Crack Propagation Simulation on Composite Wing Subjected to Gust Load by Means of XFEM

Nanda Wirawan

A thesis submitted in partial fulfilment of the requirements for the degree of

Doctor of Philosophy

The University of Sheffield

Faculty of Engineering

School of Mechanical, Aerospace and Civil Engineering

Submission Date

October 2024

This thesis is dedicated to my dearest wife
Intan Luruh Larasati
for her dedication to support my PhD journey

Acknowledgements

The author gratefully acknowledge Dr Jose Luis Curiel-Sosa and Dr Xinshan Li for their continuous support and the fruitful discussions. The author also acknowledges the funding from Indonesia Endowment Fund for Education (LPDP) and support from Indonesia National Research and Institute Agency (BRIN).

The author also would like to thank Dr Mahesa Akbar for his aeroelasticity expertise and the input on the following matters. Lastly, I want to dedicate my deepest gratitude to my family and friends.

Abstract

The utilisation of composite in aircraft industry is increasingly higher in the recent decades. The flexibility to optimise weight and strength by using composite is a great benefit for lightweight structure applications. However, as highlighted by many companies, i.e., Boeing and Airbus, further works need to be done on understanding the composite behaviours especially its damage evolution prior to structural failure.

In the current research, the development of a more accurate tool is focused to predict the structural integrity of a composite wing under operational loads. In the first part of the present work, a custom subroutine containing three-dimensional (3D) Puck failure criterion combined with exponential damage degradation model is developed. This subroutine aims to extend the capability of the finite element analysis (FEA) software to accurately model composite structures. Two benchmark studies are conducted, the single-element test and the open hole test. The results from both cases indicate good agreements with the experimental outcomes.

The second part of the present work focuses on improving the search algorithm for the Inter-Fibre Failure (IFF) on 3D Puck failure criterion. Hence, a novel fast fracture plane orientation angle (FPOA) search algorithm is proposed as a replacement for the standard linear search algorithm. The results demonstrate the proposed fast FPOA search algorithm is significantly more accurate than that of the other fast FPOA search algorithms, although this comes with a slight penalty in computational speed.

Finally, the third part of the present work focuses on the integration of the 3D Puck criterion with fast FPOA search algorithm into the Extended Finite Element Method (XFEM) framework. A benchmark test using XFEM shows that the crack propagates in similar orientation with an experimental result. Furthermore, computational analysis is conducted to evaluate the structural integrity of the composite lifting structure (wing) subjected to cruise and gust loads.

List of Publications

During his PhD studies, the author has published several works, including a journal article, conference proceedings, and a book chapter. These publications are listed below:

Journal Article

- Nanda Wirawan, Ibrahim H Abuzayed, Mahesa Akbar, and Jose L Curiel-Sosa. A robust fast fracture plane orientation angle search algorithm for Puck 3D inter-fibre failure criterion. European Journal of Computational Mechanics, 33 (3). pp. 227-254, 2024.

Conference Proceeding

- Nanda Wirawan and Jose L Curiel-Sosa. Progressive damage modelling on fibre-reinforced composite using 3D puck failure criterion with a fast fracture plane angle search algorithm. UKACM 2023 Conference Proceedings, 2023.
- Ibrahim H Abuzayed, Nanda Wirawan, Chao Zang and Jose L Curiel-Sosa. A 3D damage model for simulating damage modes in fibre metal laminates. 9th ECCOMAS Thematic Conference on the Mechanical Response of Composites, 2023.
- Faiz Hamzah, Nanda Wirawan, Fernando Cepero and Jose L Curiel-Sosa. Comparison of Computational Techniques for Prediction of Delamination Initiation and Propagation in Composite Laminates. 8th ECCOMAS Thematic Conference on the Mechanical Response of Composites, 2021.

Book Chapter

- Ibrahim H Abuzayed, Nanda Wirawan and Jose L Curiel-Sosa. A Numerical Study on the Flexural Behavior and Failure Mechanisms of Fiber Metal Laminates. Failure in Aircraft Materials, pp. 13-28. Cham: Springer Nature Switzerland, 2024.

Contents

List of Abbreviations

2D	Two-dimensional
3D	Three-dimensional
CDM	Continuum Damage Model
CFD	Computational Fluid Dynamics
CFRP	Carbon-Fibre Reinforced Polymer
DLM	Doublet Lattice Method
DoF	Degree of Freedom
EGSS	Extended Golden Search Section
FEA	Finite Element Analysis
FF	Fibre Failure
FPOA	Fracture Plane Orientation Angle
FSI	Fluid-Structure Interaction
GFRP	Glass-Fibre Reinforced Polymer
GSS	Golden Section Search
IFF	Inter-Fibre Failure
ISRBM	Improved Selective Range Brent's Method
NASA	National Aeronautics and Space Administration
NTSB	National Transportation Safety Board
OHT	Open-Hole Test
PCE	Polynomial Chaos Expansion
RANS	Reynold-Averaged Navier Stokes
SDV	State Dependent Variable
SPIS	Simple Parabolic Interpolation Search
SR	Selective Range
SRGSS	Selective Range Golden Search Section
SSM	Section Search Method
UDMGINI	User DaMaGe INItiation

UMAT	User MATerial
XFEM	eXtended Finite Element Method
WWFE	World Wide Failure Exercise

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Background and Motivation

In recent decades, composite structures implementation in high-performance applications has been growing significantly. Composite structures enable optimisation in mechanical performance, i.e., stiffness and strength, and weight reductions. Recently, composite materials, i.e., carbon fibre-reinforced polymers (CFRPs), have been utilised in the aerospace industry to construct the main components of aircraft structures [1]. For example, more than 50% of the airframe structures of Boeing 787 [2] (Figure 1.1) and Airbus A350 XWB [3] are made of CFRPs. Despite their growing application, composite structures in aircraft are still concerned with their distinct types of failure, i.e., delamination and crack.

Therefore, the composite structures must undergo strict prototype evaluation and testing before being used on the aircraft. However, with the advancement of computational resources, the high-cost and time-consuming prototype evaluation and testing iteration could be substituted with the numerical simulation. The numerical simulation could be used to predict the behaviour of the composite structures under a specific type of load up to a certain degree of accuracy. Regardless, the accuracy of the numerical simulation depends on the assumption and the fidelity of the finite element model used for the numerical simulation.

In the present work, a numerical simulation on a composite wing is performed to observe the response and the wing structural integrity after imposed by a sudden gust load. Due to the initial condition, geometric characteristics, and mechanical properties of the wing, an aeroelastic instability could arise after the load has subsided. As shown in Figure 1.2, the aeroelastic instability phenomenon is caused by the interaction between different forces, such as inertial forces, aerodynamic forces, and elastic forces during flight. This interaction if not properly mitigated can lead to performance and control degradation of the aircraft and damage or failure to the aircraft's structures, which can be catastrophic.

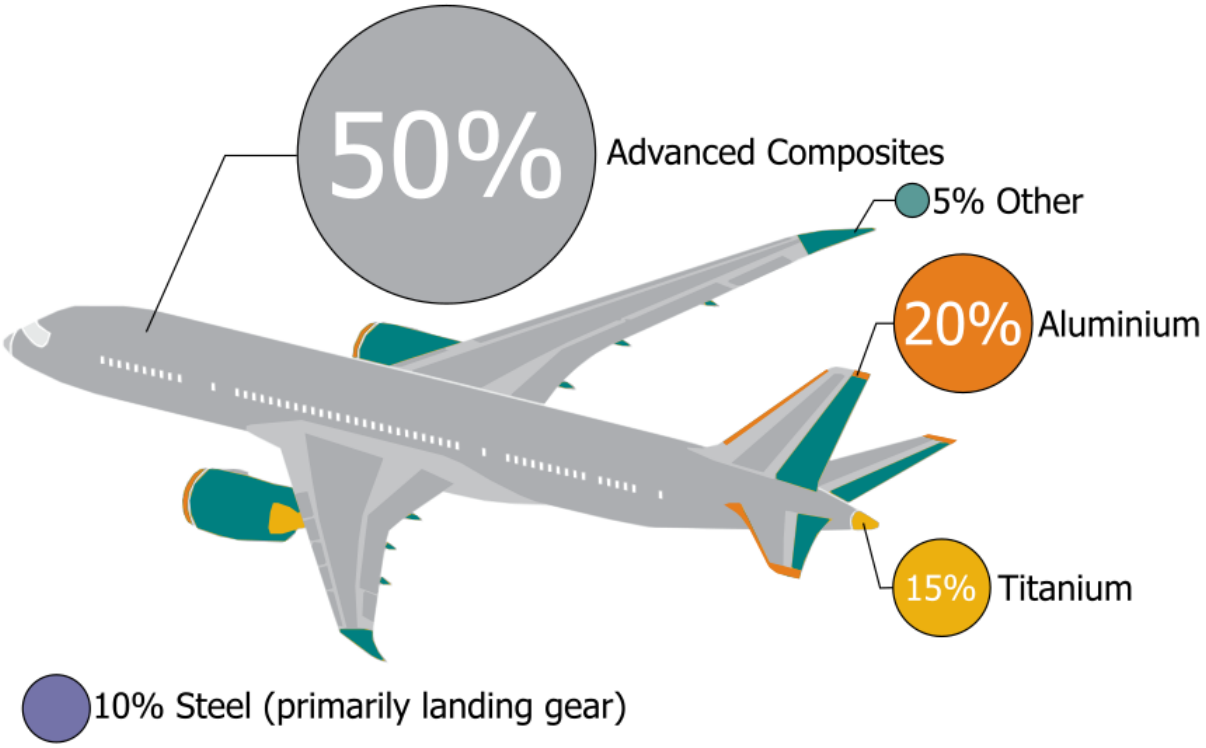


Figure 1.1: Material composition of Boeing 787 [2]

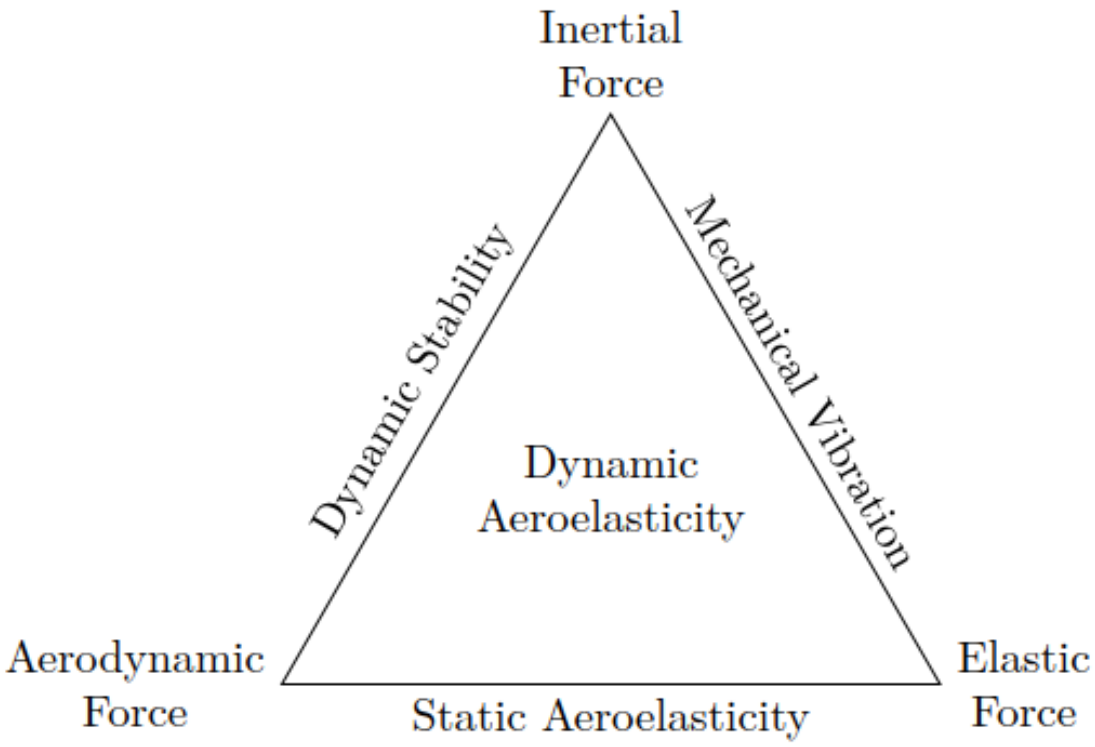


Figure 1.2: Interaction between inertial, aerodynamic and elastic force [4]

Initially, Wirawan et al. [5] carried out a numerical simulations to investigate the feasibility of using the Extended Finite Element (XFEM) to simulate crack propagation on an aluminium wing. The XFEM is an advanced numerical method developed to model [6, 7] crack initiation and propagation without re-meshing on a finite element model, significantly reducing the computational cost compared to the conventional finite element method. Later, Abdullah et al. [8] and Hoseini et al. [9] performed a similar crack propagation simulations using an XFEM, but focusing on a composite wing. However, in the works done by Abdullah et al. and Hoseini et al., the XFEM simulation was carried out using the maximum principal stress failure criterion which is derived for an isotropic material. The implementation of maximum principal stress criterion on the XFEM simulation is very limited for composite structures due to the complex combination of matrix and fibre failure.

