24)$$

where \bullet_I and \bullet_{II} refers to the node associated to the subelement below and above the interface, respectively. The enforcement of Eq. (24) will lead to a specific force assembly for the interface nodes which is illustrated in Fig. 4. The details of this will be described next.

As the nodal accelerations \mathbf{a} are updated using

$$\mathbf{M}\mathbf{a} = \mathbf{P} - \mathbf{f} - \mathbf{C}\mathbf{v}, \quad (25)$$

where \mathbf{M} is the mass matrix, \mathbf{P} is the external forces and \mathbf{C} is the damping matrix, the constraints in Eq. (24) will lead to a mass-relative scaling of the internal force assembly as

$$\begin{aligned} \mathbf{f}_I &= \frac{m_I}{m_I + m_{II}} \mathbf{f}_{I+II} + \frac{1}{m_I + m_{II}} (m_{II} \mathbf{P}_I - m_I \mathbf{P}_{II}) + \frac{1}{m_I + m_{II}} (m_I \mathbf{C}_{II} - m_{II} \mathbf{C}_I) \mathbf{v}, \\ \mathbf{f}_{II} &= \frac{m_{II}}{m_I + m_{II}} \mathbf{f}_{I+II} + \frac{1}{m_I + m_{II}} (m_I \mathbf{P}_{II} - m_{II} \mathbf{P}_I) + \frac{1}{m_I + m_{II}} (m_{II} \mathbf{C}_I - m_I \mathbf{C}_{II}) \mathbf{v}, \end{aligned} \tag{26}$$

where

$$\mathbf{f}_{I+II} = \mathbf{f}_I^{\text{sublam-}} + \mathbf{f}_{II}^{\text{sublam+}} \tag{27}$$

is the sum of nodal pair forces from the subelement below and above, and m is the nodal mass. By assuming a diagonal mass matrix, no external forces on the internal nodes of the element and mass proportional damping, Eq. (26) simplifies to

$$\mathbf{f}_I = \frac{m_I}{m_I + m_{II}} \mathbf{f}_{I+II} = c_I \mathbf{f}_{I+II} \tag{28}$$

$$\mathbf{f}_{II} = \frac{m_{II}}{m_I + m_{II}} \mathbf{f}_{I+II} = c_{II} \mathbf{f}_{I+II}. \tag{29}$$

We therefore define an interface force vector $\mathbf{f}^{\text{tied},i}$, which contains the force contribution for the tied node pairs for an interface i . This vector will have a force contribution from subelement i below and subelement $i + 1$ above as

$$\mathbf{f}^{\text{tied},i} = \begin{bmatrix} (\mathbf{f}_5^{\text{sublam},i} + \mathbf{f}_1^{\text{sublam},i+1}) c_1^i \\ (\mathbf{f}_6^{\text{sublam},i} + \mathbf{f}_2^{\text{sublam},i+1}) c_2^i \\ (\mathbf{f}_7^{\text{sublam},i} + \mathbf{f}_3^{\text{sublam},i+1}) c_3^i \\ (\mathbf{f}_8^{\text{sublam},i} + \mathbf{f}_4^{\text{sublam},i+1}) c_4^i \\ (\mathbf{f}_5^{\text{sublam},i} + \mathbf{f}_1^{\text{sublam},i+1})(1 - c_1^i) \\ (\mathbf{f}_6^{\text{sublam},i} + \mathbf{f}_2^{\text{sublam},i+1})(1 - c_2^i) \\ (\mathbf{f}_7^{\text{sublam},i} + \mathbf{f}_3^{\text{sublam},i+1})(1 - c_3^i) \\ (\mathbf{f}_8^{\text{sublam},i} + \mathbf{f}_4^{\text{sublam},i+1})(1 - c_4^i) \end{bmatrix}, \tag{30}$$

where

$$\begin{aligned} c_1^i &= \frac{m_{8(i-1)+5}}{m_{8(i-1)+5} + m_{8i+1}}, & c_2^i &= \frac{m_{8(i-1)+6}}{m_{8(i-1)+6} + m_{8i+2}}, \\ c_3^i &= \frac{m_{8(i-1)+7}}{m_{8(i-1)+7} + m_{8i+3}}, & c_4^i &= \frac{m_{8(i-1)+8}}{m_{8(i-1)+8} + m_{8i+4}}. \end{aligned}$$

In Eq. (30) the subindices on $\mathbf{f}^{\text{sublam},i}$ and $\mathbf{f}^{\text{sublam},i+1}$ represent the local node numbering for each of the subelements. The consequence of Eq. (30) is that the force from one subelement will be distributed not only to its vertex nodes, but also to the nodes on the other side of a tied interface.

Similarly as $\mathbf{f}^{\text{tied},i}$ we can define an interface force vector for the case when the interface is released to two separated surfaces as

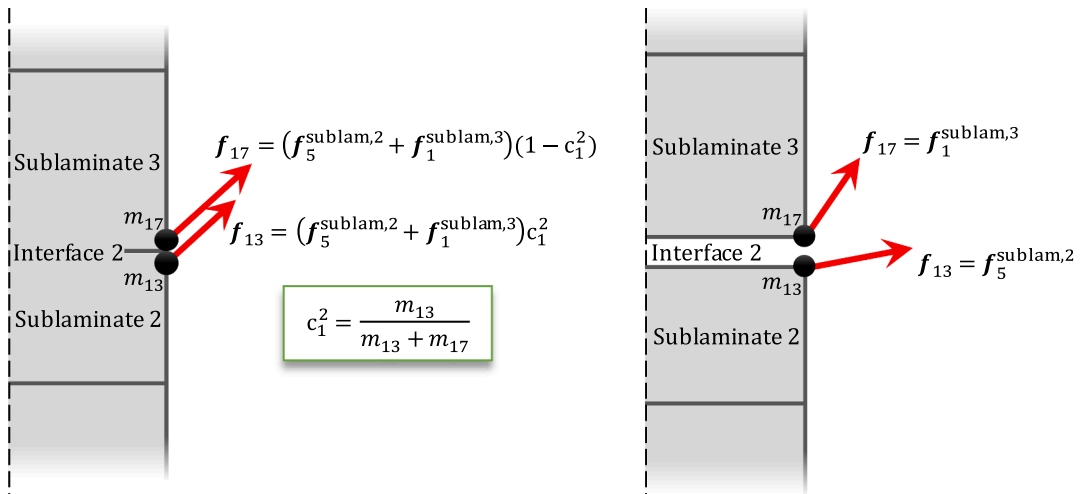


Fig. 4. Example of how the internal force from the subelements are assembled in the case the nodes are tied together (left) or released such that the interface surfaces are separated (right). In this example it can be observed that $m_{13} < m_{17}$, which makes $\|\mathbf{f}_{13}\| < \|\mathbf{f}_{17}\|$. Please note that the tied nodes coincide but are drawn separately for clarity.

$$\mathbf{f}^{\text{released},i} = \begin{bmatrix} \mathbf{f}_5^{\text{sublam},i} \\ \mathbf{f}_6^{\text{sublam},i} \\ \mathbf{f}_7^{\text{sublam},i} \\ \mathbf{f}_8^{\text{sublam},i} \\ \mathbf{f}_1^{\text{sublam},i+1} \\ \mathbf{f}_2^{\text{sublam},i+1} \\ \mathbf{f}_3^{\text{sublam},i+1} \\ \mathbf{f}_4^{\text{sublam},i+1} \end{bmatrix}. \quad (31)$$

Following [27], the release of interface i is then made gradually by shifting from the tied force $\mathbf{f}^{\text{tied},i}$ to the released force $\mathbf{f}^{\text{released},i}$ as

$$\mathbf{f}^{\text{lam}} = \mathbf{f}^{\text{lam}} + \mathcal{P}^{\text{coh},i\text{T}} \left((1 - \kappa_i) \mathbf{f}^{\text{tied},i} + \kappa_i \mathbf{f}^{\text{released},i} \right), \quad (32)$$

where we have used the projection matrix of the CZ element $\mathcal{P}^{\text{coh},i}$ in Eq. (13) to assemble to the parent element force vector. Please note $\mathbf{f}^{\text{released},i}$ is not part of the normal force assembly in Eq. (11) during the separation phase of the interface.

The weight κ_i above is chosen as

$$\kappa_i = \begin{cases} 0 & \text{if } t_{\text{CZ}} < t \\ \frac{1}{2} \left[1 - \cos \left(\frac{\pi(t - t_{\text{CZ}})}{\Delta t_{\text{CZ}}} \right) \right] & \text{if } t_{\text{CZ}} \leq t \leq t_{\text{CZ}} + \Delta t_{\text{CZ}}, \\ 1 & \text{if } t_{\text{CZ}} + \Delta t_{\text{CZ}} < t \end{cases} \quad (33)$$

where t_{CZ} is the time of CZ element introduction (strong refinement) and Δt_{CZ} is the duration of the transition, see Fig. 5 (right). This way the CZ element is introduced smoothly. Like the damping in Eq. (21) this transition is made over a very short period of time.

2.5. Nodal mass

In order for LS-DYNA to perform the time integration, the nodal masses must be defined. During the initiation phase LS-DYNA will assemble mass to the base nodes (marked \diamond in Eq. (1)) by summing the contribution from all connected element. Each element will contribute with a mass based on the material density ρ and the reference volume V of the element

$$m_k = \frac{\rho V}{8} \quad k = 1 \dots 8. \quad (34)$$

Since the extra nodes are not active during the initiation phase they are not assigned any mass, i.e. $m = 0$ for these nodes. However, if a refinement is activated the mass of both the base and extra nodes must be updated to accurately represent the subelements they are defining.

2.5.1. Weak refinement

In the case of a weak refinement the mass of the parent element is subtracted from its eight base nodes. After that the appropriate masses related to the new subelements are added to the 16 nodes now active:

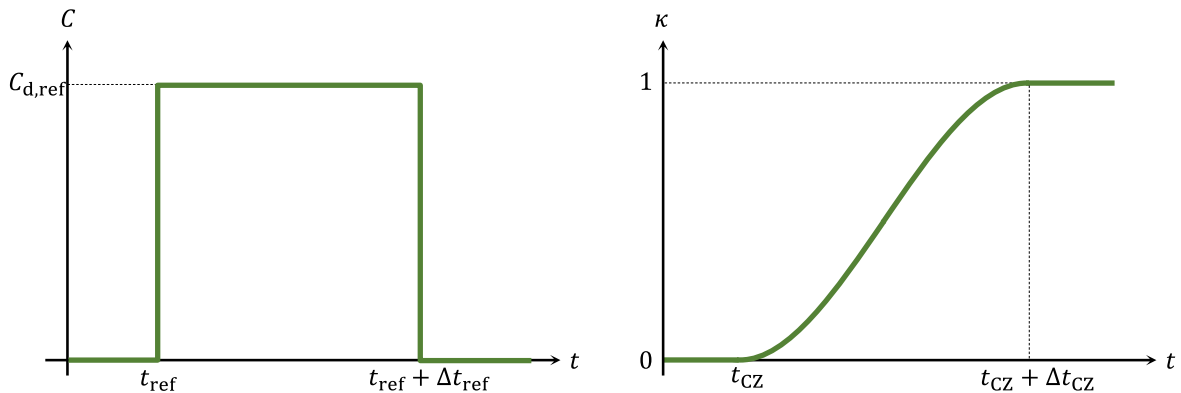


Fig. 5. Damping function employed to remove the weak refinement oscillations (left). Weight function to gradually release the node pairs of the strong refinement (right).

$$m_k = \begin{cases} m_k - \frac{\rho V_{old}}{8} + \frac{\rho V_{sub,1}}{8} & \text{if } k = 1...4 \\ m_k + \frac{\rho V_{old}}{8} & \text{if } k = 5...8 \\ m_k & \text{if } k = 9...12 \\ m_k - \frac{\rho V_{old}}{8} + \frac{\rho V_{sub,2}}{8} & \text{if } k = 13...16 \end{cases}, \quad (35)$$

where $V_{sub,1}$ is the volume of the bottom subelement, $V_{sub,2}$ that of the upper subelement and $V_{old} = V_{sub,1} + V_{sub,2}$ is the volume of the parent element. In Eq. (35) the masses of the reserved set of extra nodes (marked in gray), which will be used if the interface is to be strongly refined, are not modified at the moment. Please note that node 1–4 and 13–16 are the nodes of the unrefined element and had a mass assigned using Eq. (34) while nodes 5–12 had no initially assigned mass from the element (but possibly from the element neighbours). If a subelement is refined the procedure in Eq. (35) is simply repeated, but for new node numbers. For example, if the upper subelement is refined, the nodal masses associated with the two new subelements are modified as

$$m_k = \begin{cases} m_k - \frac{\rho V_{old}}{8} + \frac{\rho V_{sub,2}}{8} & \text{if } k = 5...8 \\ m_k + \frac{\rho V_{old}}{8} & \text{if } k = 13...16 \\ m_k & \text{if } k = 17...20 \\ m_k - \frac{\rho V_{old}}{8} + \frac{\rho V_{sub,3}}{8} & \text{if } k = 21...24 \end{cases}, \quad (36)$$

where $V_{sub,2}$ is the volume of the new middle subelement, $V_{sub,3}$ that of the new upper subelement and $V_{old} = V_{sub,2} + V_{sub,3}$ is the volume of the old upper subelement. The numbers of the first four cases in Eq. (36) ($k = 5...8$) are dependent on whether the interface below is weakly or strongly refined. In this example, SS the interface is weakly refined and the nodal numbers are 5 to 8. If the interface below would have been strongly refined, eight nodes would be active for the interface and the nodal numbers would range from 9 to 12 instead. For further subelement divisions this procedure is then repeated.