Therefore, in the current work, the numerical simulation of composite wing subjected to sudden gust load is conducted using XFEM with a more accurate damage model developed specifically for a composite material: A three-dimensional (3D) Puck failure criterion [10, 11]. Since the 3D Puck failure criterion was developed specifically for a composite material, each failure modes on fibre and matrix are evaluated independently. This approach allows the 3D Puck failure criterion to more accurately model the failure modes in composite structures.

Additionally, in 3D Puck failure criterion, when the matrix failure occurs on the composite structures (failure exposure calculation result on the inter-fibre failure is ≥ 1), further iterations are performed to determine the orientation angle of the fracture plane. The fracture plane represents the crack that splits the matrix during matrix failure in composite materials, while the orientation angle indicates the direction of crack propagation. However, the calculation of the orientation angle of the fracture plane still uses a linear search algorithm, which requires a considerable amount of computational time. For this reason, a new fast search algorithm is developed and implemented to replace the linear search algorithm in 3D Puck failure criterion. This algorithm aims to improve the computational time by reducing the number of iterations while improving the accuracy of the results.

1.2 Research Gaps

Several research gaps have been identified after reviewing previous studies related to numerical simulation on composite structure under aeroelastic loading. However, to limit the scope, the research gaps that will be addressed for improvement in this thesis are as follows:

- The current XFEM simulation of crack propagation in composite wing subjected to gust load implements the maximum principal stress damage criterion, which is inadequate for accurately simulating damage in composite materials.
- There is a limited number of built-in damage models for composite materials in current commercial finite element analysis (FEA) software, which are unsuitable for implementation in 3D finite element model.
- The proposed composite material damage model utilises a conventional linear iteration algorithm, which significantly affects the computational performance of damage calculations in composite materials.

1.3 Research Objectives

The main objective of this research is to implement a more advanced composite damage model into the XFEM framework to assess the structural integrity of composite structures subjected to sudden gust loads, thereby improving the accuracy of computational simulations. However, due to the proposed composite damage model complexity, the computational performance of the numerical simulation will be severely impacted. Therefore, a novel fast iteration algorithm is also implemented to reduce the required iteration for the composite damage model calculation. In this research, the numerical simulation is conducted using Abaqus FEA software, utilising the built-in XFEM module and the software expandability through user-defined subroutine, which significantly reduces the complexity of developing, improving and implementing the advanced composite damage model into the XFEM framework.

In summary, the current work aims to achieve the following objectives:

- Expand the capability of the Abaqus FEA software to accurately model the damage progression on the composite structures by developing a user-defined 3D Puck composite failure criterion subroutine.
- Improve the algorithm on the 3D Puck composite failure criterion subroutine in finding fracture plane orientation angle for matrix failure to reduce computational time while maintaining the reliability of the results using an improved fast fracture plane search algorithm.
- Integrate the user-defined 3D Puck composite failure criterion subroutine into the XFEM framework to simulate crack propagation in the composite structure subjected to the gust load.

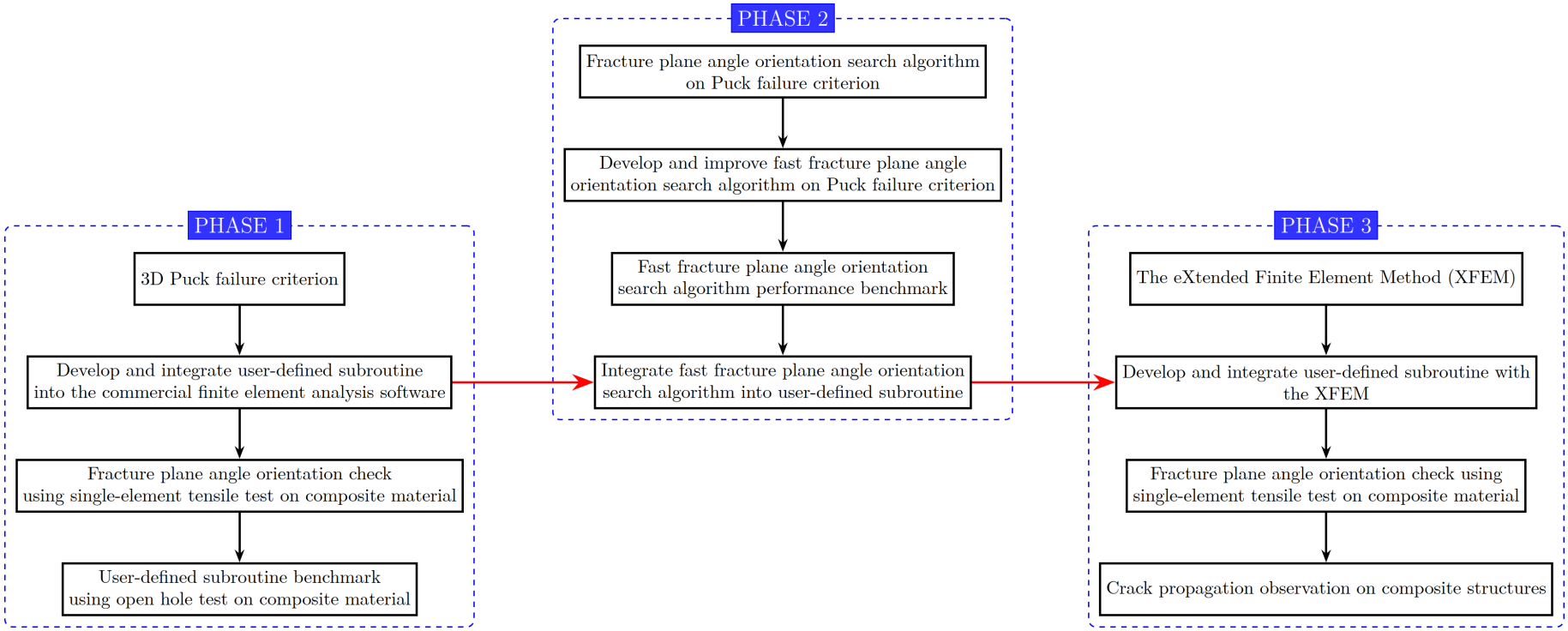


Figure 1.3: The research flow diagram

1.4 Research Work Plan

In the course of conducting this research, a researh flow diagram has been created to emphasise several key processes, as illustrated in Figure 1.3. The research is compartmentalised into three phases to structure the project into manageable components with attainable milestones.

The first phase of the research focuses on developing a user-defined subroutine to address the limited material model library for composite materials in Abaqus FEA software. The constitutive equations for the composite material are derived and implemented from the ground up into a user-defined subroutine. The 3D Puck failure criterion is employed to evaluate the fibre and matrix failure within the composite material. Additionally, an exponential damage degradation model is integrated to simulate the reductions in stiffness and strength of the composite material following failure, due to damage accumulation.

At the end of phase one, a single-element tensile test is employed to validate the orientation angle of the fracture plane upon matrix failure. Additionally, a numerical simulation of an open-hole test is conducted to compare the results with those obtained from the open-hole experiment test.

The second phase of the research is centred on developing a novel fast search algorithm to replace the existing linear search method for determining the orientation angle of the fracture plane during matrix failure in the 3D Puck failure criterion. The reliability and computational efficiency of the proposed fast search algorithm are then compared with the current linear search method and other fast search algorithms. Subsequently, the newly proposed fast search algorithm is integrated into the user-defined 3D Puck failure criterion subroutine developed in the first phase to improve the computational efficiency of the current user-defined subroutine.

The final phase of the research concentrates on the development, integration, and implementation of a user-defined subroutine of the 3D Puck failure criterion into the built-in XFEM framework of Abaqus FEA software. A single-element tensile test is conducted to verify the user-defined subroutine integration into the XFEM framework. Later, the user-defined subroutine is employed to simulate a composite wing subjected to sudden gust loads, in order to observe crack formation and potential crack propagation.

1.5 Thesis Outline

This thesis is organised into six chapters, which summarise the methods, results, and discussions of the research conducted. A brief description of each chapter is presented below:

- **Chapter 1** introduces the background, motivations, research gaps, research objectives and workflow of the current research.
- **Chapter 2** presents a review of current research on aerofracture elasticity, a brief overview of the Extended Finite Element Method (XFEM) algorithm and the development of failure calculations for composite materials.
- **Chapter 3** details the development of progressive damage model of a composite material that incorporates 3D Puck failure criterion with an exponential damage degradation model. The composite material model is then implemented into user-defined subroutine and integrated into the Abaqus FEA software, where it is validated and compared against experimental results.
- **Chapter 4** discusses the comparison between currently available fast search algorithm for determining the fracture plane orientation angle on 3D Puck interfibre failure criterion. A new improved fast fracture search algorithm is proposed to improve the reliability of the fast fracture search algorithm. Later, the performance of the currently available fast search algorithms and the conventional search algorithm are compared with the proposed fast search algorithm.
- **Chapter 5** presents the development and integration of a user-defined subroutine of the composite material model from the previous chapter into the XFEM framework. The newly integrated user-defined subroutine is then employed to simulate a composite wing subjected to sudden gust loads to assess the structural integrity of the composite material.
- **Chapter 6** summarises the findings and contributions of the current research to the field of aerofracture elasticity.

Chapter 2

Literature Review

In this chapter, a review on the published works that related to the current research are presented. The review encompassed several different topics which separate into four section, i.e.:

- current state of research on aerofracture elasticity,
- brief review on the Extended Finite Element (XFEM) and
- development of failure criteria calculations for composite materials.