2.5.2. Strong refinement

When a strong refinement is activated at an interface it is always preceded by a weak refinement, which have four active nodes and four reserved nodes. This is the case even if the weak and the strong refinements are made in the same time step. The reserved set is activated and the nodal masses of the interface is adjusted by distributing the nodal masses of the four weak refinement nodes to the eight nodes now active in the interface. For example, if the first interface in the element with two weak refinements (in the Subsection above) is strongly refined the masses are adjusted as

$$m_k = \begin{cases} m_k & \text{if } k = 1...4 \\ m_k - \frac{\rho V_{old}}{8} + \frac{\rho V_{sub,1}}{8} + \frac{\rho_{CZ} A_{CZ,1}}{8} & \text{if } k = 5...8 \\ m_k + \frac{\rho V_{sub,2}}{8} + \frac{\rho_{CZ} A_{CZ,1}}{8} & \text{if } k = 9...12 \\ m_k & \text{if } k = 13...16 \end{cases}, \quad (37)$$

where $V_{old} = V_{sub,1} + V_{sub,2}$, ρ_{CZ} is the cohesive area density and $A_{CZ,1}$ is the interface area.⁴ Furthermore, if the second interface is strongly refined, the masses are adjusted as

$$m_k = \begin{cases} m_k & \text{if } k = 9...12 \\ m_k - \frac{\rho V_{old}}{8} + \frac{\rho V_{sub,2}}{8} + \frac{\rho_{CZ} A_{CZ,2}}{8} & \text{if } k = 13...16 \\ m_k + \frac{\rho V_{sub,3}}{8} + \frac{\rho_{CZ} A_{CZ,2}}{8} & \text{if } k = 17...20 \\ m_k & \text{if } k = 21...24 \end{cases} \quad (38)$$

and so on.

3. Results

We will in this section present some numerical examples to verify that we can replicate the same results as a reference model (with standard LS-DYNA elements). Furthermore, we will study the values of the adaptive parameters associated with the weak and strong refinements, $C_{d,ref}$, Δt_{ref} , $\hat{f}_{CZ,lim}$ and Δt_{CZ} . By measuring the computational time for an unrefined and a refined user element we will calculate the computational efficiency of the adaptive approach. However, since our user element is not optimised for speed this efficiency will only be an estimate and should be considered as a lower bound.

In all the examples below, material model 138 MAT_COHESIVE_MIXED_MODE [28] is used for the cohesive elements (user element and reference). Likewise, for all models the intralaminar material model is equivalent to LS-DYNA material model 2, i.e. an elastic orthotropic material formulated in terms of Piola-Kirchhoff stress and Green-Lagrange strain [28].

⁴ All interfaces are modelled with zero initial thickness.

3.1. Element with one delamination

Our first example is a 1x1x1 mm element which is loaded in the top nodes with a prescribed acceleration of $\ddot{\delta} = 0.030\text{m/s}^2$ for a duration of 1 ms. The layout is $[-45, 90, 0, 45]$ and a delamination is expected to be able to form in the (middle) 90/0 interface, which is where the element will be refined. The placement of the refinement is controlled by only allowing refinements in interfaces where the pitch angle between the plies $\Delta\theta$ is larger than 45° . In order to verify that the user element can reproduce different levels of refinement, we compare it to three different references.

The first reference consists of one element through the thickness representing the entire laminate (no refinement); the second is made of two sublaminates elements sharing nodes in the 90/0 interface corresponding to a weak refinement (refined w/o CZ) and the last reference model is two sublaminates elements connected by a CZ element in the 90/0 interface corresponding to a strong refinement (refined with CZ). The reference models are based on standard LS-DYNA elements, corresponding to the same type as the user element internal subelements. That is, solid shell formulation 3 for the laminate plies and solid element formulation 19 for the CZ interface. The problem and the reference cases are illustrated in Fig. 6, where we have also indicated the positions of the nodes we will sample the displacements from.

The values of the material parameters can be found in Table 1 and 2. Specific options for the element and material models, e.g. AOPT = 3 and INTFAIL = 4, are the same in the user element and the references.

3.1.1. Non-adaptive verification

First, the example is run *non-adaptively*, i.e. with the 0/90 interface in the user element (strongly) refined with a CZ element from the start. The nodal displacements and reaction forces in Fig. 7 and 8 show that the user element exactly reproduces the results from the reference model with a CZ element (Ref 3 in Fig. 6).

3.1.2. Weak refinement

To determine the adaptive parameters we first focus on weak refinement of the 90/0 interface. In Fig. 9 we have plotted the CZ failure initiation criterion \hat{f}_{CZ} in Eq. (17) evaluated using the average intralaminar stresses of the adjacent 90 and 0 plies in an unrefined and weakly refined element. These are compared to the actual failure initiation criterion f_{CZ} of the CZ element in the reference solution (Ref 3). We observe that the unrefined and refined elements yield 15% and 20% lower values compared to the CZ element. This should be taken into account if using the intralaminar stresses to evaluate interlaminar failure initiation.

However, this is not the focus of this example. Instead, we ran a series of (unstablised) adaptive simulation with weak refinements times corresponding to different f_{CZ} levels of the reference CZ element. Fig. 10 shows an example of the resulting displacement when refining at $f_{CZ} = 0.4$, corresponding to a weak refinement time of $t_{ref} = 0.38$ ms according to Fig. 9. When the weak refinement is activated the extra nodes will move to the interpolated positions following Eq. (15) and oscillations around their balance positions follow.

In order to quantify the error, we calculate the displacement difference Δu between the user element and the reference case (refined w/o CZ). The oscillation Ω_0 of the displacement difference, defined as

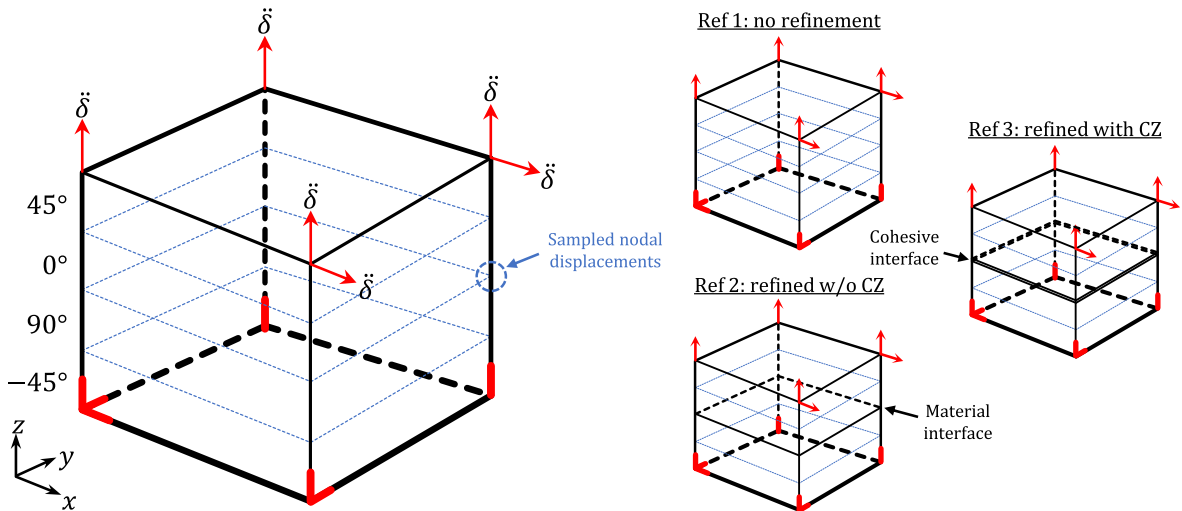


Fig. 6. Illustration of one-element numerical example with one delamination. The model is constrained in the bottom (thick red marks) and loaded by prescribed acceleration $\ddot{\delta}$ at the top. A strong refinement with CZ element will be introduced in the 90/0 interface and the displacements of a node on the bottom and the top of the interface are extracted. The user element is compared to three reference models of standard LS-DYNA elements: One element representing the entire laminate (no refinement); Two sublaminates elements sharing nodes (refined w/o CZ) and two sublaminates elements connected by a CZ element (refined with CZ). The 0° orientation corresponds to the x-axis.

Table 1
Ply material properties used in the one-element examples.

E_1 (GPa)	E_2, E_3 (GPa)	G_{12}, G_{13} (GPa)	G_{23} (GPa)	ν_{21}, ν_{31} (-)	ν_{32} (-)	ρ (kg/m ³)
136	9.15	4.9	3.27	0.0195	0.4	1600

Table 2
Interface material properties used in the one-element examples.

σ_I (MPa)	\mathcal{G}_{Ic} (J/m ²)	k_I (GPa/mm)	σ_{II} (MPa)	\mathcal{G}_{IIc} (J/m ²)	k_{II} (GPa/mm)	ρ_{CZ} (kg/m ²)
40	222	1000	60	1920	1000	0.0

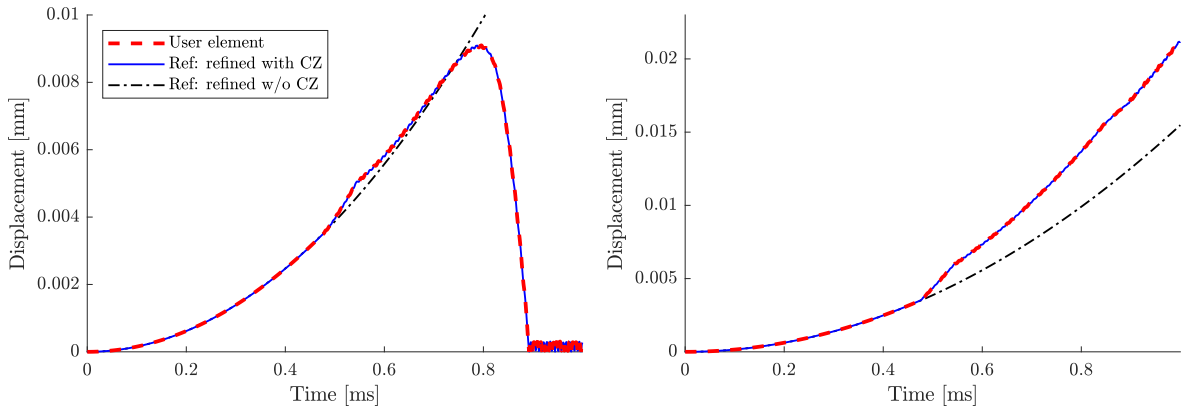


Fig. 7. *Non-adaptive simulation:* Resulting displacement of one node in the bottom (left) and in the top (right) of the middle interface (cf. Fig. 6). Up to the point of damage initiation in the CZ element, all three models behave similarly. After initiation, a delamination is formed in the user element and the reference with CZ element, whose displacement curves match exactly.

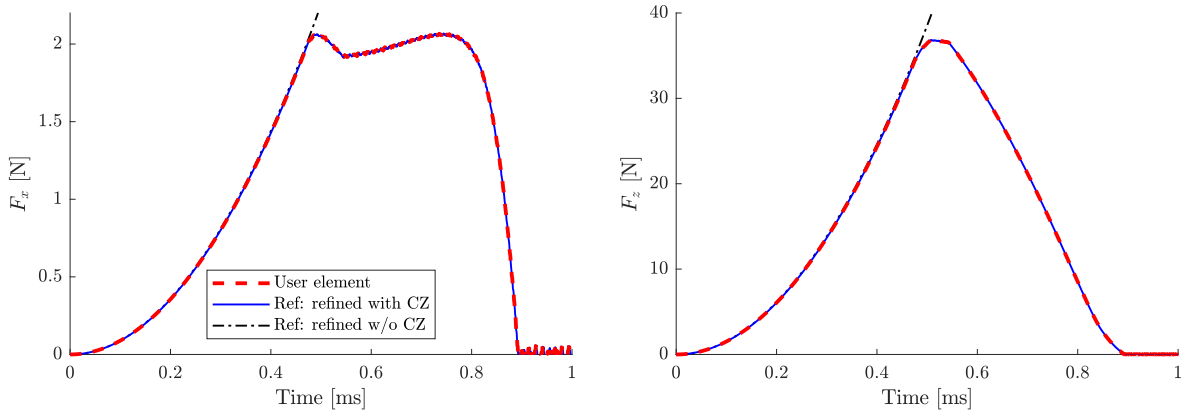


Fig. 8. *Non-adaptive simulation:* Total reaction force in x (left) and z (right) direction in the bottom nodes. The user element and the reference with CZ match exactly.

$$\Omega_0 = \sup_{t \in [\tau_1, \tau_2]} \Delta u(t) - \inf_{t \in [\tau_1, \tau_2]} \Delta u(t), \tag{39}$$

and the smallest (and dominating) oscillating angular frequency ω_n are measured over a time period $[\tau_1, \tau_2]$ after the refinement activation.

Evaluating over a time period of [0.5,0.8] ms, the oscillations Ω_0 grow slightly with increasing f_{CZ} , as shown in Fig. 11. Fig. 11 also shows that the mean of the oscillations $\Delta \bar{u}$ is not zero. Instead, it has an offset to the reference case and, like the oscillations, increase the later the refinement is activated. That means that, if filtering the displacement of the adaptive simulations it is not exactly the same as the reference case. However, the angular frequency ω_n is almost constant at 915 rad/ms regardless of refinement time.

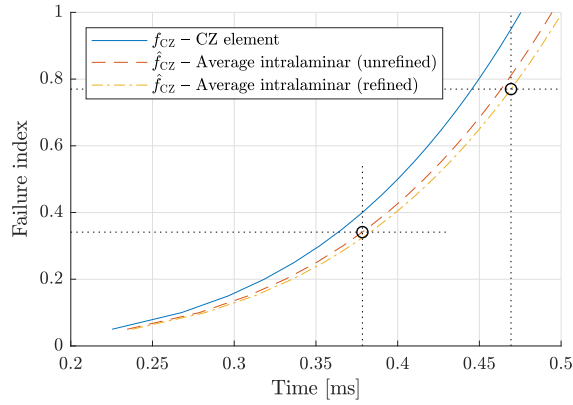


Fig. 9. Interlaminar failure initiation criterion \hat{f}_{CZ} in Eq. (17) evaluated using the average intralaminar stresses of the adjacent 90 and 0 plies in an unrefined and weakly refined element. Compared to the failure initiation criterion f_{CZ} of the CZ element in the reference solution. The refinement activation times used in the fully adaptive simulation are also marked.

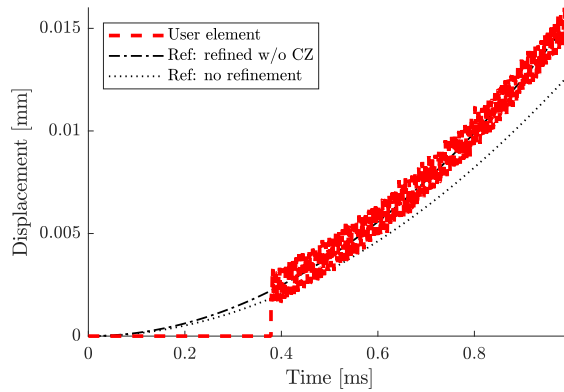


Fig. 10. Resulting displacement of one node in the bottom of the middle interface when weakly refining at $f_{CZ} = 0.4$ ($t_{ref} = 0.38$ ms). Before the refinement is activated the extra nodes have zero displacement. At activation they are placed at the interpolated middle position, i.e. the same as the unrefined reference. However, their true positions should be the same as the refined reference. This error in the activation positions causes oscillations.

In order to remove the oscillations damping is applied following the procedure in Section 2.3.1. Based on the smallest (dominating) angular frequency $\omega_n = 915$ rad/ms, the critical damping value used is

$$C_{crit} = 2\omega_n \approx 1830 \text{ rad/ms.}$$

In Fig. 12 the oscillations are plotted versus the duration of damping Δt_{ref} for a series of simulations with different refinement

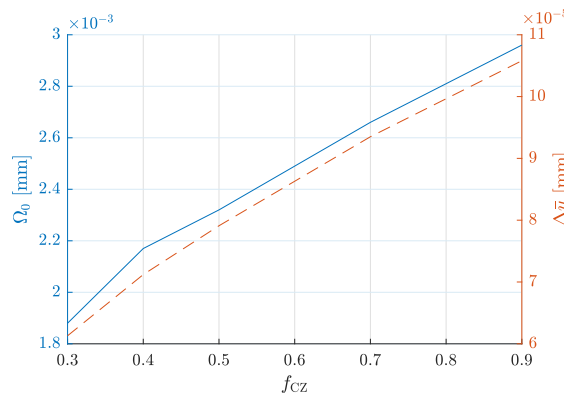


Fig. 11. Oscillation level Ω_0 of the displacement difference Δu (cf. Eq. (39)) and displacement offset $\Delta \bar{u}$ between adaptive simulations (filtered using SAE10) and reference case when refining at different f_{CZ} .

times and damping ratios. The results are compared to the decay expression in Eq. (22), where Ω_0 is chosen as the average oscillation of the undamped simulations in Fig. 11. Fig. 12 shows that the damping yields an expected exponential decay of the oscillations, and that the results are insensitive to the refinement time when damping is applied.

3.1.3. Strong refinement

When activating a strong refinement this can be done smoothly over a small time period Δt_{CZ} as described in Section 2.4.2. In Fig. 13 the remaining oscillations in the sampled nodes are plotted for different refinement times (f_{CZ} levels) and separation period Δt_{CZ} . First of all there is a reduction of the oscillations when applying the gradual separation over just one time step ($\Delta t_{CZ} = \Delta t$) and in the range of 0.001 to 0.018 ms separation period there is an exponential decay. However, using longer separation periods show little improvement. If we assume that the dominating frequency of the oscillations after CZ introduction is the same as that after refinement, the gradual separation corresponds to a damping ratio of approximately 40%.

Furthermore, Fig. 13 shows that there is little difference between different refinement times but also that it is important to introduce the CZ element so the separation does not interfere with the failure initiation of the CZ element. This means that the CZ element should be introduced such that the separation period is over before damage starts to evolve. In this example this would be for f_{CZ} values over 0.90, where there is a very short time before the CZ element will start to fail.

3.1.4. Fully adaptive simulation

We will here present the results from a (fully) adaptive simulation of the element with one delamination. The weak refinement was (arbitrarily) chosen to be activated when $\hat{f}_{CZ} = 0.34$ (corresponding to $t_{ref} = 0.378$ ms and $f_{CZ} = 0.40$, cf. Fig. 9) and the strong refinement when $\hat{f}_{CZ} = f_{CZ,lim} = 0.77$ (corresponding to $t_{CZ} = 0.47$ ms and $f_{CZ} = 0.95$, cf. Fig. 9). Based on the parametric study above the simulation was run using the parameter values in Table 3. The motivation is as follows:

- A refinement damping ratio of $\zeta = 0.10$ ($C_{d,ref} = 183$ rad/ms) was applied for $\Delta t_{ref} = 0.12$ ms, according to Eq. (23) for remaining oscillations of $\bar{\Omega} = 0.5 \times 10^{-7}$ mm and initial oscillation of $\Omega_0 = 0.0019$ mm (cf. Fig. 9).
- The strong refinement was introduced over $\Delta t_{CZ} = 0.01$ ms, which theoretically should result in remaining oscillations of $\bar{\Omega} < 10^{-7}$ mm according to Fig. 13.

The resulting nodal displacements and the difference between the user element and the reference (refined with CZ) are plotted in Fig. 14 and 15. These show that after the refinements and stabilisation have passed, there is a small difference between the user element and the reference. In Fig. 16 and 17 the reaction forces and the force differences are plotted. Compared to the nodal displacements, the reaction forces of the user element matches the reference better. Once the refinement phase has passed, both the absolute and the relative errors are very small. The increased errors at the end of the simulation occur when the CZ element is fully failed. At this stage, however, a reasonable comparison is difficult to make since the top subelement floats freely in the y direction.

The chosen damping duration of $\Delta t_{ref} = 0.12$ ms means that the damping is still active when the strong refinement is introduced and the CZ element starts to fail. This is not a problem. On the contrary the refinement damping can help to reduce the oscillations from the CZ introduction. The refinement can actually be simultaneous with the CZ element introduction.

To show this we present the results from a simulation where the weak and the strong refinements were activated simultaneously at $t_{CZ} = 0.47$ ms (corresponding to $\hat{f}_{CZ} = 0.81$ in the unrefined element, cf. Fig. 9). The results in Fig. 18 to 19 are (after oscillations have decayed) hard to distinguish from those when the weak refinement was made earlier (Fig. 14–17).

In this adaptive example, the average computational cycle time for the unrefined user element is 0.145 ms, while the (strongly) refined user element has a cycle time of 0.163 ms. This results in a computational save of 11% when the extra nodes are not active.

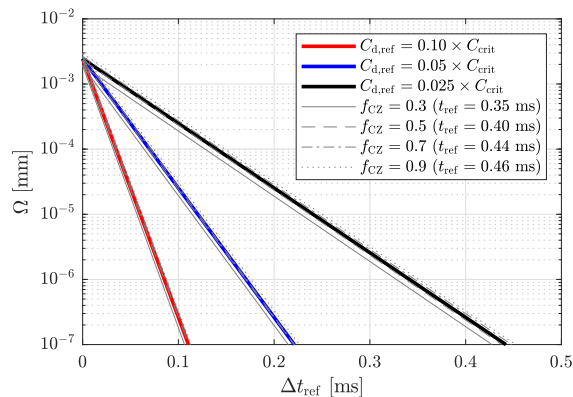


Fig. 12. Remaining oscillations for different damping ratios and CZ failure initiation values. The thick curves represent the theoretical decay in Eq. (22) and behind each line the corresponding simulations for each damping level is plotted (please note that each set has the same line properties). The theoretical decay matches the simulations results very well.

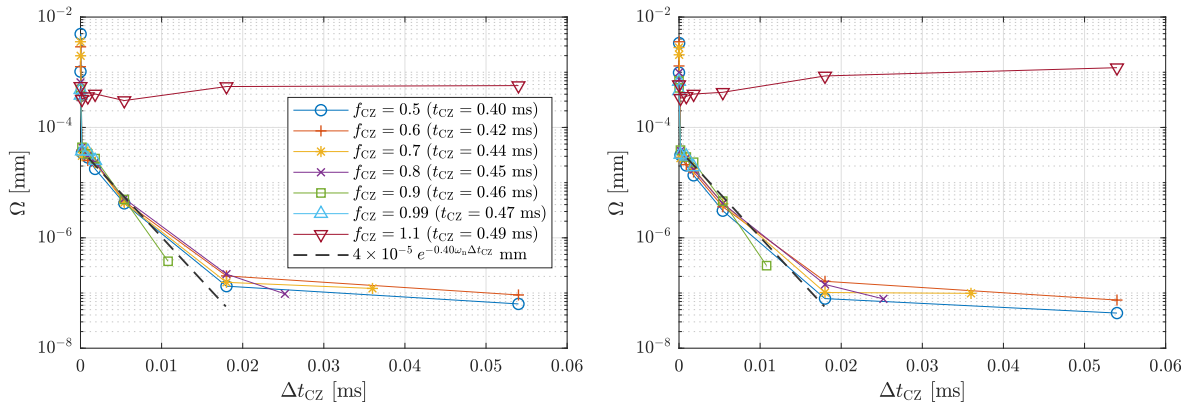


Fig. 13. Remaining oscillations in one node in the bottom (left) and in the top (right) of the middle interface for different strong refinement times and separation periods. A curve representing an exponential decay with damping ratio of 40% has also been added for comparison. Please note that f_{CZ} above 0.90 are very close to the failure initiation of the CZ elements meaning that only short separation periods can be employed.

Table 3
Values of adaptive parameters.