2.1 Aerofracture Elasticity

In an aircraft, the flight operating conditions may exert various combined load cases, i.e., due to the interaction of air and structures. Airflow passing the aircraft structure generates aerodynamic loads, i.e., lift and drag, which further causes structural deformation. Aircraft structure is known to be lightweight and relatively flexible in comparison to the other vehicles's structures, i.e., train, ship. One of the main components in an aircraft is its wings. The wings are aerodynamically shaped, constructed of airfoil sections, which mainly functioning as a component to produce lift. A two-way coupling between the airflow and structure occurs as the aerodynamic loads create structural deformation, and further the deformation also changes the wing shape; thus, changes the aerodynamic loads distribution. In the aerospace engineering field, the interaction between airflow and structure is called as aeroelastic phenomenon [12].

Some of the critical conditions in aircraft may caused by the aeroelastic instability. In an aircraft, the critical speed of the instability, i.e., flutter and divergence, must be considered in the design of the maximum speed and part of the certification process. Flutter and divergence may cause catastrophic structural failure. Divergence is a static aeroelastic problem, in which the aerodynamic forces overcome the restoring moment causing the deformation to increase significantly. Meanwhile, at a particular flight speed, known as the flutter speed, if the aircraft experiencing disturbances, i.e,

gust and turbulence, the interaction between airflow and elastic structures may cause undamped oscillation and modes coupling, e.g., wing bending and torsion [13].

Figure 2.1: The damaged F-117A before crash at Chesapeake Air Show [14]

In the modern era, there are still some recorded incidents concerning flutter and divergence in aircraft. One of the infamous incidents is the crash of F-117A Nighthawk (Figure 2.1) due to the flutter occurred on its elevon, parts of the control surface at the wing [14]. The excessive fluttering caused the elevon detached and hit the aircraft's tail. Thus, created control problem and the aicraft crashed to the ground [15]. Another incident could be found in the crash of NASA's Helios [16]. The aircraft excited by a turbulence and resulted in a persistently high dihedral, caused divergent pitch which led to structural failure (Figure 2.2).

Another incident is reported by the National Transportation Safety Board on a crash in an air race in Reno, Nevada [17]. One of the aircraft, a North American P-51D, *The Galloping Ghost*, was losing control and collided with the spectator box, causing fatalities to the pilot and people on the ground (Figure 2.3). Based on NTSB investigation, it was found that flutter happened at the aircraft's racing speed. There was evidence of fatigue crack on one of the screws of the elevator. Thus, the structural stiffness of the elevator decreased, and the flutter onset compromised, the flutter occurred at a lower speed than its design. The flutter generated excessive dynamic loads and resulted in further structural damage, bending fracture.

(a) Helios during normal flight

(b) Helios experiencing high wing dihedral

(c) Helios disintegrating over Pacific

Figure 2.2: NASA's Helios unmanned aerial vehicle [18]

The incident in Reno Air Races depicts further complication of aeroelastic instability with fracture mechanics. In this case, a crack on a screw caused flutter which led to a more massive fracture. This phenomenon highlighted the importance on how fracture could affect the flutter boundary, in this case whether the aircraft will fail due to the damage growth or the critical flutter speed reduced and structural failure due to flutter may occur at aircraft's operating speed.

One of the earliest studies on a dynamic behaviour of structure with crack is discussed by Viola et al. [19]. In their paper, Viola et al. developed a special beam element to detect a cracked steel structure via numerical method and modal analysis data. Further, Viola and Marzani [20] investigated the effect of crack depth and location on the dynamic stability of beams. Local flexibility on a cracked member was modelled via a line-spring model. The results showed that the crack could affect that transition from divergence to flutter, or from flutter to divergence. Viola et al. [21] extended the cracked beam element formulation to evaluate double cracked beams which claimed could be applied for multi-cracked beams. To be noted, aerodynamic loads were not modelled in the formulations proposed by [20, 21], instead a form of distributed loads were introduced to the models. In addition, the formulations were developed for beams made of isotropic material.

Figure 2.3: P-51D Mustang, *The Galloping Ghost*, lost its elevator trim tabs during Reno Air Races [17]

In the application of lifting structures, Chen and Lin studied the effect of crack on the aeroelastic instability of a thin panel [22]. Finite element was employed to model the plate structure. Bending plate element was used with a so-called hybrid crack element. The crack element was a modified element with stress intensity factors of the bending modes. The unsteady aerodynamic loads was modelled using piston theory, which is applicable for supersonic and hypersonic speed. The aeroelastic instability was solved in the frequency domain analysis. The results depicted that the instability, i.e., flutter, was achieved faster with the presence of crack. Since then, the interest on the crack effect on aeroelastic instability was growing more in concern of composite structures.

Lin et al. [23] proposed a finite element formulation to evaluate the aeroelastic instability of a composite panel. A high Degree of Freedom (DoF) element was used to model the crack. Piston theory was applied for the unsteady aerodynamic model. The frequency domain analysis was conducted to obtain the flutter solution. In general, the flutter boundaries were found to be reduced with the existence of crack, except at specific orientations the flutter speed were not changed much.

Strganac and Kim investigated panel flutter and bending-torsion flutter of a wing which represented as composite plate [24]. The crack was introduced as damage which calculated as a factor which reduced the material's elasticity modulus. The piston theory was also used to model the aerodynamic loads. Flutter solutions were obtained in a time domain analysis concerning a gust loads with harmonic pattern. From their

study, the flutter boundary was found to be reduced with the presence of damage as its reduce the structural stiffness.

A study by Pidaparti also used a crack model as a modification to the stiffness the composite material [25]. The composite structures were modelled by finite element with classical laminate theory. The flutter solutions were evaluated in a frequency domain analysis for supersonic regime with aerodynamic piston theory. Based on their results, the eigenvalues or the natural frequencies of fundamental modes were reduced and the coalescence of modes was achieved faster, means the flutter speeds was lower, with the existence of damage. Pidaparti and Chang, further modelled an edge crack with double nodes or duplicated nodes on the neighboring elements [26]. The study highlighted the effect of fiber orientation, which despite generally there were reduction in flutter boundary, for particular orientations there were insignificant change.

As highlighted by the studies in [22, 23, 24, 25, 26], generally, the presence of cracks reduced the stiffness of materials; hence, the flutter boundaries were reduced. However, important to note that these studies only valid in supersonic regime. The article by Wang et al. discussed the effect of fiber orientation, crack length and crack location for damaged composite plate in subsonic regime [27]. The composite plate was modelled using a one-dimensional beam and the aerodynamic load was evaluated using two-dimensional quasi-steady strip theory. In this case, the crack also represented by the changes in composite's local flexibility matrix. The flutter solutions were obtained in a frequency domain analysis. Interestingly, from their results, for certain fiber orientations, the flutter speed could increase with the existence of crack. However, Wang et al. noted that eventually the crack could grow and lead to structural failure. Thus, the monitoring of the crack growth will be critical.

For the first time, a wingbox model was investigated in [28]. The wingbox structure was modelled with laminated shell elements. Georgiou et al. also used an approach to model the damage as the reduction of material properties. Time domain analysis with 1-cos gust disturbance in subsonic regime was conducted. A commercial finite element-based software was used to solve the flutter analysis. In this case, some case studies were evaluated; however, the overall analysis on the effect of damage to the aeroelastic instability was performed using probabilistic method, i.e., Polynomial Chaos Expansion (PCE). It was found that the mean flutter speed value was significantly reduced while the maximum stress was increased with the existence of damage.

Abdullah et al. also modelled the composite structures by means of laminated shell elements [29]. Similar unidirectional composite configurations as given in [27] were evaluated. In this case, the edge crack was modelled using similar approach with [26], as duplicated nodes on the neighboring element. Commercial finite element model was also used to solve the flutter solution with Doublet Lattice Method (DLM) and Strip Theory. The frequency domain analysis was conducted to obtain the complex eigenvalues

of the aeroelastic system for a certain range in subsonic regime. The models with DLM gave slightly higher flutter speeds in comparison to the ones with Strip Theory. Moreover, for a specific configuration of crack length and location, they also found that the flutter speed could be higher than the undamaged structure.

In recent article, Varun et al. [30] also used a similar strategy with [29]. In this case, delamination was modelled instead of crack. Similar conclusion was found as the flutter speed of the plate reduced considerably due to the presence of delamination. Despite the elaborated efforts in the study of aeroelastic-crack interaction of composite, there is a lack of study on the crack or damage growth under aeroelastic loads. Most of the previous case studies were static crack or static damage and some of the models included the damage as part of reduced material properties.

One of the few studies on the crack growth on composite structure under aeroelastic loads is discussed in Abdullah et al. [8]. Similar to the one in [8], the flutter solutions of composite plates with static crack were conducted in a commercial software. However, they extended the computational process with another software with the ability to model crack propagation via XFEM. This procedure has been proposed in Wirawan et al. [5] to evaluate the crack propagation of a metal wingbox model under aerodynamic loads.

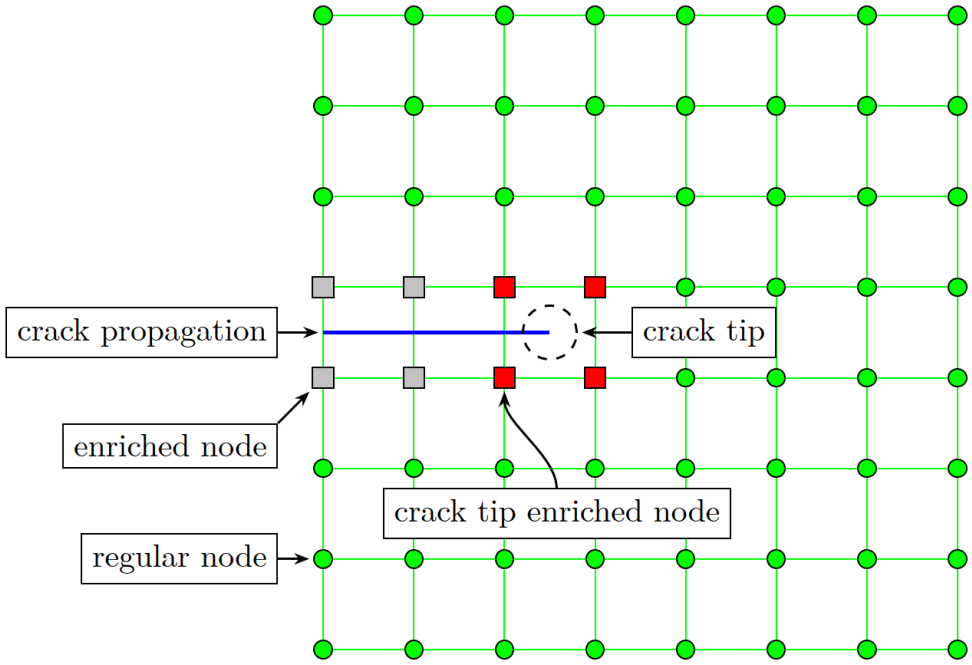
Abdullah et al. [8] conducted the time domain analysis to evaluate the structural responses due to gust load disturbances at certain cruise speeds. In this case, it was found that for a particular set of orientation, crack length and location, the crack could propagate and led to failure even before the flutter speed was achieved. Another study by Hoseini et al. also applied XFEM to evaluate the crack propagation on a laminated composite structure under aerodynamic loads [9]. In their study, Hoseini et al. proposed a unique combination of 1D elements and 3D elements to reduce the computational cost. One of the limitations of XFEM module in commercial software, often, that it is only applicable to 3D elements and not available for 2D shell elements. Hence, if a full 3D elements are used to model the structure, the computational time could increase significantly.

Both Abdullah et al. and Hoseini et al. utilise the maximum stress failure criterion within the XFEM framework to assess the structural integrity of composite structures. However, this criterion does not account for stress interactions from other directions due to the Poisson effect and also different failure modes in composite materials. Therefore, in this research, a more appropriate failure criterion is proposed to replace the maximum stress failure criterion.

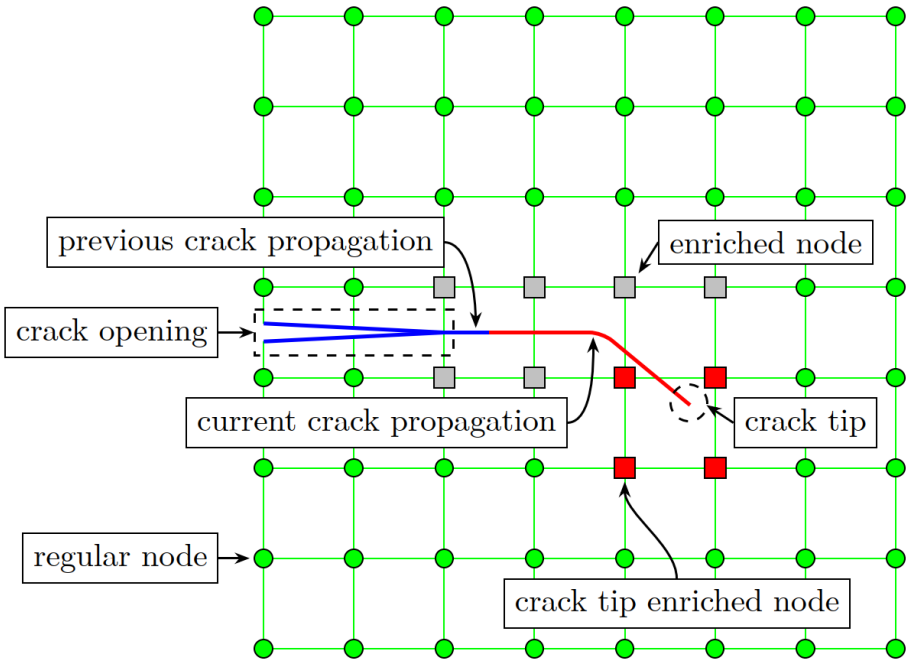
In the next section, a brief review of the XFEM, the development of failure criteria for composite materials and some XFEM implementation on the composite materials are presented.

2.2 The Extended Finite Element Method

The Extended Finite Element (XFEM) is an advanced technique in finite element analysis that detects discontinuities within the elements and uses an enrichment function to augment the existing mesh around the discontinuities. Belytschko and Black [6] initiated the development of the XFEM for modelling crack in 2D finite element cases, utilising partition-of-unity properties, where the sum of the shape functions is equal to one. These partition-of-unity properties were identified by Melenk and Babuška [31] and Duarte and Oden [32]. Moës et al. [7] later generalised the Belytschko and Black previous XFEM function. This was achieved by employing the combination of the asymptotic function derived by Fleming et al. [33] to enrich the element near the crack tip and discontinuous Haar function to enrich the element for the remainder of the crack. Afterward, Sukumar et al. [34] sucessfully expand the XFEM equations, enabling the XFEM application to 3D finite element cases. In addition to modelling regular cracks, the XFEM can also be used to represent interface separations between materials, as demonstrated by Sukumar et al. [35].



(a) The crack is initiated and propagate when the failure criteria value is achieved



(b) The crack advance into the next element, a new crack tip enriched nodes are applied on the element where the crack tip reside. The previous crack split the elements which leads to a crack opening.

Figure 2.4: Crack propagation on the finite element model using XFEM

In the finite element method, the displacement field on a regular node is approximated using the following equation:

$$u^h = \sum_{i=1}^{k} u_i \phi_i \tag{2.1}$$

where u^h is the finite element approximation associated with the mesh, while u_i represents the displacement field and ϕ_i denotes the bilinear shape function at node i, respectively. While in the XFEM, the Equation (2.1) is expanded into:

$$u^h = \sum_{i=1}^{k} u_i \phi_i + \sum_{j=1}^{k} b_j \phi_j H(x) + \sum_{i=1}^{k} \phi_k \left(\sum_{l=1}^{4} c_k^l \gamma_l(x) \right) \tag{2.2}$$

The b_j indicates the displacement field for the enriched node term, while and c_k^l represents the displacement field for the crack tip enriched node term. The function $H(x)$ denotes the discontinuous Haar function that is used on the enriched node:

$$H(x) = \begin{cases} 1 & \text{if the crack above the current node} \\ -1 & \text{if the crack below the current node} \\ 0 & \text{if the crack lies in the current node} \end{cases} \tag{2.3}$$

Whereas the function $\gamma_l(x)$ express asymptotic function that is used on the crack tip

enriched node:

$$\gamma_1(x)(r,\theta) = \sqrt{r}cos\left(\frac{\theta}{2}\right)$$
$$\gamma_2(x)(r,\theta) = \sqrt{r}sin\left(\frac{\theta}{2}\right)$$
$$\gamma_3(x)(r,\theta) = \sqrt{r}sin\left(\frac{\theta}{2}\right)sin(\theta)$$
$$\gamma_4(x)(r,\theta) = \sqrt{r}cos\left(\frac{\theta}{2}\right)sin(\theta) \tag{2.4}$$

These enrichment functions on the XFEM reduce the need of remeshing on the finite element model, thereby allowing for faster and more accurate simulation of crack propagation. Figure 2.4 illustrate the finite element model with enriched node surrounding the crack propagation and the crack tip. Using Equations (2.2) to (2.4), Giner et al. [36] successfully developed a subroutine to implement XFEM framework into the commercial FEA software to simulate crack growth on an open-hole aluminium plate and validate the result using experimental test.

In the XFEM, a crack initiates when a specific threshold value is reached. This value is determined by the result of the failure criterion calculation. Therefore, an accurate representation of the failure criterion is essential for effectively simulating crack propagation using XFEM in FEA software.

However, the current implementation of XFEM in FEA software is limited to modeling a single crack path in the finite element model. Additionally, the built-in XFEM function in commercial FEA software includes only a limited set of failure criteria, such as maximum principal stress/strain, maximum nominal stress/strain, and quadratic nominal stress/strain failure criteria. Thus, if a custom or more complex combination of failure criterion is employed, a separate subroutine must be developed independently.

For instance, to simulate damage in anisotropic materials, Feerick et al. [37] combine a modified Hashin failure criterion with the maximum principal stress criterion to simulate the crack propagation during screw pull-out in the cortical bone tissue. The modified Hashin failure criterion is used to model the crack propagation along each orthogonal axis, while the maximum principal stress criterion is used to model the crack propagation in the off-axis direction. As a result, a subroutine is developed to accomodate the complex combination of Hashin failure criteria and maximum principal stress criterion. Another example is provided by Van Dongen et al. [38], which use a custom LaRC05 [39] failure criterion in XFEM to simulate an open-hole experimental test.

In the following section, the development of the failure criteria on the composite materials are presented. The improvement on each composite failure criteria are explained. The rationale for selecting a particular failure criterion for the current XFEM

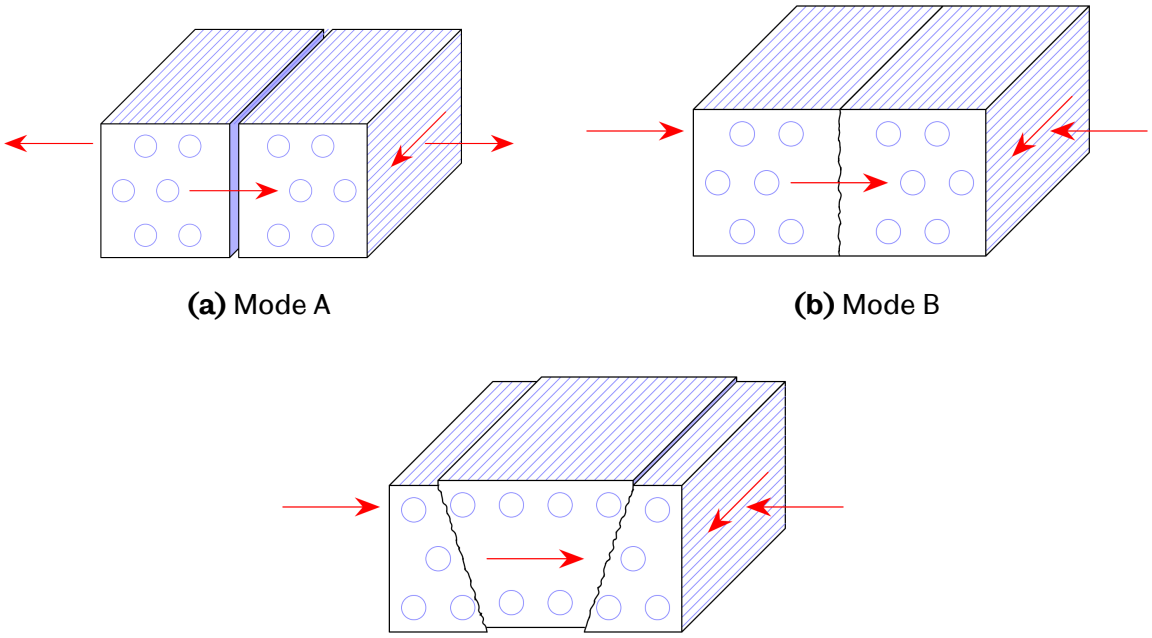
simulation of crack propagation in composite wing subjected to gust load is also explained in the following section.

2.3 Failure Criteria on Composite Material

The increasing use of composite materials has catalysed extensive research into failure criteria for the past few decades [10, 40, 41, 42, 43, 44, 45, 46]. These failure criteria are established based on the analysis of lamina failure in composite materials, utilising different approaches to achieve an accurate representation of intra-laminar failure within the composite materials.

Initially, composite failure criteria employ a scalar-valued function to quantify damage across the entire lamina without differentiating between fibre and matrix failure. In these criteria, lamina failure in composite materials is modelled by fitting experimental data using a general quadratic expression. The failure criteria proposed by Tsai and Wu [40], Tsai and Hill [41], Azzi and Tsai [47], Hoffman [48], and Chamis [49] are examples that utilise this approach.

In maximum stress or strain failure criteria, the failure indices for the fibre and matrix are calculated separately to identify failure modes in composite structures. These failure criteria calculates the failure index of the composite materials by comparing the strength of the lamina's fibre and matrix in each longitudinal, transverse, and out-of plane direction against the stresses or strains induced by the current loading conditions. However, these failure criteria do not account for the interaction between different stresses or strains acting on the lamina, which can lead to inaccuracies under multiaxial stress loading conditions.



(a) Mode A

(b) Mode B

(c) Mode C

Figure 2.5: Fracture modes on 2D plane stress Puck failure

Hashin and Rotem [50] enhanced the maximum stress failure criterion by developing a 2D failure criterion that incorporates interactions between stresses within the lamina specifically for matrix failure mode. Later, Hashin [51] expanded the 2D failure criterion to a 3D failure criterion, incorporating a quadratic approach for the matrix failure mode and integrating shear stress into the tensile failure calculation for the fibre.