Parameter	Value
$\Delta\theta_{del}$	50°
t_{ref}	0.378 ms ($f_{CZ} = 0.31$)
$C_{d,ref}$	183 rad/ms ($\zeta = 0.10$)
Δt_{ref}	0.12 ms
$f_{CZ,lim}$	0.77 ($t_{CZ} = 0.47$ ms)
Δt_{CZ}	0.01 ms

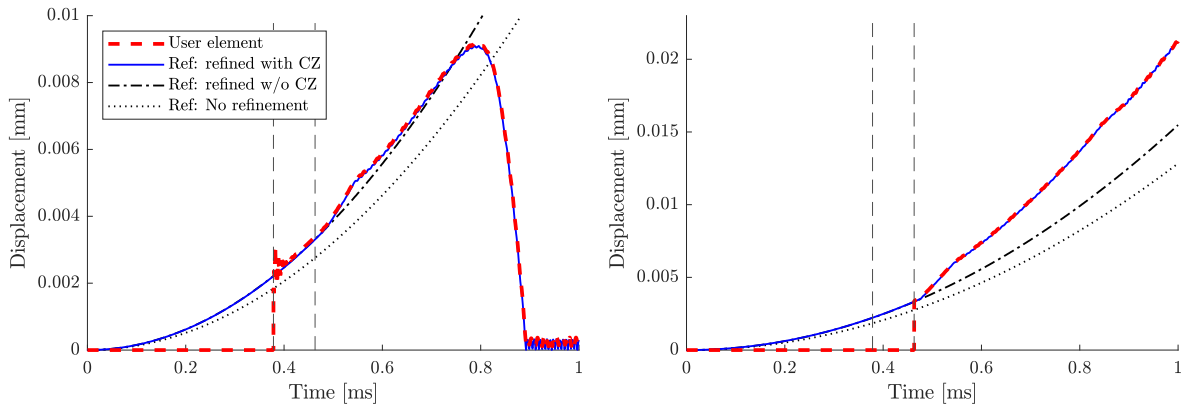


Fig. 14. Adaptive simulation: Resulting displacement of one node in the bottom (left) and in the top (right) of the middle interface. Before the weak refinement is activated the extra nodes have zero displacement. At activation (first dashed vertical line) they are placed at the interpolated middle position, i.e. the same as the unrefined reference. Damping is applied to reduce the oscillations when the extra nodes move to their balance positions (along the refined reference). The strong refinement is introduced just before failure is to initiate (second dashed vertical line) and the nodal positions continue to follow the refined-with-CZ reference.

3.2. Element with two delaminations

Our second example is similar to the first one, but with a different layup [-30,0,90,-60,30,60]. Again, the delaminations are expected to grow in interfaces with a 90° ply offset, resulting in delaminations in the second and fourth interface. We once again sample the displacements of a node in the bottom and in the top of each delaminated interface, see Fig. 20.

First, we present the results from non-adaptive simulations, i.e. with the 0/90 interfaces strongly refined from the start. The nodal displacements and reaction forces in Fig. 21 and 22 show that the model with the user element exactly reproduces the results from the reference model with CZ elements.

To run the simulation adaptively we made a similar, but shorter, study of the oscillations induced by the adaptive weak refinements as before. This study shows that the average oscillation (in the two interfaces) is $\Omega_0 = 0.00265$ mm, according to Eq. (39),

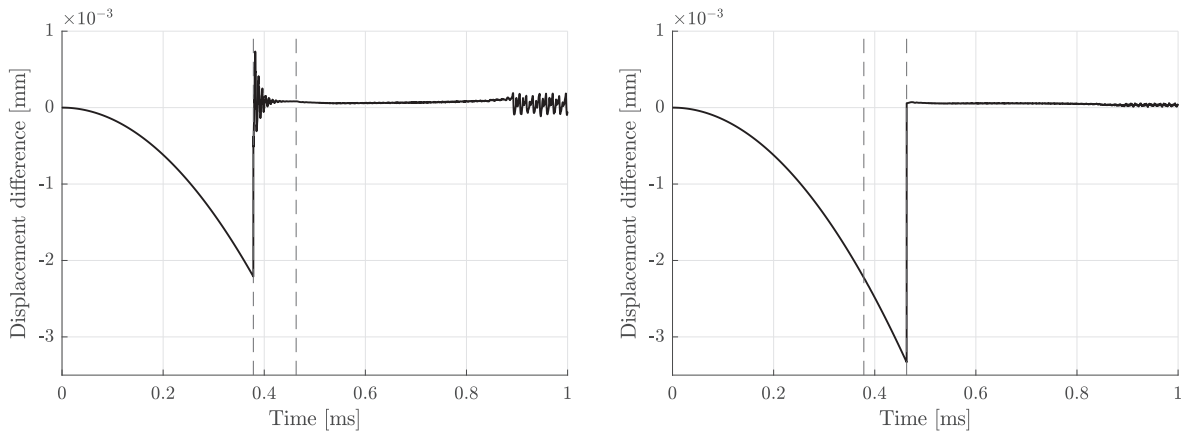


Fig. 15. Adaptive simulation: Difference in displacement between user element and refined-with-CZ reference of bottom (left) and top (right) node in 90/0 interface (cf. Fig. 14). The displacement differences are <2% of the reference displacements (in the time range. 0.5–0.85 ms).

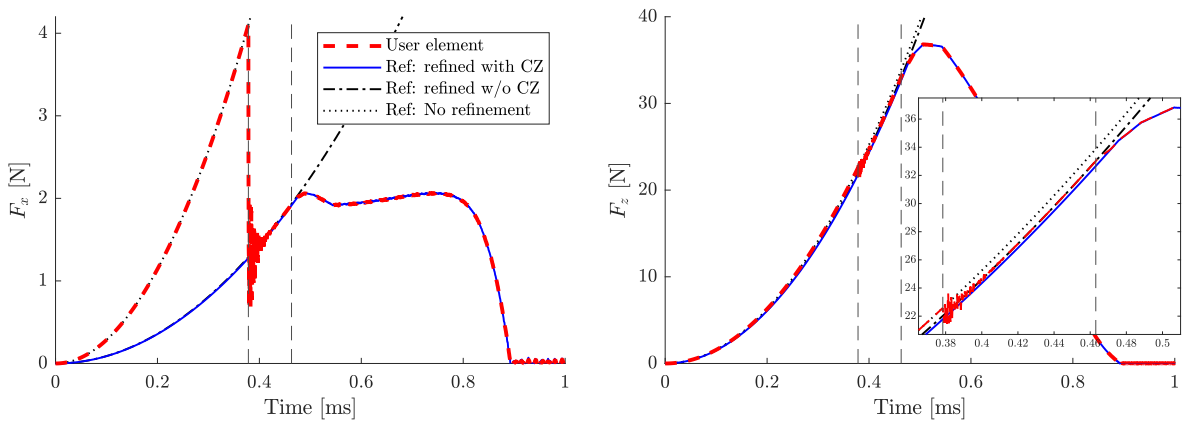


Fig. 16. Adaptive simulation: Total reaction force in x (left) and z (right) direction in the bottom nodes. Before the weak refinement is activated the force follows the unrefined reference. At activation (first dashed vertical line) the load drops to the refined reference curve (and damping is applied). The strong refinement is introduced just before failure is to initiate (second dashed vertical line) and the force continue to follow the refined-with-CZ reference.

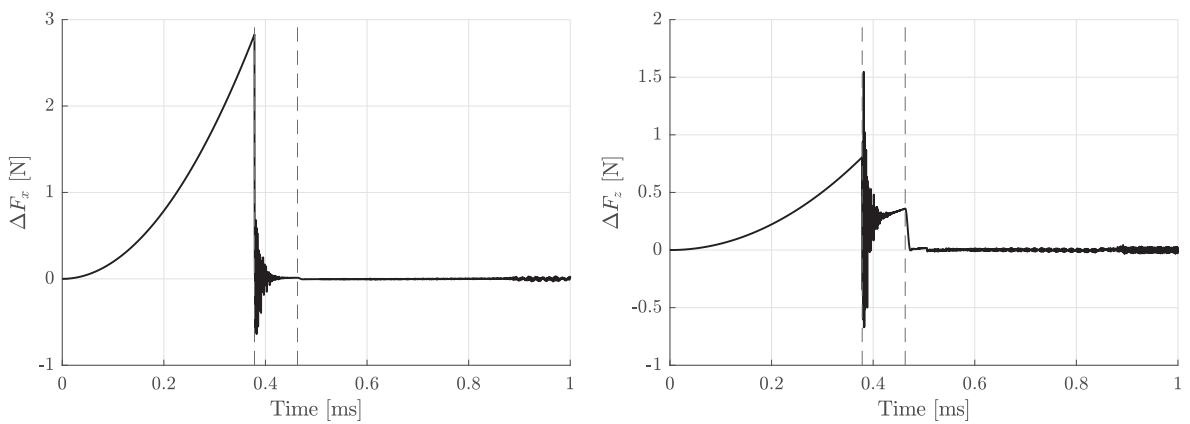


Fig. 17. Adaptive simulation: Force difference between user element and refined-with-CZ reference from Fig. 16. The force differences are <0.5% of the reference forces (in the time range. 0.5–0.85 ms).

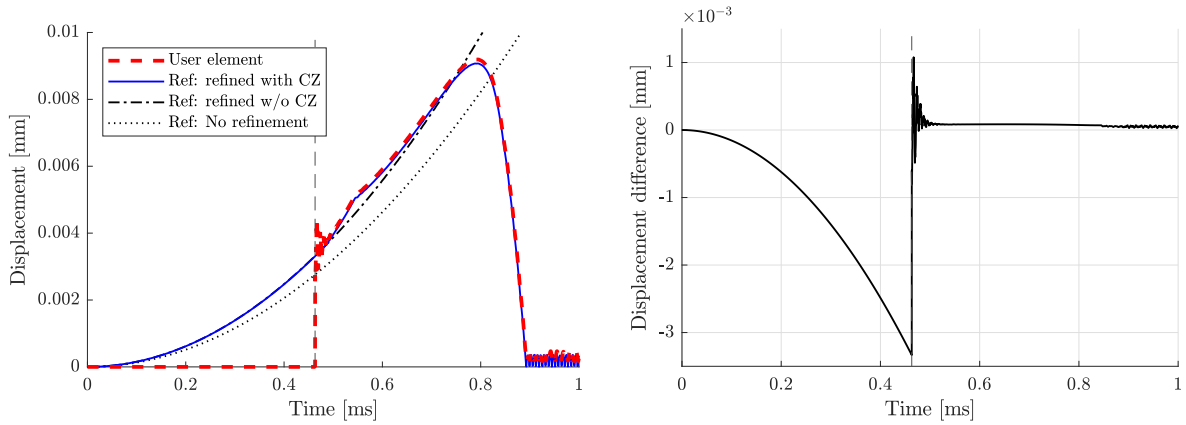


Fig. 18. Adaptive simulation with simultaneous weak and strong refinements: Resulting displacement of one node at the bottom of the middle interface (left) and the displacement difference between user element and refined-with-CZ reference (right). Before the weak refinement is activated the extra nodes have zero displacement. The weak and strong refinements are activated just before failure is to initiate (dashed vertical line) and the nodal positions continue to follow the refined-with-CZ reference. The displacement differences are <2.5% of the reference displacements (in the time range 0.5–0.85 ms).

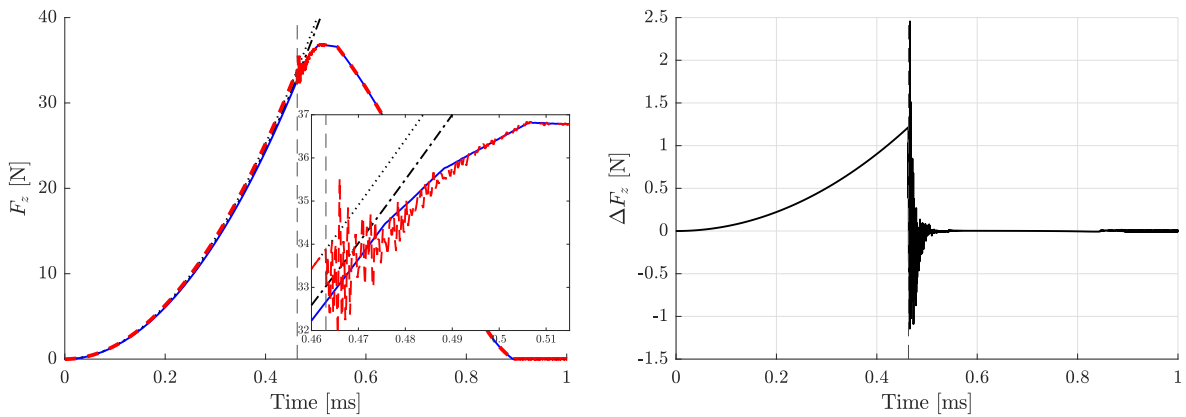


Fig. 19. Adaptive simulation with simultaneous refinement and CZ introduction: Total reaction force in z direction (left) and the force difference between user element and refined-with-CZ reference (right). Initially the force follows the unrefined reference. When the weak and strong refinements are activated just before failure is to initiate (dashed vertical line) the force continue to follow the refined-with-CZ reference. The force differences are <0.3% of the reference forces (in the time range 0.5–0.85 ms).

and the smallest dominating frequency is 724 rad/ms.⁵ We chose a damping ratio of 10% and in order to achieve a remaining oscillation level of 0.5×10^{-7} the damping was applied a duration of $\Delta t_{ref} = 0.15$ ms. Furthermore, the weak and strong refinements were activated simultaneously at 0.46 ms, corresponding to $\hat{f}_{CZ} = f_{CZ,lim} = 0.78$, evaluated from the unrefined element. As in the previous example with one delamination, the corresponding damping ratio of the gradual separation of the strong refinements was approximately 40%, so the interface was released during $\Delta t_{CZ} = 0.01$ ms here as well. The adaptive parameter values are summarised in Table 4.

The resulting nodal displacements and reaction forces are plotted in Fig. 23–26, which show (once again) that we can almost exactly recreate the same results as the reference case.

The average computational cycle time for the unrefined user element is here 0.228 ms, while the (strongly) refined user element has a cycle time of 0.266 ms, resulting in a computational save of 14% when the extra nodes are not active.

3.3. Triple cantilever beam

As a final numerical example we have simulated a triple cantilever beam (TCB), illustrated in Fig. 27, to show that several elements can be coupled together. The beam consists of eight plies with material properties given in Table 5, all having the fibres oriented along the beam. Since there was no pre-crack present we decided that the refinements should be made in the second, fourth

⁵ The lower frequency compared to the first example can be explained by the fact that there are no boundary conditions on the middle subelement. This is only suspended by the adjacent CZ elements.

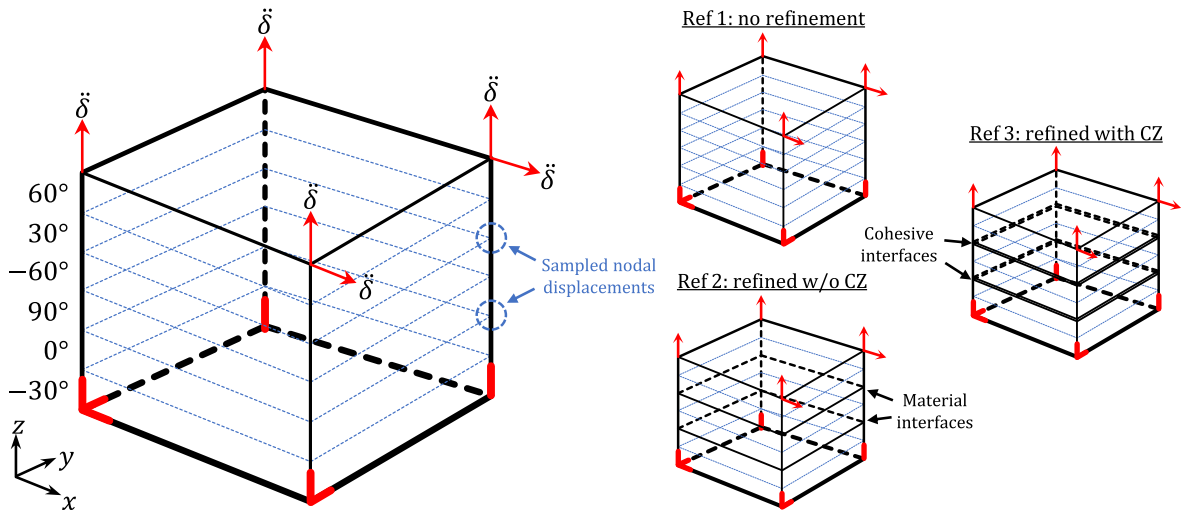


Fig. 20. Illustration of one-element numerical example with two delaminations. The model is constrained in the bottom (thick red marks) and loaded by prescribed velocity $\deltä$ at the top. CZ elements will be introduced in the 90/0 and $-60/30$ interfaces ($\Delta\theta_{del} > 45deg$) and the displacements of a node on the bottom and the top of the interface are extracted (left). The user element is compared to three different references modelled with standard LS-DYNA elements: One element representing the entire laminate (no refinement); Three sublaminate elements sharing nodes (refined w/o CZ) and three sublaminate elements connected by CZ elements (refined with CZ). The layout is with respect to the x -axis.

and sixth interfaces. The second and sixth are allowed to delaminate as strong refinements, while the fourth is considered as a material interface, *i.e.* a weak refinement. The beam was loaded down- and upwards in one end and was clamped in the other. In order to force two delaminations to grow a symmetry condition was applied to the mid material interface (once it was activated).⁶ An in-plane element size of 0.5 mm was chosen and in order to achieve stable delamination propagation we assigned the CZ elements a rather high fracture toughness and a low failure initiation strength, cf. Table 6. To verify the results we compare to a reference model made with standard LS-DYNA elements and materials (corresponding to those used internally in the user element).

In Fig. 28 we have plotted the reaction force versus displacement for the reference, a non-adaptive and an adaptive simulation. The user element with an initial refinement matches the reference exactly. The adaptive simulation shows a lower initial stiffness, but when refinements are activated, the curve approaches the others. After the delamination has started to propagate all models match very well.

The average computational cycle time for the unrefined user element is 1.84 ms, while the (strongly) refined user element has a cycle time of 2.18 ms. This results in a computational save of 16% when the extra nodes are not active.

In the adaptive simulation above, the refinements were made at a displacement level of $\delta = 0.0023$ mm. This corresponds to a failure initiation level of $f_{CZ} = 1.0$ in the outermost CZ element of the reference model. If we instead evaluate the cohesive failure initiation criterion in Eq. (17) using the intralaminar stresses of the unrefined element, we get a level of $\hat{f}_{CZ} = 0.48$. This indicates that it is difficult to predict the failure initiation of the CZ elements using the intralaminar stresses. A knock down factor of 50% might be necessary. No damping or gradual separation of the CZ element was needed to stabilise the solution and in Fig. 30 the deformed beams are shown.

If the refinement activation was postponed to a time long after cohesive failure initiation in the reference model (*e.g.* at $\delta = 0.005$ mm), there was a non-physical propagation of the delamination, as is shown in Fig. 29. However, by releasing the refinement over $\Delta t_{CZ} = 0.01$ ms the solution was stabilised, even if the force peak was a bit lower and postponed. For this example, the start position and velocity of the extra nodes during refinement (cf. Eqs. (15), (16), (19) and (20)) is likely rather good due to the trough-the-thickness homogeneity in the material properties. Instead, oscillations are the result of refining past the correct failure initiation of the elements, which should be avoided.

The main conclusion from this example is that for many practical applications (*i.e.* more than one element) it might unnecessary to stabilise the weak and strong refinements. If non-physical oscillations do occur there is the possibility to apply damping and/or gradual release as a stabilisation measure. However, a better option is to make sure that the refinements are made before failure should have initiated.

4. Concluding remarks

In this paper we have addressed the need to use high-fidelity models when performing simulations of fracture in laminated FRP. The reason is that the associated extreme computational costs are infeasible in industrial applications such as full vehicle crash

⁶ Otherwise one delamination will dominate due to numerical truncation errors and we will not have two delaminations that initiate and propagate.

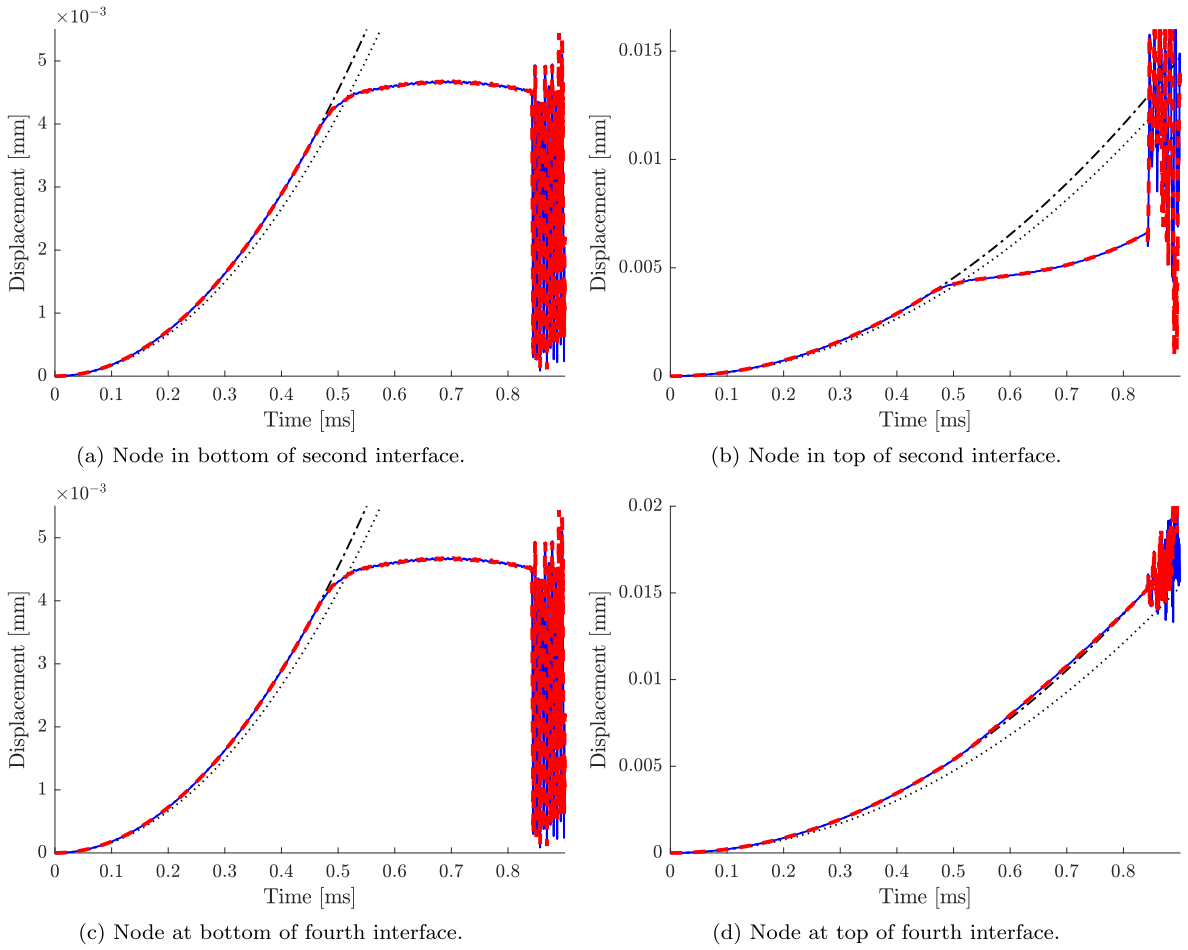


Fig. 21. *Non-adaptive simulation:* Resulting displacement of one node in the bottom (a and c) and in the top (b and d) of the second and fourth interfaces, respectively (cf. Fig. 20). Up to the point of damage initiation in the CZ element, the user element and the two refined reference behave the same. After initiation, a delamination is formed in the user element and the reference with CZ element, whose displacement curves match exactly up to the point where both interfaces fail. After this the middle subelement floats freely resulting in heavy oscillations, whereby any comparison between solutions after this time instant is irrelevant.