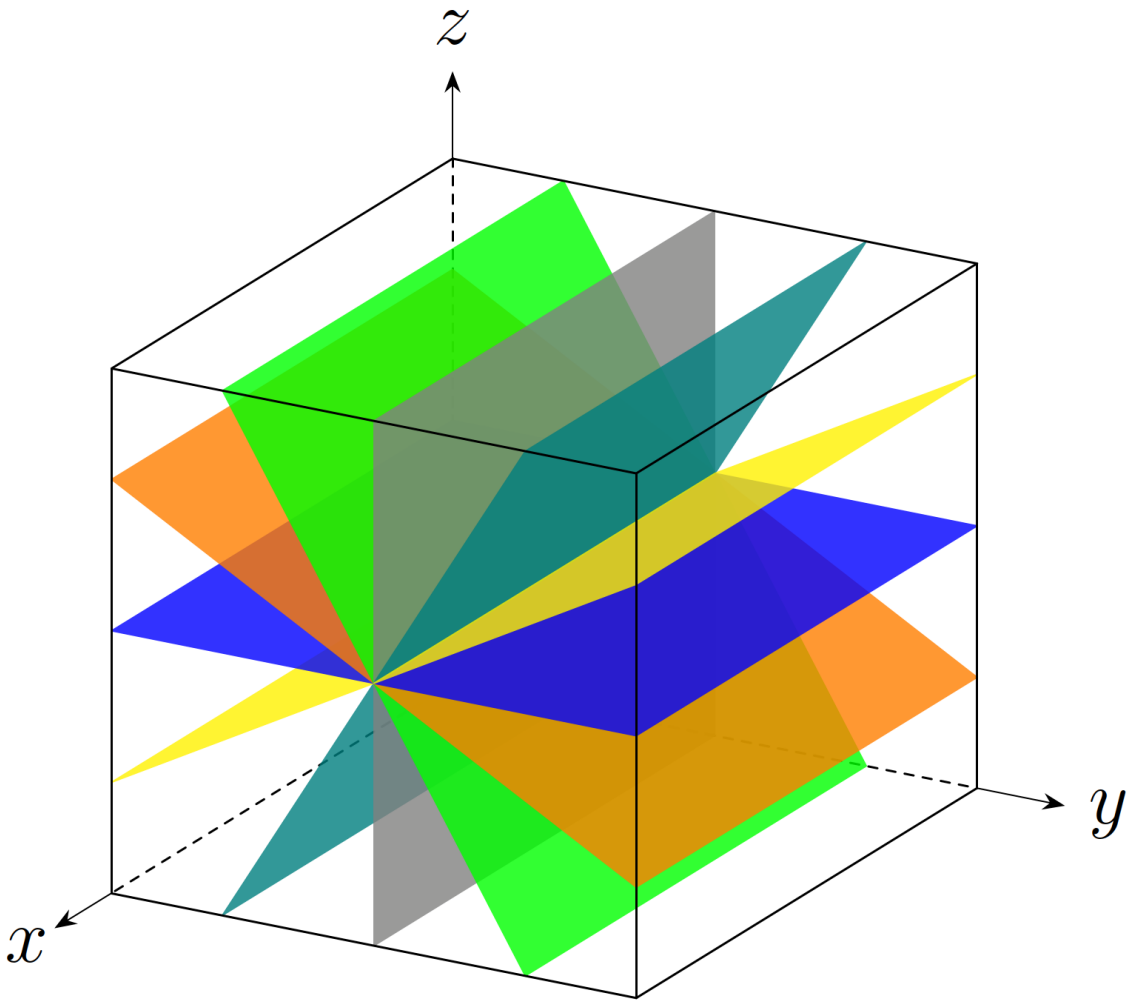


Figure 2.6: Example of potential fracture planes with $30°$ interval on a single-element on a finite element analysis model

Puck [10, 11] refines the Hashin failure criterion by integrating the transverse and shear stress components within the stress tensor into the failure criterion calculation for both fibre and matrix failures. Furthermore, based on experimental observations, Puck introduced the concept of the fracture plane to improve the matrix failure calculations. When the conditions for matrix failure are met, a fracture plane forms, splitting the matrix at a specific angle. For a thin plate, which can be assumed to be a 2D plane stress case, a tensile load will result in a fracture plane perpendicular to the direction of the load (Figure 2.5a). In the case of a compressive load, two possible outcomes may occur. If the ratio between the compressive stress at failure and the transverse compressive strength of the material is less than 0.4, the fracture plane will be perpendicular to the direction of the load (Figure 2.5b). Otherwise, the fracture plane will form at a certain angle, which necessitates additional calculation to obtain the orientation angle of the fracture plane (Figure 2.5c).

In a 3D stress case, the fracture plane on the Puck failure criterion is determined by comparing the failure exposure values across each potential fracture plane (Figure 2.6). Since the Puck failure criterion employs a linear search algorithm, the accuracy of the fracture plane orientation angle is dependent on the number of potential fracture planes included in the calculation. For instance, to achieve an accuracy of 1° on the fracture plane orientation angle, 180 potential fracture planes with 1° interval has to be evaluated, which requires significant amount of computation time.

However, these calculations improve the accuracy of the failure criterion of the Puck failure criterion in predicting the behaviour of composite materials. As a result, during the World Wide Failure Exercise (WWFE) [52], the Puck failure criterion demonstrated high accuracy in reproducing failures in composite structures under various loading conditions. Therefore, in this research the 3D Puck failure criteria is chosen as the basis for evaluating the structural integrity of composite materials. Due to the limitations of the built-in failure criteria in the FEA software, the 3D Puck failure criterion is developed using a separate subroutine from the ground up and subsequently integrated into the FEA software. This subroutine is subsequently combined with the XFEM to enhance reliability when discontinuities, such as cracks, arise in finite element simulation.

Chapter 3

Progressive Damage Model on 3D Puck Failure Criterion Combined with Exponential Damage Degradation Model

Due to the limited built-in composite material failure criterion and damage model on the commercial FEA software, a user-subroutine containing custom failure criterion formula and damage degradation model is developed and embedded into the FEA software. In this works, the 3D Puck failure criterion combined with exponential damage degradation model is derived based on the progressive damage model and then implemented into the user-defined subroutine. Two finite element models are generated in the FEA software to demonstrate the capability of the custom failure criterion formula and damage degradation model to predict the behaviour of the composite structures under loading conditions. In the later chapter, the custom failure criterion formula and damage degradation model in the user-defined subroutine is integrated into the XFEM framework to simulate the crack propagation within a finite element model without the need for remeshing.

3.1 Progressive Damage Model

The progressive damage model is based on the fundamental principles of continuum damage mechanics (CDM) as proposed by Kachanov [53]. Drawing from thermodynamic concepts, Kachanov asserts that damage progression in structures is an irreversible process, which can be characterized by damage variables. A scalar damage

Part of this chapter has been published in the European Journal of Computational Mechanics, https://doi.org/10.13052/ejcm2642-2085.3332

variable is sufficient to model an isotropic damage; however, for a more complex damage quantification such as anisotropic damage, a tensor valued damage variables are required. For example, in Matzenmiller et al. [54], A four-rank damage variable is used to quantify the anisotropic damage on the 2D plane stress composite structures. Simo and Ju [55, 56] were the first to utilise this concept and incorporate it into a computational model to simulate concrete structures.

In the CDM, damage is associated with the initiation of micro-cracks within the model. These micro-cracks reduce the undamaged nominal cross-sectional area A into an effective area $\bar{A}$ which has less undamaged cross-sectional area. Consequently, this leads to a higher effective Cauchy stress $\bar{\sigma}$ to compensate for the actual Cauchy stress σ experienced by the undamaged model. By relating effective Cauchy stress $\bar{\sigma}$to the damage value, the actual Cauchy stress σ can be expressed using the following formula:

$$\sigma = (1-\delta)\bar{\sigma}; \qquad \sigma = M(\tilde{\delta}) : \bar{\sigma} \tag{3.1}$$

The Equation $(3.1)_1$ is used for isotropic scalar-based damage model, where δ ranges from 0 for undamaged model to 1, representing fully damaged model. The Equation $(3.1)_2$ is utilised for anisotropic tensorial-damage model, where the operator $M(\tilde{\delta})$ denotes the damage tensor and $\tilde{\delta}$ contains the set of all damage variables that is applied on the model.

Wagner and Balzani [57] decomposed the Helmholtz free-energy function $\Psi(\epsilon, \tilde{\delta})$ resulting in a constitutive equation that incorporates anisotropic damage variables:

$$\Psi(\epsilon, \tilde{\delta}) = \sum_{i=f,m1,m2} (1-\delta_{it})(1-\delta_{ic})\Psi_i^e \qquad \text{with} \qquad \Psi_i^e = \frac{1}{2}\epsilon : \mathbb{C}_i^e : \epsilon \tag{3.2}$$

The subscript i on the equation above represent the damage variables related to fibre (f) and matrix failures $(m1, m2)$. Whereas subscript t and c stands for tension and compression load, respectively. The Ψ_i^e stands for Helmholtz free-energy function for the elastic part of the material, while $\mathbb{C}_i^e$ is the elastic constitutive equation of the material. Deriving the Equation (3.2) with respect to strain ϵ the Helmholtz free-energy function becomes,

$$\begin{aligned} \frac{\partial \Psi}{\partial \epsilon} &= \sum_{i=f,m1,m2} (1-\delta_{it})(1-\delta_{ic})\frac{\partial \Psi_i^e}{\partial \epsilon} \\ \sigma &= \sum_{i=f,m1,m2} (1-\delta_{it})(1-\delta_{ic})\mathbb{C}_i^e : \epsilon \\ &= \sum_{i=f,m1,m2} (1-\delta_{it})(1-\delta_{ic})\bar{\sigma} \end{aligned} \tag{3.3}$$

3.1.1 Constitutive Equation

The property of the composite material at lamina level can be assumed as transversely isotropic material. Thus, according to Vogler et al. [58] the elastic constitutive

operator $\mathbb{C}_i^e$ can be expanded into,

$$\mathbb{C}^e = \frac{\partial^2 \Psi}{\partial \epsilon^2} = \lambda \mathbf{1} \otimes \mathbf{1} + 2\mu_T \mathbb{I} + \alpha(\mathbf{1} \otimes \mathbf{A} + \mathbf{A} \otimes \mathbf{1}) + 2(\mu_L - \mu_T)\mathbb{I}_A + \beta \mathbf{A} \otimes \mathbf{A} \tag{3.4}$$

The tensor $\mathbf{1}$ and $\mathbb{I}$ are second and fourth order identity tensor, respectively, with tensor $\mathbb{I}_A = \mathbf{A}_{im}\mathbb{I}_{jmkl} + \mathbf{A}_{jm}\mathbb{I}_{mikl}$. The vector $\mathbf{a}$ contains the information of the fibre direction of the composite lamina, with tensor $\mathbf{A} = \mathbf{a} \otimes \mathbf{a}$. If the direction of the fibre coincides with X_1-direction in Cartesian coordinate system, then the vector $\mathbf{a} = [1 \quad 0 \quad 0]^T$, then the elastic constitutive operator $\mathbb{C}^e$ on the Equation (3.4) can be rewritten as,

$$\begin{aligned}
\mathbb{C}^e &= \Sigma_i \mathbb{C}_i^e \qquad \text{with i = f, m1, m2} \\
&= \mathbb{C}_f^e + \mathbb{C}_{m1}^e + \mathbb{C}_{m2}^e \\
&= \begin{bmatrix} \mathbb{C}_{11}^e & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} + \begin{bmatrix} 0 & \mathbb{C}_{12}^e & \mathbb{C}_{13}^e & 0 & 0 & 0 \\ \mathbb{C}_{21}^e & \mathbb{C}_{22}^e & \mathbb{C}_{23}^e & 0 & 0 & 0 \\ \mathbb{C}_{31}^e & \mathbb{C}_{32}^e & \mathbb{C}_{33}^e & 0 & 0 & 0 \\ 0 & 0 & 0 & \mathbb{C}_{44}^e & 0 & 0 \\ 0 & 0 & 0 & 0 & \mathbb{C}_{55}^e & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \mathbb{C}_{44}^e & 0 & 0 \\ 0 & 0 & 0 & 0 & \mathbb{C}_{55}^e & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \\
&= \begin{bmatrix} \mathbb{C}_{11}^e & \mathbb{C}_{12}^e & \mathbb{C}_{13}^e & 0 & 0 & 0 \\ \mathbb{C}_{21}^e & \mathbb{C}_{22}^e & \mathbb{C}_{23}^e & 0 & 0 & 0 \\ \mathbb{C}_{31}^e & \mathbb{C}_{32}^e & \mathbb{C}_{33}^e & 0 & 0 & 0 \\ 0 & 0 & 0 & \mathbb{C}_{44}^e & 0 & 0 \\ 0 & 0 & 0 & 0 & \mathbb{C}_{55}^e & 0 \\ 0 & 0 & 0 & 0 & 0 & \mathbb{C}_{66}^e \end{bmatrix} \\
&= \begin{bmatrix} \lambda + 2\alpha + \beta + 4\mu_L - 2\mu_T & \lambda + \alpha & \lambda + \alpha & 0 & 0 & 0 \\ & \lambda + \alpha & \lambda + 2\mu_T & \lambda & 0 & 0 & 0 \\ & \lambda + \alpha & \lambda & \lambda + 2\mu_T & 0 & 0 & 0 \\ & 0 & 0 & 0 & \mu_L & 0 & 0 \\ & 0 & 0 & 0 & 0 & \mu_L & 0 \\ & 0 & 0 & 0 & 0 & 0 & \mu_T \end{bmatrix}
\end{aligned} \tag{3.5}$$

The invariant coefficients on the Equation $(3.5)_4$ in terms of modulus of Elasticity E, Poisson ratio ν and shear stress μ are:

$$\begin{aligned}
\lambda &= E_{22}(\nu_{23} + \nu_{31}\nu_{13})/D \\
\alpha &= E_{22}[\nu_{31}(1 + \nu_{32} - \nu_{12}) - \nu_{32}]/D \\
\beta &= E_{11}(1 - \nu_{32}\nu_{23})/D - E_{22}[1 - \nu_{21}(\nu_{12} + 2(1 + \nu_{23}))]/D - 4\mu_{12} \\
\mu_L &= \mu_{12} \\
\mu_T &= \mu_{23} \\
D &= 1 - \nu_{32}^2 - 2\nu_{13}\nu_{31} - 2\nu_{32}\nu_{31}\nu_{13}
\end{aligned}$$

Substituting the elastic constitutive operator $\mathbb{C}^e$ in Equation $(3.5)_3$ into Equation $(3.3)_2$,

$$\sigma = \sum_{i=f,m1,m2} (1-\delta_{it})(1-\delta_{ic})\mathbb{C}_i^e : \epsilon$$
$$= \begin{bmatrix} (1-\delta_f)\mathbb{C}_{11}^e & (1-\delta_{m1})\mathbb{C}_{12}^e & (1-\delta_{m1})\mathbb{C}_{13}^e & 0 & 0 & 0 \\ (1-\delta_{m1})\mathbb{C}_{21}^e & (1-\delta_{m1})\mathbb{C}_{22}^e & (1-\delta_{m1})\mathbb{C}_{23}^e & 0 & 0 & 0 \\ (1-\delta_{m1})\mathbb{C}_{31}^e & (1-\delta_{m1})\mathbb{C}_{32}^e & (1-\delta_{m1})\mathbb{C}_{33}^e & 0 & 0 & 0 \\ 0 & 0 & 0 & (1-\delta_{m2})\mathbb{C}_{44}^e & 0 & 0 \\ 0 & 0 & 0 & 0 & (1-\delta_{m2})\mathbb{C}_{55}^e & 0 \\ 0 & 0 & 0 & 0 & 0 & (1-\delta_{m1})\mathbb{C}_{66}^e \end{bmatrix} : \epsilon \quad (3.6)$$

with $(1-\delta_f) = (1-\delta_{ft})(1-\delta_{fc})$; $(1-\delta_{m1}) = (1-\delta_{m1t})(1-\delta_{m1c})$; $(1-\delta_{m2}) = (1-\delta_{m2t})(1-\delta_{m2c})$. Using Equation (3.6), the actual Cauchy stress σ can be obtained using damage variables δ and elastic constitutive operator $\mathbb{C}^e$.

3.1.2 3D Puck Failure Criterion

In the current progressive damage model, the 3D Puck failure criterion [10, 11] is employed to predict the failure point in composite materials. Puck utilises the failure exposure value (f_E) to assess the integrity of the composite materials by evaluating the ratio between the applied stress tensor and the material's strength. Two types of failure modes are calculated in Puck failure criterion, Fibre Failure (FF) and Inter-Fibre Failure (IFF).

Fibre Failure

The Fibre Failure (FF) in 3D Puck failure criterion incorporates the stress due to the Poisson's effect on the matrix ($\nu_{\perp\parallel}$) and the fibre itself ($\nu_{\perp\parallel f}$). Additional variable called magnification factor ($m_{\sigma f}$) is included to account for the fibre compacting effect due to compressive load. This magnification factor has different value depending on the composite type. According to the Deuschle and Puck [59, 60] the magnification factor value for Glass-Fibre Reinforced Polymer (GFRP) and Carbon-Fibre Reinforced Polymer (CFRP) are 1.3 and 1.1, respectively.

$$f_{E,FF+} = \frac{1}{R_\parallel^t}\left[\bar{\sigma}_{11} - \left(\nu_{\perp\parallel} - \frac{E_\parallel}{E_{\parallel f}}\nu_{\perp\parallel f}\right)(\bar{\sigma}_{22}+\bar{\sigma}_{33})\right] \quad (3.7)$$

$$f_{E,FF-} = \frac{1}{R_\parallel^c}\left[\bar{\sigma}_{11} - \left(\nu_{\perp\parallel} - \frac{E_\parallel}{E_{\parallel f}}\nu_{\perp\parallel f} m_{\sigma f}\right)(\bar{\sigma}_{22}+\bar{\sigma}_{33})\right] \quad (3.8)$$

The result is ultimately compared with either $R_\parallel^t$ for tensile loads or $R_\parallel^c$ compressive loads to determine the failure exposure related to fibre failure ($f_{E,FF}$). The terms $R_\parallel^t$ and $R_\parallel^c$ represents the composite material's strength in longitudinal direction under tension and compression, respectively. If the ratio of the modulus of elasticity of the composite in the longitudinal direction to that of the fibre, $\frac{E_\parallel}{E_{\parallel f}}$, is unavailable, this ratio can be approximated using the fibre volume fraction V_f.

Inter-Fibre Failure

The Puck Inter-Fibre Failure (IFF) criterion expands the work by Hashin's idea on matrix mode, which is based on Mohr failure theory [61]. According to Hashin, if the failure plane could be identified, the failure is only caused by the normal stress (σ_n), shear stress transverse to the fibre direction (τ_{nt}) and shear stress parallel to the fibre direction (τ_{n1}) acting on the oblique plane. Puck uses the normal (σ_n) and shear stresses (τ_{nt}, τ_{n1}) as inputs to calculate the failure exposure on IFF. The failure exposure on IFF is obtained by finding the maximum value of the failure exposure for every "action plane" that is rotated parallel to the X_1 axis on each inclination angle θ.

The normal and shear stresses on the action plane are derived from the stress tensor. The stress tensor is transformed according to the current inclination angle. The σ_n and τ_{nt} are derived from the stress tensor components σ_{22}, σ_{33}, σ_{23}, while τ_{n1} is derived from the stress tensors σ_{12},σ_{13} as shown in equations (3.9), (3.10), and (3.11). Figure 3.1 shows the normal (σ_n) and shear stresses (τ_{nt}, τ_{n1}) that are being transformed on the action plane at the inclination angle θ.

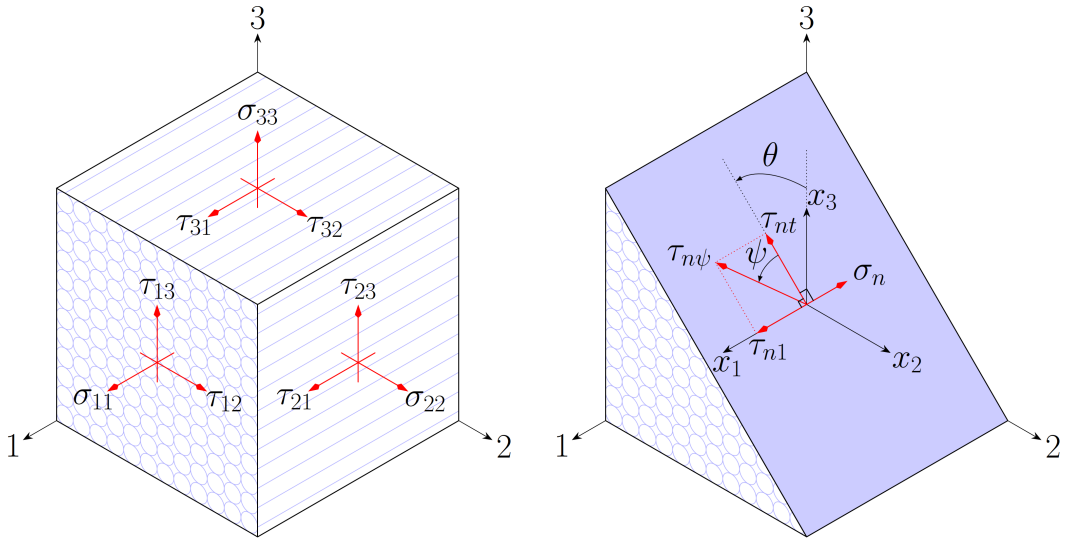


Figure 3.1: The stress tensor on an element (left) and the transformation of the stress tensor into the potential fracture plane with inclination angle θ on the same element with the top half is omitted for clarity (right)

$$\sigma_n(\theta) = \sigma_{22}\cos(\theta)^2 + \sigma_{33}\sin(\theta)^2 + 2\tau_{23}\sin(\theta)\cos(\theta) \tag{3.9}$$

$$\tau_{nt}(\theta) = -\sigma_{22}\sin(\theta)\cos(\theta) + \sigma_{33}\sin(\theta)\cos(\theta) + \tau_{23}(\cos(\theta)^2 - \sin(\theta)^2) \tag{3.10}$$

$$\tau_{n1}(\theta) = \tau_{31}\sin(\theta)^2 + \tau_{21}\cos(\theta)^2 + 2\tau_{23}\sin(\theta)\cos(\theta) \tag{3.11}$$

Puck states that the failure exposure value on the action plane at the current angle can be calculated using equation (3.12) for tension load ($\sigma_n \geq 0$) and equation (3.13) for compression load ($\sigma_n < 0$).

$$f_{E,IFF+}(\theta) = \sqrt{\left[\left(\frac{1}{R_{\perp}^{At}} - \frac{p_{\perp\psi}^{t}}{R_{\perp\psi}^{A}}\right)\bar{\sigma}_n(\theta)\right]^2 + \left(\frac{\bar{\tau}_{nt}(\theta)}{R_{\perp\perp}^{A}}\right)^2 + \left(\frac{\bar{\tau}_{n1}(\theta)}{R_{\parallel\perp}^{A}}\right)^2} + \frac{p_{\perp\psi}^{t}}{R_{\perp\psi}^{A}}\bar{\sigma}_n(\theta) \quad for\ \bar{\sigma}_n \geq 0 \tag{3.12}$$

$$f_{E,IFF-}(\theta) = \sqrt{\left(\frac{p_{\perp\psi}^{c}}{R_{\perp\psi}^{A}}\bar{\sigma}_n(\theta)\right)^2 + \left(\frac{\bar{\tau}_{nt}(\theta)}{R_{\perp\perp}^{A}}\right)^2 + \left(\frac{\bar{\tau}_{n1}(\theta)}{R_{\parallel\perp}^{A}}\right)^2} + \frac{p_{\perp\psi}^{c}}{R_{\perp\psi}^{A}}\bar{\sigma}_n(\theta) \quad for\ \bar{\sigma}_n < 0 \tag{3.13}$$