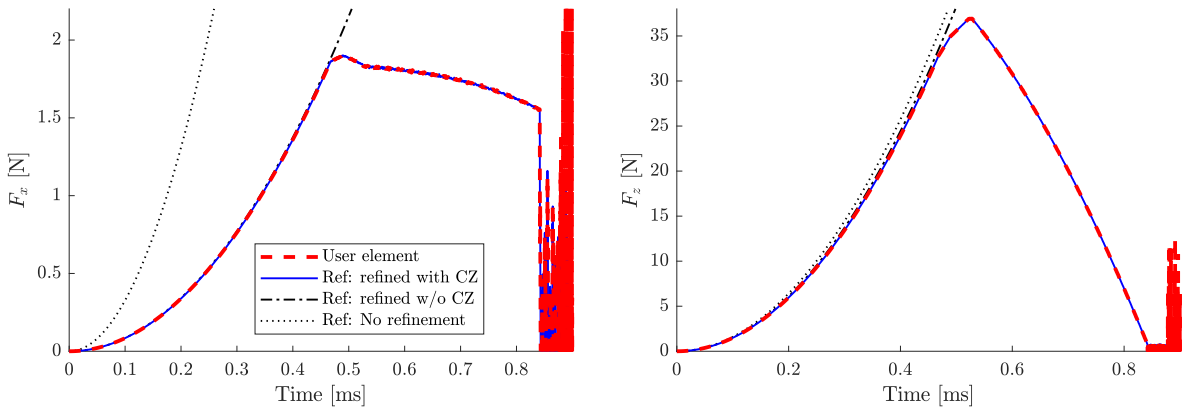


Fig. 22. *Non-adaptive simulation:* Total reaction force in x (left) and z (right) direction in the bottom nodes. The user element and the reference with CZ match exactly up to the point where both interfaces fail (after which the middle subelement floats freely).

Table 4
Values of adaptive parameters in second example.

Parameter	Value
$\Delta\theta_{del}$	50°
t_{ref}	0.46 ms ($\hat{f}_{CZ} = 0.78$)
$C_{d,ref}$	145 rad/ms ($\zeta = 0.10$)
Δt_{ref}	0.15 ms
$\hat{f}_{CZ,lim}$	0.78 ($t_{CZ} = 0.46$ ms)
Δt_{CZ}	0.01 ms

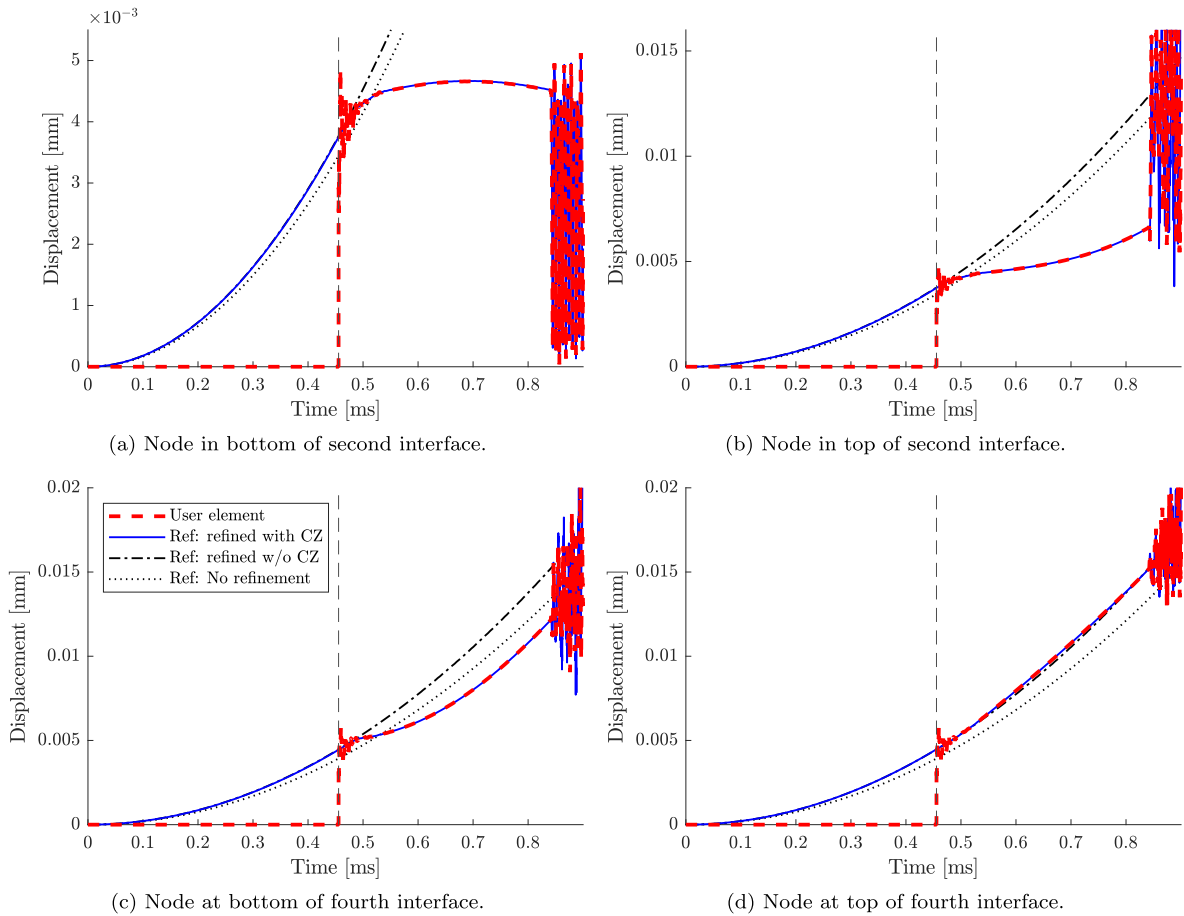


Fig. 23. Adaptive simulation: Resulting displacement of one node in the bottom (a and c) and in the top (b and d) of the second and fourth interfaces, respectively. Before the refinements are activated the extra nodes have zero displacement. The weak and strong refinements are activated just before failure is to initiate (dashed vertical line) after which the nodal positions continue to follow the refined-with-CZ reference up to the point where both interfaces fail (and the middle subelement starts to float freely).

simulations. As a solution, we present an adaptive refinement method that can when fully explored can reduce computational time by limiting the detailed modelling to areas where it is needed. Furthermore, we present stabilisation methods to deal with the non-physical oscillations that can occur when refining elements during a simulation in an explicit dynamic solver.

4.1. Summary of method

We have implemented our adaptive method for modelling multiple and arbitrarily located delamination cracks in an ESL shell element in the LS-DYNA explicit solver. The method starts with one ESL shell through the thickness. Using different refinement indicators the element can first be refined through the thickness to account for material interfaces (weak refinement). As a second refinement step, the interfaces can be separated and CZ elements be inserted in order to model delamination initiation and propagation (strong refinement). As the refinements are introduced in an explicit dynamic solver setting, stabilisation measures can be

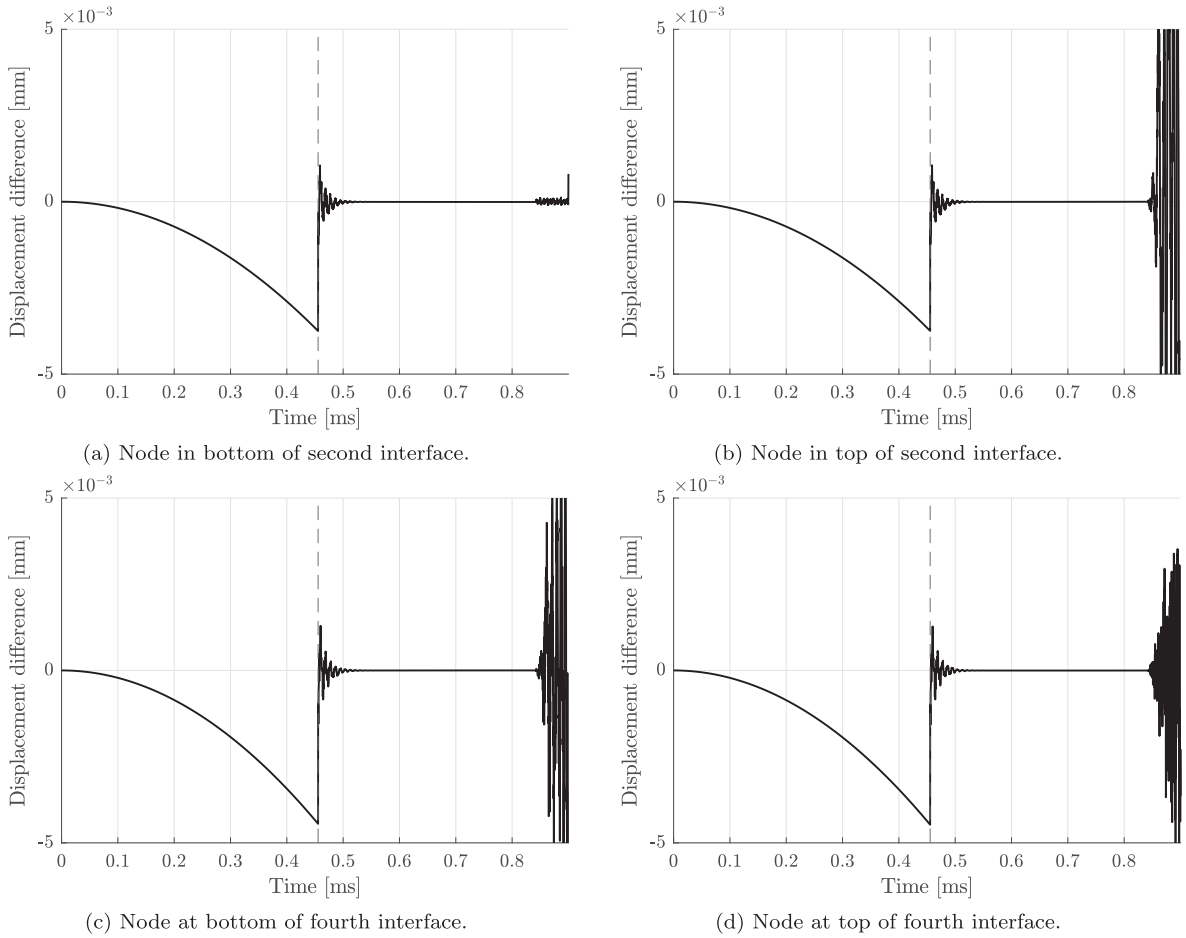


Fig. 24. Adaptive simulation: Difference in displacement between user element and refined-with-CZ reference of bottom (a and c) and top (b and d) node in 90/0 interfaces (cf. Fig. 21). The displacement differences are <0.2% of the reference displacements (in the time range 0.55–0.83 ms).

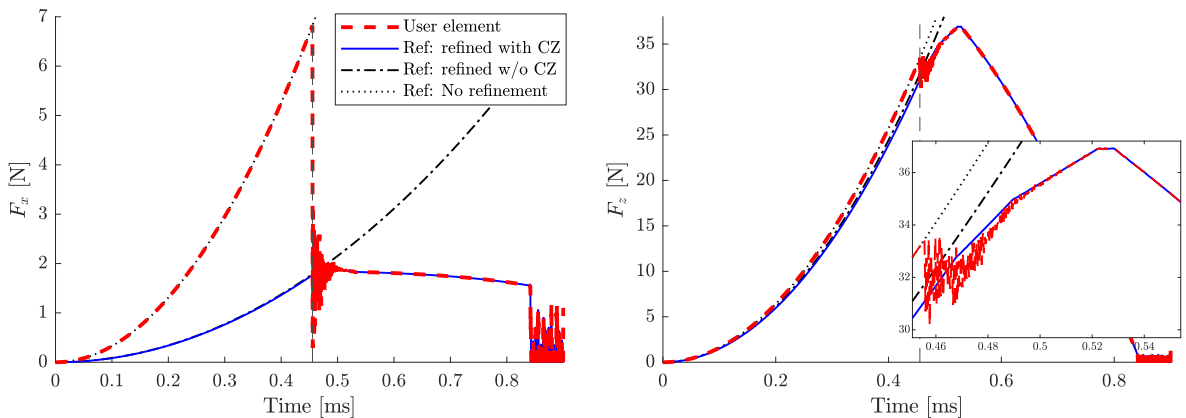


Fig. 25. Adaptive simulation: Total reaction force in x (left) and z (right) direction in the bottom nodes. Before the refinements are activated the force follows the unrefined reference. The weak and strong refinements activated just before failure is to initiate (dashed vertical line) and the force continue to follow the refined-with-CZ reference.

needed. The stabilisation involves applying a small amount of damping for a short time period when the weak refinements are activated. To stabilise the strong refinements, a force that ties the (duplicated) interfaces together is applied and gradually decreased in order to yield a smooth introduction. The adaptivity and stabilisation are controlled by a set of parameters summarised in Table 7, where we also have suggested some default values based on our experience.