With each term in (3.12) and (3.13) is obtained using:

$$\frac{p_{\perp\psi}^{t}}{R_{\perp\psi}^{A}} = \frac{p_{\perp\perp}^{t}}{R_{\perp\perp}^{A}}\cos^2\psi + \frac{p_{\perp\parallel}^{t}}{R_{\perp\parallel}^{A}}\sin^2\psi$$

$$\frac{p_{\perp\psi}^{c}}{R_{\perp\psi}^{A}} = \frac{p_{\perp\perp}^{c}}{R_{\perp\perp}^{A}}\cos^2\psi + \frac{p_{\perp\parallel}^{c}}{R_{\perp\parallel}^{A}}\sin^2\psi$$

$$\cos^2\psi = \frac{\tau_{nt}^2}{\tau_{nt}^2 + \tau_{n1}^2}; \quad \sin^2\psi = \frac{\tau_{n1}^2}{\tau_{nt}^2 + \tau_{n1}^2}$$

$$R_{\perp\perp}^{A} = \frac{R_{\perp}^{c}}{2(1 + p_{\perp\perp}^{c})}$$

The $R_{\perp\perp}^{A}$, $R_{\perp\parallel}^{A}$, $R_{\perp\psi}^{A}$ are the fracture strength of the material. The $R_{\perp\perp}^{A}$ quantifies the fracture strength of the action plane due to shear stress. The $R_{\perp\parallel}^{A}$ is the shear strength that is obtained from the in-plane shear stress test. The $R_{\perp}^{t}$ and $R_{\perp}^{c}$ are the tensile and compressive strength of the material transverse to the fibre direction. The $R_{\perp}^{t}$ and $R_{\perp}^{c}$ are obtained using an uniaxial test on the material.

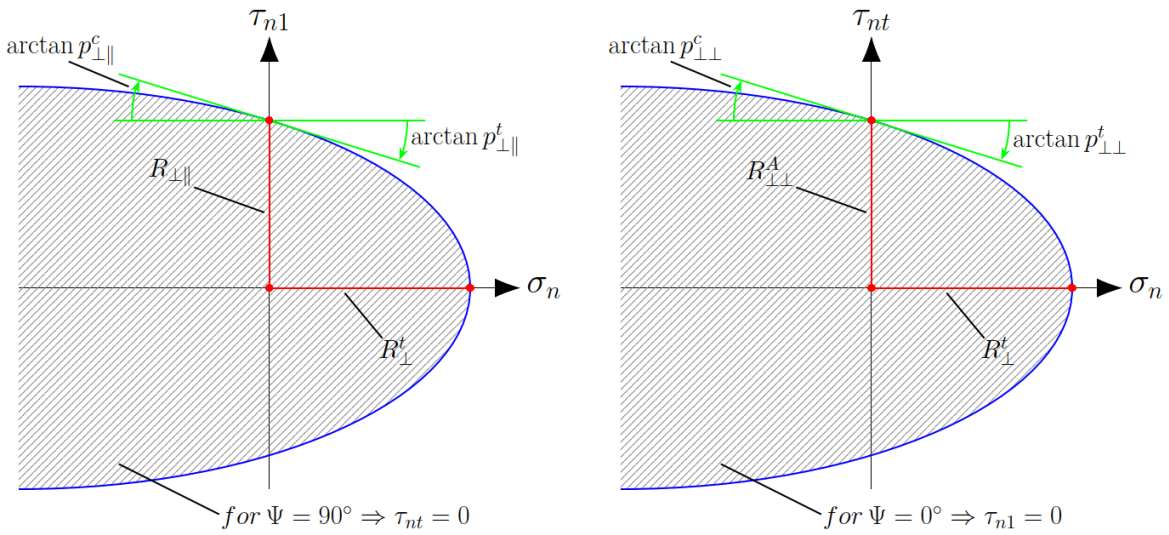


Figure 3.2: Master fracture body $\sigma_n - \tau_{n1}$ at $\psi = 90°$ (left) and $\sigma_n - \tau_{nt}$ at $\psi = 0°$ (right)

Table 3.1: Inclination factor for different types of composite material [10]

Material	$p^t_{\perp\perp}$	$p^c_{\perp\perp}$	$p^t_{\perp\parallel}$	$p^c_{\perp\parallel}$
GFRP	0.30	0.25	0.20	0.25
CFRP	0.35	0.30	0.25	0.30

The $p^t_{\perp\perp}, p^c_{\perp\perp}, p^t_{\perp\parallel}, p^c_{\perp\parallel}$ are the inclination factors for composite material. The values of the inclination factor depend on the type of the composite material. The inclination factor values are obtained from the slope of the master fracture body $\sigma_n - \tau_{n1}$ at $\psi = 90°$ and $\sigma_n - \tau_{nt}$ at $\psi = 0°$ when $\sigma_n = 0$ as illustrated in Figure 3.2. If the experimental data is unavailable, Puck provides the typical inclination factor values for the GFRP and CFRP in Table 3.1.

Each action plane at the corresponding angle will have its own unique failure exposure value. The angle of the action plane, when the failure exposure value is maximum, will define the fracture plane angle orientation of 3D Puck IFF for the current load. If the failure exposure value reaches 1 ($f_{E,IFF} \geq 1$), the material will have a permanent fracture plane at the corresponding angle.

Based on the numerical and experimental investigation by Correa et al. [62, 63], the stress component transverse to the fibre direction, $\bar{\sigma}_{33}$, influences the failure exposure on IFF ($f_{E,IFF}$). However, Reinoso et al. [64] suggested that the stress component parallel to the fibre direction $\bar{\sigma}_{11}$ affects the failure exposure on IFF. In the current 3D Puck failure criterion model, a weakening factor η_w is introduced to account the effect of $\bar{\sigma}_{11}$ on the calculation of failure exposure on IFF as proposed by Reinoso et al.

$$f_{E,IFF\pm} = \frac{f_{E,IFF\pm}}{\eta_w} \quad \text{with} \quad \eta_w = \sqrt{1 - \frac{(1 - f_{E,IFF})^2}{a^2}} \quad \text{and} \quad a = \frac{1 - s}{\sqrt{1 - m^2}} \tag{3.14}$$

The s and m values on the Equation $(3.14)_3$ are obtained from the Puck's elliptic stress plane $\bar{\sigma}_{11} - \bar{\sigma}_{22}$. Due to insufficient experimental data, a 0.5 value for both s and m is recommended [10, 11].

3.1.3 Damage Degradation Model

A damage degradation profile is introduced to simulate failure in composite materials by incorporating damage variables into the constitutive equation, resulting in a reduction of the stiffness and strength of the composite materials. This condition is activated when the damage threshold is equal to or greater than one ($r_E \geq 1$). The damage threshold is expressed into the following equation:

$$r_{E,n+1} = max\,(1, f_{E,n+1}) \tag{3.15}$$

where $r_{E,n+1}$ is the current damage threshold value and $f_{E,n+1}$ is the failure exposure value on the current iteration. The Equation (3.15) captures the phenomenon of damage irreversibility, as previously stated by Kachanov when proposing CDM.

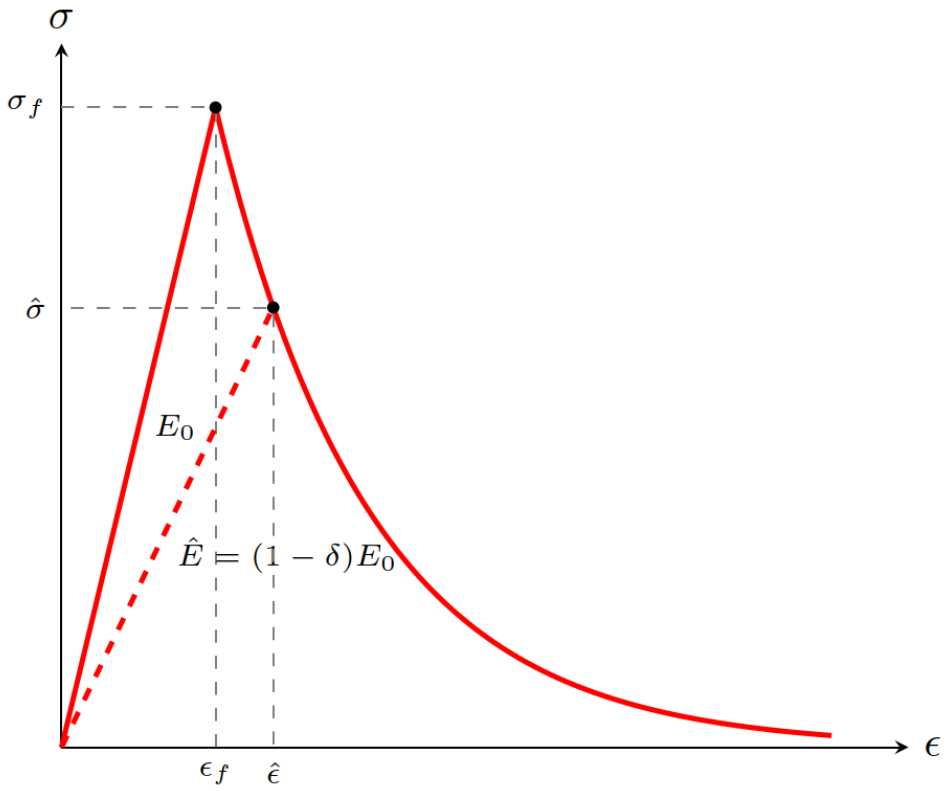


Figure 3.3: Illustration of the exponential damage degradation model

Various damage degradation profiles in the progressive damage model has been presented on several researches. For example, Falzon and Apruzzese [65, 66], and Lapczyk and Hurtado [67] and Eliopoulos et al. [68] employ a linear damage degradation model, while Maimí et al. [69, 70] and Reinoso et al. [64] utilise an exponential damage degradation model as illustrated in Figure 3.3. In this research, an exponential damage degradation model is designated as the damage degradation profile. The equations for the exponential damage degradation are expressed as follows:

$$\delta_{fj} = 1 - \frac{e^{[A_f^j(1-f_{E,FF})]}}{f_{E,FF}}; \text{ with } j = t, c \tag{3.16}$$

$$\delta_{mij} = 1 - \frac{e^{[A_{mi}^j(1-f_{E,IFF})]}}{f_{E,IFF}}; \text{ with } i = 1, 2;\ j = t, c \tag{3.17}$$

While the factors A_f^j and A_{mi}^j in the Equation (3.16) and (3.17) are:

$$A_f^j = \frac{2L_c R_\parallel^j}{2E_\parallel \mathcal{G}_\parallel^j - L_c(R_\parallel^j)^2};$$

$$A_{m1}^j = \frac{2L_c R_\perp^j}{2E_\perp \mathcal{G}_\perp^j - L_c(R_\perp^j)^2};$$

$$A_{m2}^j = \frac{2L_c R_{\perp\parallel}^j}{2G_{\perp\parallel}\mathcal{G}_{\perp\parallel} - L_c(R_{\perp\parallel}^j)^2}; \text{ with } j = t, c$$

With $\mathcal{G}_{\parallel}^{t}$ and $\mathcal{G}_{\parallel}^{c}$ are fracture energy in tension and compression parallel to the fibre direction, respectively. Whereas $\mathcal{G}_{\perp}^{t}$ and $\mathcal{G}_{\perp}^{c}$ are the fracture energy in tension and compression transverse to the fibre direction, respectively, while $\mathcal{G}_{\perp\parallel}$ is the in-plane shear fracture energy. The L_c stands for the characteristic length of the element on the finite element model.