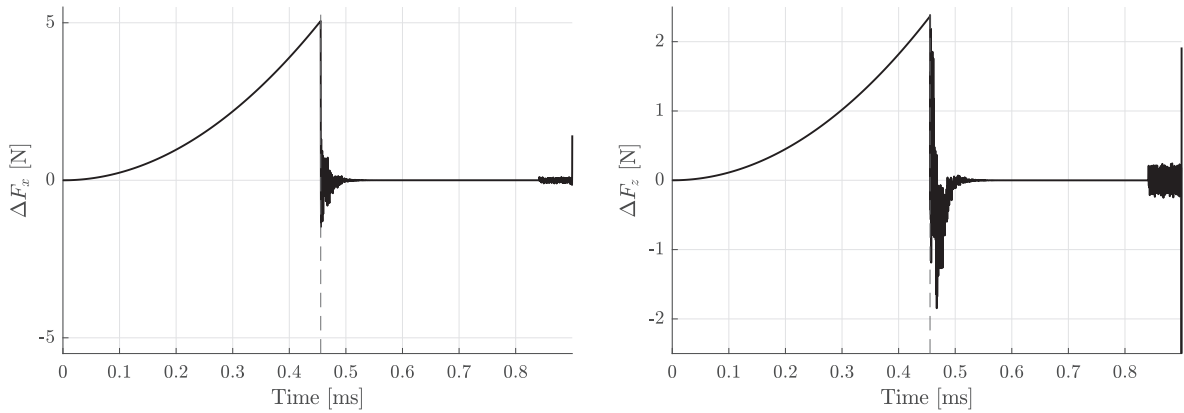


Fig. 26. Adaptive simulation: Force difference between user element and refined-with-CZ reference from Fig. 25. The force differences are <0.04% of the reference forces (in the time range 0.55–0.83 ms).

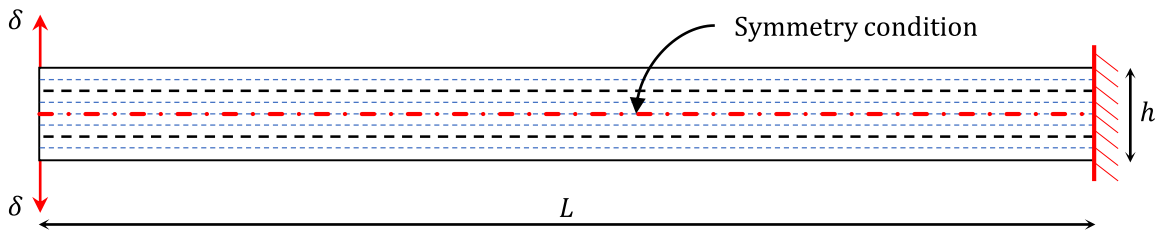


Fig. 27. Illustration of triple cantilever beam numerical example. The beam, which has a length of $L = 50$ mm long and thickness of $h = 2.048$ mm, is clamped in one end and loaded by prescribed displacement δ down- and upwards at the other end. Strong refinements with CZ elements will be added to the second and sixth interface (marked with black dashed line) and the fourth interface will be weakly refined and have a symmetry condition in the thickness direction (marked with red dot-dashed line). The beam is modelled with one element across the width. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Table 5
Ply material properties used in the beam example.

E_1 (GPa)	E_2, E_3 (GPa)	G_{12}, G_{13} (GPa)	G_{23} (GPa)	ν_{21}, ν_{31} (-)	ν_{32} (-)	ρ (kg/m ³)
100	10	5	4	0.025	0.25	1600

Table 6
Interface material properties used in the beam example.

σ_I (MPa)	\mathcal{G}_{Ic} (J/m ²)	k_I (GPa/mm)	σ_{II} (MPa)	\mathcal{G}_{IIc} (J/m ²)	k_{II} (GPa/mm)	ρ_{CZ} (kg/m ²)
20	400	1000	20	500	1000	0.4096

Generally, the best method for finding suitable values of the stabilisation parameters is to start with no stabilisation at all. If the (weak or strong) refinement causes immediate damage in large parts of the model, the refinement is likely made too late with respect to when failure should have initiated. As a first step, the refinement should be advanced to an earlier time. If this does not remedy the problem, one should try to assess whether the unwanted oscillations are related to the weak or the strong refinement.

If it is related to weak refinements, an analysis of the oscillations should be performed. Applying a damping value $C_{d,ref}$ equal to 10% of the dominating (lowest) frequency for a short period should reduce the oscillations. The damping period Δt_{ref} can be calculated using Eq. (23). If instead the oscillations are related to the strong refinement, this refinement should be introduced gradually over a short period Δt_{CZ} . Similar to damping, a gradual separation of the strong refinement will produce less oscillations compared to a sudden separation, however, only to a certain degree. With regards to calculating the separation period Δt_{CZ} , we once again refer to Eq. (23) where a damping ratio of 30–40% can be assumed. However note that long separation periods should be avoided since it can interfere with the failure initiation of the CZ elements.

Finally, as the ultimate goal of adaptive refinements is to be able to refine small parts of a larger structure, refining a bit too early will not make a large difference when it comes to computational costs.

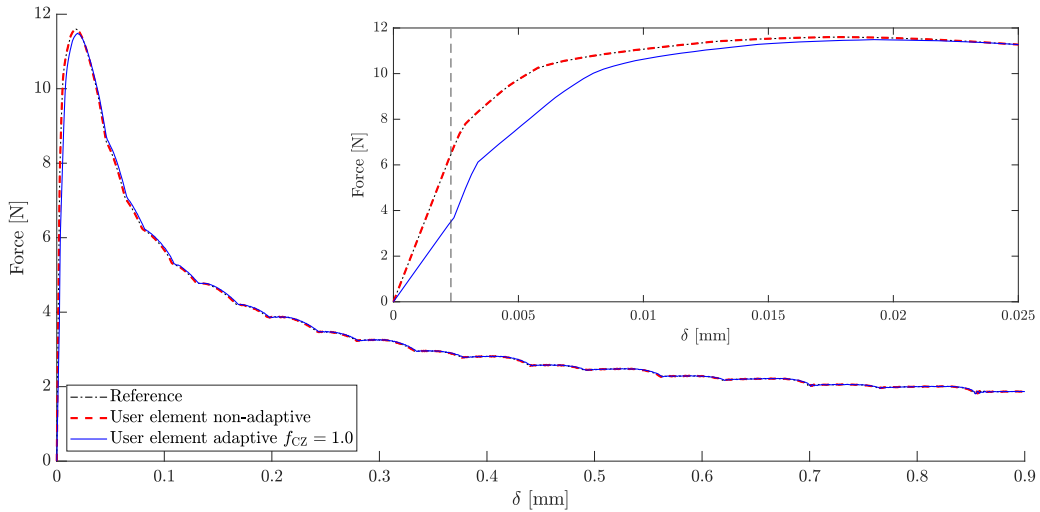


Fig. 28. Load versus displacement curve for TCB example. The user element with an initial refinement matches the reference exactly. The adaptive simulation shows a lower initial stiffness, but when refinements are activated at $\delta = 0.0023$ mm (vertical line in zoom), the curve begins to join the others. After the delamination has started to propagate all models match very well. The peak force is <1% lower and the difference in total work is <0.2% lower in the adaptive simulation compared to the reference.

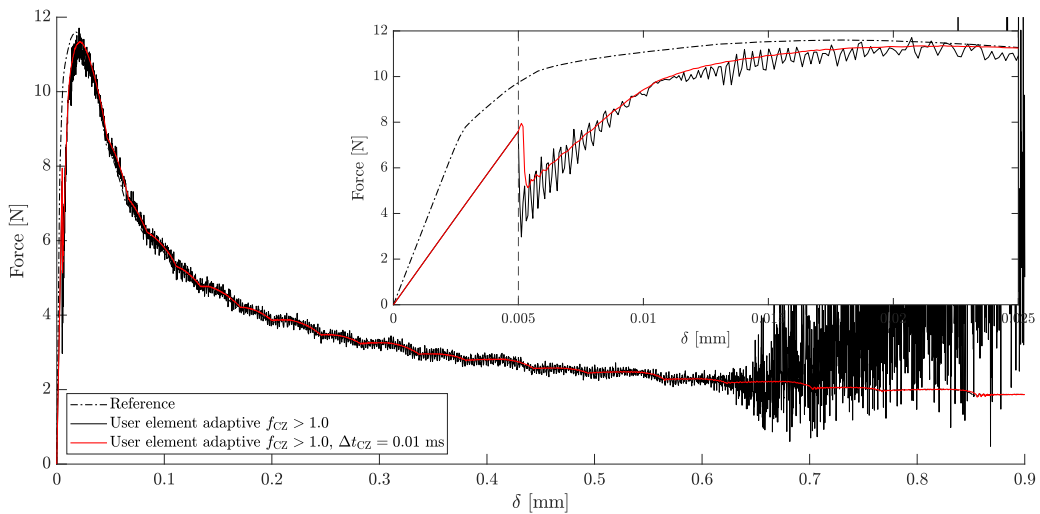


Fig. 29. Load versus displacement curve for TCB example. If the refinements are activated to late (vertical line in zoom), there will be an abrupt jump in the reaction force followed by small oscillations and a non-physical propagation of the delamination. By releasing the refinement over $\Delta t_{CZ} = 0.01$ ms the solution is stabilised. The peak force is 2% lower and the difference in total work is <0.2% lower in the stabilised adaptive simulation compared to the reference.

4.2. Perspectives on related work

The adaptive method we have implemented is intended to enable computationally efficient crash simulations of laminated FRP. As we mentioned already in the introduction, the idea to improve efficiency using adaptive refinements or remeshing techniques is not unique.

Different enrichments techniques, including the eXtendend Finite Element Method (XFEM) [29,30], Phantom Node Method (PNM) [31] or similar have been used extensively to model propagating cracks by dynamic insertion of cohesive segments. For example element enriched with intralaminar cracks were modelled by van der Meer and Sluys [9], Iarve et al. [32], Fang et al. [33], Kawashita et al. [34], Ahmed and Sluys [35], Brouzoulis and Fagerström [10], Reiner et al. [12], Lu et al. [15] and Zhi and Tay [16]. Elements enriched with interlaminar cracks can be found in e.g. Remmers et al. [8], Shor and Vaziri [13], McElroy [14] and Brouzoulis et al. [11] and Adams et al. [17].⁷

⁷ An isogeometric shell element where delaminations are modelled by modifying the knot-vector associated to the thickness interpolation.

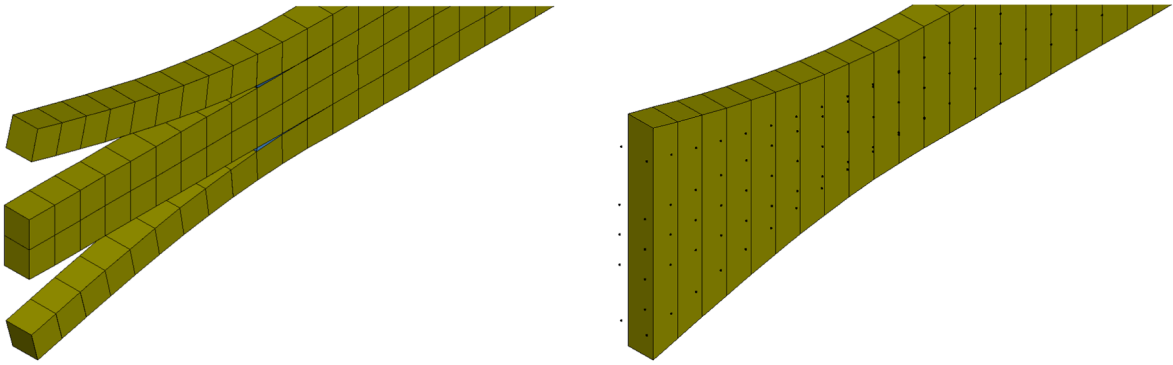


Fig. 30. Deformed reference (left) and user element (right) beam at $\delta = 0.9$ mm displacement. The nodes which define the internal subelements in the user elements are visible as black dots.

Table 7
Parameters used to control adaptivity and stabilisation thereof.

Parameter	Meaning	Default
t_{ref}	Weak refinement time	(before intralaminar damage)
$C_{d,ref}$	Weak refinement damping value.	$\leq 0.10 \times C_{crit}$
Δt_{ref}	Duration of weak refinement damping.	0.1–0.2 ms
$f_{CZ,lim}$	CZ failure index level when strong refinements are made	$\leq 0.5^a$
Δt_{CZ}	Duration of strong refinements introduction.	≤ 0.01 ms

^a Can be increased if stress recovery of out-of-plane stresses is made.

Generally, the sources mentioned above can capture the kinematics of cracked elements, however, the degree of adaptivity employed is sometimes hard to assess. Furthermore, an implicit solver is used in most cases, making the method unsuitable for crash simulations. Worth to highlight is the method developed by Shor and Vaziri [13], which indeed is intended for crash simulations. In their approach, one centrally placed delamination can be represented in each through-the-thickness solid shell element by remodelling and restarting the FE simulation. They present very promising results and their method has many similarities to ours. The obvious differences are that our method is implemented directly in LS-DYNA⁸ and we can add several delaminations in one element without the need to do restarts. The choice to perform restarts has its benefits, e.g. there is no need to adjust the mass as we do in Section 2.5. However, also Shor and Vaziri report instabilities, although related to the restart, not the refinement itself.

In the current work we have not addressed the issue that the critical time step can be decreased by both the through-the-thickness refinement and the introduction of CZ elements. We have not implemented any method for manipulating the model, e.g. by adding nodal mass, such that the critical time step can be adjusted. Instead, we have manually monitored that the initial computational time step chosen by LS-DYNA is not larger than the critical time step of the refined model. This leads us to the next subsection where we will address some of the remaining issues with our current implementation.

4.3. Future work

Here we will list some areas of our method that needs to be improved in order to be of practical use in industrial crash simulations:

- The time step calculation does not exploit the fact that the stiffness in the thickness direction for most laminated FRP is around an order of magnitude lower than the stiffness in the in-plane directions. This means that thickness of the element can be considerably smaller than the in-plane length without having the computational time step violate the (true) critical time step. This is not considered by the current critical time step estimation in LS-DYNA. However, refining an element through the thickness can lead to violation of the critical time step. Thus, the computational time step must also be adjusted accordingly, either for the entire model or by utilising a sub-cycling technique. Alternatively, the mass of the refined elements must be increased.
- The damping method we are currently using is not very versatile and a more refined method that can damp individual nodes should be used. This requires additional modifications of the LS-DYNA user interface. Alternatively, the damping can be replaced, e.g. by employing some time-bridging technique like the Arlequin method [36,37] where the coarse mesh is superimposed by a refined mesh. Then the model can go smoothly from unrefined to refined over a period of time.
- The numerical examples show that the unrefined user element is, at least, 11–16% faster compared to when it is refined. However, in the current implementation a refinement is not limited to an in-plane patch which then can expand (as was the case in our

⁸ Their method is implemented as an external Python script which uses LS-DYNA as solver.

implementation in the open-source software OOFEM [7]). The main reason for this is that the refinements are represented using the AFEM. While this technique is rather straightforward from an implementation point of view, continuity at the refinement patch boundary requires that the movement of the boundary nodes can be constrained. This is difficult to achieve without having access to the LS-DYNA source code. Nevertheless, by implementing a patch refinement larger parts of the structure can remain in an unrefined state during the simulation and the computational efficiency can be increased. In addition, to fully exploit the computational efficiency of an adaptive refinement approach all non-active extra DOF should be excluded by the solver, which is not the case at the moment.⁹ To achieve this exclusion requires modifications to the software, which are outside the scope of the present work. We would like to point out that the computational efficiency of an adaptive refinement method is depending on the choice of element formulation and material model, especially when an explicit solver is used.¹⁰ An initially complex model (e.g. high number of IP and advanced material model) will be computationally expensive, even if all IP through the thickness are included in one element. Refining the element will then not significantly increase the total computational expenses. To achieve a computationally efficient adaptive method when using an explicit solver, the initial model should be inexpensive and having complex features added adaptively during the simulation.

- We showed already in [7] that recovery of the out-of-plane stresses can significantly improve the prediction of the stress state in the ESL shell element (and thus when to refine the element). However, the method we used there is not suitable for our current implementation (since it requires non-local information). An alternative approach would be to resort to the so-called Extended 2D method by Rolfes and colleagues [38,39].

In the list above we have not mentioned anything related to parallelisation of the computational domain, which must be done in order to perform crash simulations. This is of course of interest but more related to the particular choice of solver and therefore not within the current research scope. Furthermore, we have only used elastic intralaminar material models but will focus more on intralaminar fracture in Part II [20].

4.4. Implications

We have presented a framework for facilitating adaptive through-the-thickness refinements and insertion of CZ elements to model delaminations in ESL shell elements. The framework is general and many parts of it can be exchanged, e.g. we are not restricted to a certain element formulation. If other element or material formulations, better suited to modelling fracture in FRP, are developed these can easily be used in the framework.

Nevertheless, we showed that we can reproduce similar results as reference high-fidelity models while saving computational effort. This is a good starting point for enabling computationally efficient crash simulations of laminated FRP. In the long run this will help to develop automotive structures made of FRP.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

The current work has been funded by Vinnova through the Swedish Strategic Vehicle Research and Innovation Programme on Vehicle and Traffic Safety via the project *Modelling crash behaviour in future lightweight composite vehicles – Step 2* (Grant No. 2016-04239) which is gratefully acknowledged. Martin Fagerström also acknowledges additional financial support from Vinnova through the project LIGHTer Academy (Grant No. 2017-05200). The simulations were performed on resources provided by the Swedish National Infrastructure for Computing (SNIC) at Chalmers Centre for Computational Science and Engineering (C3SE).

References

- [1] Carruthers JJ, Kettle AP, Robinson AM. Energy absorption capability and crashworthiness of composite material structures: A review. *Appl Mech Rev* 1998;51(10):635–49.
- [2] Jacob GC, Fellers JF. Energy absorption in polymer composites for automotive crashworthiness. *J Compos Mater* 2002;36(07):813–50.
- [3] Park Chung-Kyu, Kan Cing-Dao (Steve), Hollowell William Thomas. Investigation of opportunities for light-weighting a body-on-frame vehicle using advanced plastics and composites. In: 23rd Enhanced Safety of Vehicles Conference, Seoul, Korea; 2013.
- [4] ERTRAC. European Roadmap Safe Road Transport. Technical report. European Road Transport Research Advisory Council; 2011.
- [5] Hull D. A unified approach to progressive crushing of fibre-reinforced composite tubes. *Compos Sci Technol* 1991;40(4):377–421.
- [6] Grauers Lisa, Olsson Robin, Gutkin Renaud. Energy absorption and damage mechanisms in progressive crushing of corrugated NCF laminates: Fractographic analysis. *Compos Struct* 2014;110:110–7.
- [7] Främby Johannes, Fagerström Martin, Brouzoulis Jim. Adaptive modelling of delamination initiation and propagation using an equivalent single-layer shell

⁹ Currently, LS-DYNA does not allow for the extra nodes to be dynamically allocated during the simulation.

¹⁰ The cost of adding DOF only increases linearly in an explicit solver while the cost increases exponentially in an implicit solver. An adaptive method can therefore be considered more computationally efficient in an implicit solver.

- approach. *Int J Numer Meth Eng* 2017;112(8):882–908.
- [8] Remmers Joris JC, Wells Garth N, de Borst René. A solid-like shell element allowing for arbitrary delaminations. *Int J Numer Meth Eng* 2003;58(13):2013–40.
- [9] van der Meer FP, Sluys LJ. A phantom node formulation with mixed mode cohesive law for splitting in laminates. *Int J Fract* 2009;158(2):107–24.
- [10] Brouzoulis Jim, Fagerström Martin. An enriched shell element formulation for efficient modeling of multiple delamination propagation in laminates. *Compos Struct* 2015;126:196–206.
- [11] Brouzoulis Jim, Fagerström Martin, Svenning Erik. An enriched shell element formulation for modeling of inter- and intralaminar crack propagation in laminates. *Compos Struct* 2015;136:616–25.
- [12] Reiner Johannes, Veidt Martin, Dargusch Matthew, Gross Lutz. A progressive analysis of matrix cracking-induced delamination in composite laminates using an advanced phantom node method. *J Compos Mater* 2016;51(20):2933–47.
- [13] Shor Ofir, Vaziri Reza. Application of the local cohesive zone method to numerical simulation of composite structures under impact loading. *Int J Impact Eng* 2017;104:127–49.
- [14] McElroy Mark. Use of an enriched shell finite element to simulate delamination-migration in a composite laminate. *Compos Struct* 2017;167:88–95.
- [15] Lu X, Chen BY, Tan VBC, Tay TE. Adaptive floating node method for modelling cohesive fracture of composite materials. *Eng Fract Mech* 2018;194:240–61.
- [16] Zhi Jie, Tay Tong-Earn. Explicit modeling of matrix cracking and delamination in laminated composites with discontinuous solid-shell elements. *Comput Methods Appl Mech Eng* 2019;351:60–84.
- [17] Adams Camiel, Fagerström Martin, Remmers Joris JC. Efficient modelling of delamination growth using adaptive isogeometric continuum shell elements. *Comput Mech* 2020;65(1):99–117.
- [18] Borek Patzák. OOFEM; 2000.
- [19] LSTC. LS-DYNA; 2011.
- [20] Främby Johannes, Fagerström Martin. An adaptive shell element for explicit dynamic analysis of failure in laminated composites – Part 2: Progressive failure and model verification and validation; 2020 [Submitted for publication].
- [21] Wohlecker Roland. Modelling And Testing for Improved Safety of key composite Structures in alternatively powered vehicles - Summary of implemented composite modelling methods and outcome of their validation. Technical report; 2015.
- [22] Ling Daosheng, Yang Qingda, Cox Brian. An augmented finite element method for modeling arbitrary discontinuities in composite materials. *Int J Fract* 2009;156(1):53–73.
- [23] LSTC. LS-DYNA Theory Manual. 02/20/19 edition; 2019.
- [24] LSTC. LS-DYNA KEYWORD USER'S MANUAL - VOLUME I. 01/23/19 edition; 2019.
- [25] Belytschko Ted, Wong Bak Leong, Plaskacz Edward J. Fission-fusion adaptivity in finite elements for nonlinear dynamics of shells. *Comput Struct* 1989;33(5):1307–23.
- [26] Li Long-yuan, Bettess Peter. Adaptive finite element methods: a review. *Appl Mech Rev* 2009;50(10):581.
- [27] Menouillard Thomas, Belytschko Ted. Smoothed nodal forces for improved dynamic crack propagation modeling in XFEM. *Int J Numer Meth Eng* 2010;84(1):47–72.
- [28] LSTC. LS-DYNA KEYWORD USER'S MANUAL - VOLUME II. 01/23/19 edition; 2019.
- [29] Wells GN, Sluys LJ. A new method for modelling cohesive cracks using finite elements. *Int J Numer Meth Eng* 2001;50(12):2667–82.
- [30] Moës Nicolas, Belytschko Ted. Extended finite element method for cohesive crack growth. *Eng Fract Mech* 2002;69(7):813–33.
- [31] Hansbo Anita, Hansbo Peter. A finite element method for the simulation of strong and weak discontinuities in solid mechanics. *Comput Methods Appl Mech Eng* 2004;193(33):3523–40.
- [32] Iarve EV, Gurvich MR, Mollenhauer DH, Rose CA, Dávila CG. Mesh independent matrix cracking and delamination modeling in laminated composites. *Int J Numer Meth Eng* 2011;88(8):749–73.
- [33] Fang XJ, Yang QD, Cox BN, Zhou ZQ. An augmented cohesive zone element for arbitrary crack coalescence and bifurcation in heterogeneous materials. *Int J Numer Meth Eng* 2011;88(9):841–61.
- [34] Kawashita Luiz F, Bedos Alexandre, Hallett Stephen R. Modelling mesh independent transverse cracks in laminated composites with a simplified cohesive segment method. *Comput Mater Continua* 2012;32(2):133–58.
- [35] Ahmed A, Sluys LJ. A computational model for the simulation of dynamic fracture in laminated composite plates. *J Compos Mater* 2015;49(14):1717–38.
- [36] Dhia Hashmi Ben, Rateau Guillaume. The Arlequin method as a flexible engineering design tool. *Int J Numer Methods Eng* 2005;62(11):1442–62.
- [37] Xiao SP, Belytschko T. A bridging domain method for coupling continua with molecular dynamics. *Comput Methods Appl Mech Eng* 2004;193(17–20):1645–69.
- [38] Rolfes R, Rohwer K. Improved transverse shear stresses in composite finite elements based on first order shear deformation theory. *Int J Numer Meth Eng* 1997;40:51–60.
- [39] Rohwer K, Rolfes R. Calculating 3D stresses in layered composite plates and shells. *Mech Compos Mater* 1998;34(4).