3.1.4 Subroutine Algorithm

In order to simulate the composite materials using progressive damage model using 3D Puck failure criterion combined with exponential damage degradation model, a subroutine has been developed and integrated into the Abaqus FEA software via the UMAT (User MATerial) subroutine using an implicit scheme. In the Abaqus FEA software the subroutine is invoked at each Gauss/integration point for every time increment until the maximum time increment is reached. The flowchart of the Gauss/integration point iteration involving an UMAT subroutine in Abaqus FEA sofware is illustrated in Figure 3.4. For the 3D Puck failure criterion combined with exponential damage degradation model, the UMAT subroutine is illustrated in Figure 3.5 and the pseudocode is presented on the Algorithm 1.

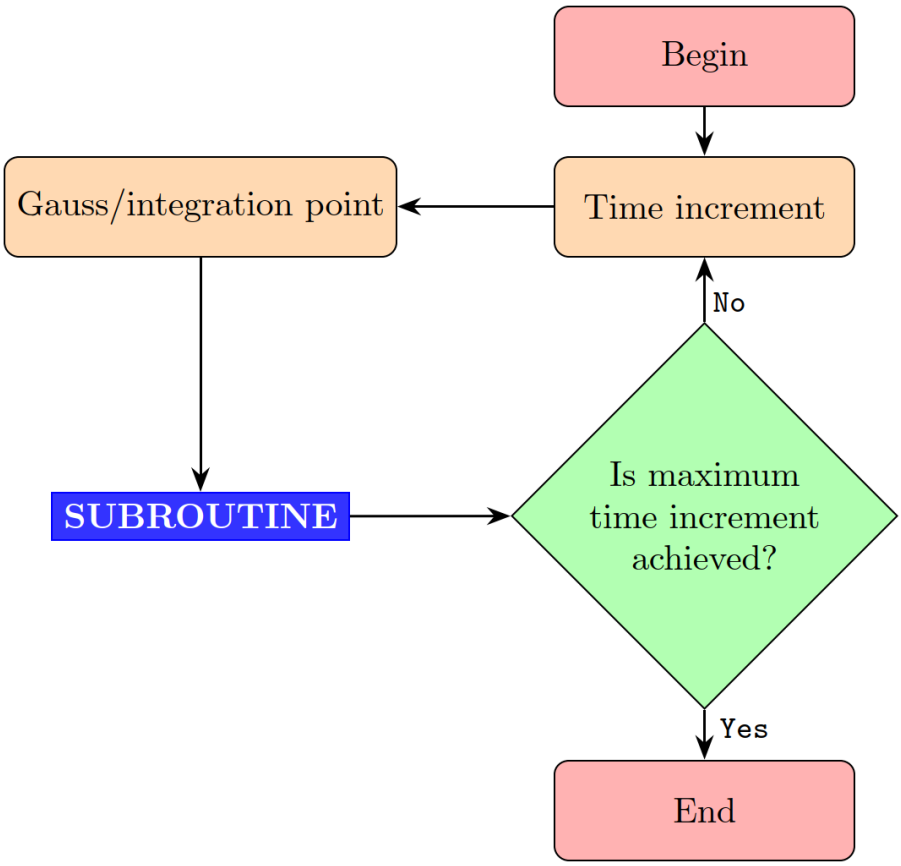


Figure 3.4: Flowchart of the time increment iteration on the Gauss/integration point in the Abaqus FEA software

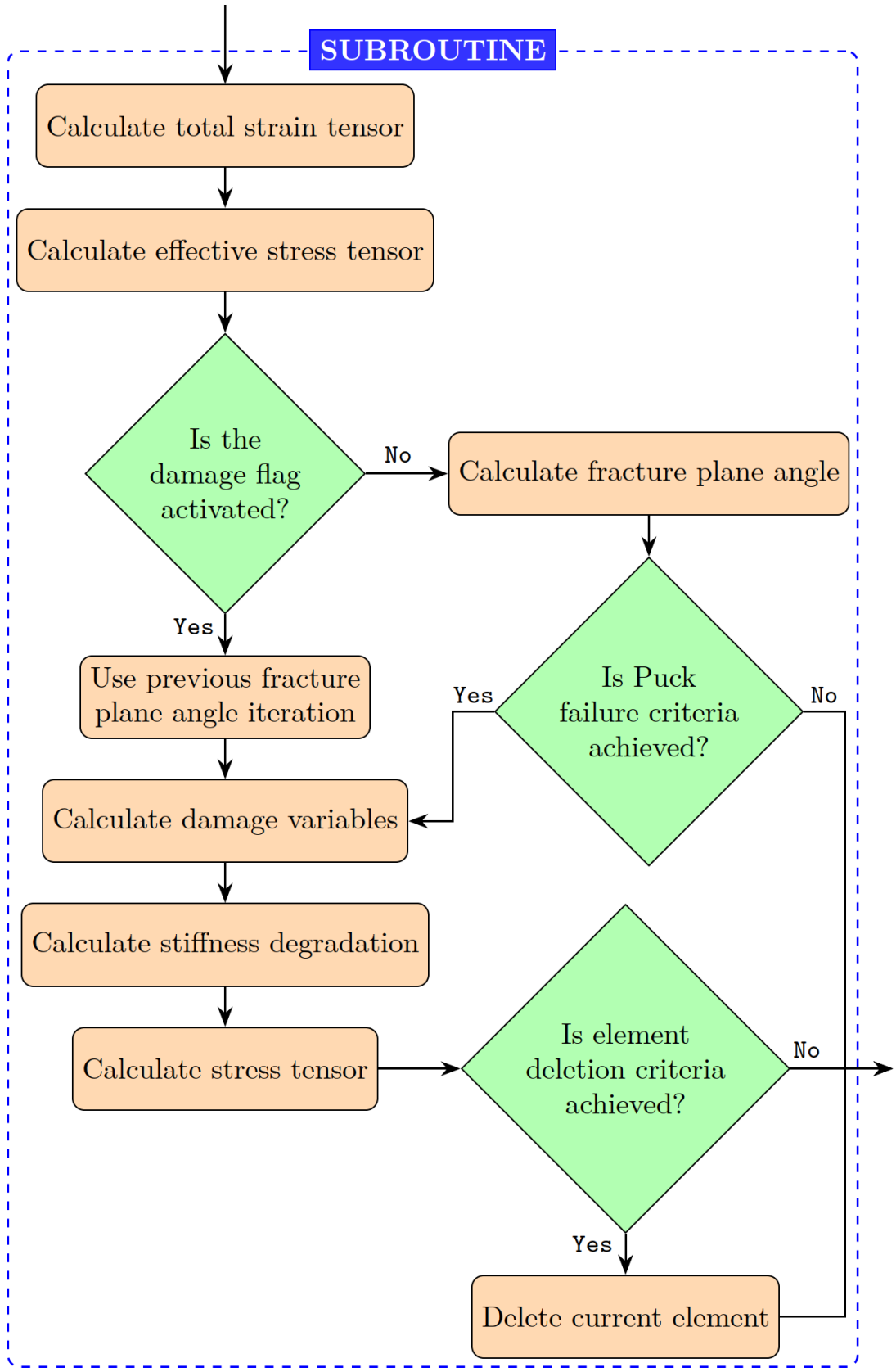


Figure 3.5: Flowchart of the UMAT subroutine containing 3D Puck failure criterion combined with exponential damage degradation model

Algorithm 1 Pseudocode of the UMAT subroutine on progressive damage model

Given the strain ϵ_n, stress tensor σ_n, damage value δ_n and damage threshold r_n data at the previous iteration n,

Update the new strain increment data $\Delta\epsilon_{n+1}$
Calculate the strain tensor $\epsilon_{n+1} = \epsilon_n + \Delta\epsilon_{n+1}$
Calculate the composite material's constitutive equation $\mathbb{C}^e$ [Eq.(3.5)]
Calculate the effective stress tensor $\bar{\sigma}_{n+1} = \mathbb{C}^e : \epsilon_{n+1}$
Calculate the failure exposure on FF criterion
IF $\bar{\sigma}_{11,n+1} \geq 0$ **THEN**
    Calculate $f_{E,FF+,n+1}$ [Eq.(3.7)]
ELSE
    Calculate $f_{E,FF-,n+1}$ [Eq.(3.8)]
END IF
Calculate $r_{FF,n+1}$ [Eq.(3.15)]
IF $r_{FF,n+1} > 1$ **THEN**
    Calculate $\delta_{fj,n+1}$ [Eq.(3.16)]
END IF
Calculate the failure exposure on IFF criterion
IF $r_{E,IFF+} == 1$ **AND** $r_{E,IFF-} == 1$ **THEN**
    Iterate the failure exposure $f_{E,IFF}(\theta)$ with 1° fracture plane angle θ_{fp} increment
    FOR $\theta_{fp,i+1} = 0, 180, 1$ (i+1 is used as the counter for the current for-loop) **DO**
        IF $\sigma_n \geq 0$ [Eq.(3.9)] **THEN**
            Calculate $f_{E,IFF+}(\theta)$ [Eq.(3.12)]
        ELSE
            Calculate $f_{E,IFF-}(\theta)$ [Eq.(3.13)]
        END IF
        Find the corresponding θ_{fp} for the highest $f_{E,IFF}(\theta)$ value
        IF $f_{E,IFF,i}(\theta)$ on the $\theta_{fp,i}$ < $f_{E,IFF,i+1}(\theta)$ on the $\theta_{fp,i+1}$ **THEN**
            Save the current increment $f_{E,IFF,i+1}(\theta)$
            Save the corresponding $\theta_{fp,i+1}$ as fracture plane angle θ_{fp}
        END IF
    END FOR
    Calculate $r_{IFF,n+1}$ [Eq.(3.15)]
    IF $r_{E,IFF,n+1} > 1$ **THEN**
        Calculate $\delta_{mij,n+1}$ [Eq.(3.17)]
    END IF
ELSE
    Use the $\theta_{fp,n}$ from the previous iteration as the fracture plane angle θ_{fp}
    Calculate $f_{E,IFF}(\theta)$, $r_{IFF,n+1}$ and damage $\delta_{mij,n+1}$
END IF
Update the relevant state dependent variable (SDV)
Calculate secant constitutive equation tensor $\mathbb{C}_{n+1}^{secant}$
Calculate Cauchy stress tensor σ
Calculate tangent constitutive equation tensor $\mathbb{C}_{n+1}^{tangent}$
IF $\delta_{fj,n+1} > 0.95$ **OR** $\delta_{mij,n+1} > 0.95$ **THEN**
    Delete current element
END IF

The UMAT subroutine provides a special variable, state dependent variable (SDV), for storing calculation results from the previous time increment iteration. In this particular UMAT subroutine, the failure exposure calculation result, current state of the damage and fracture plane orientation angle for each element from the previous iteration are stored using the SDVs. The values from the SDVs are used as inputs for the failure calculation on the current time increment iteration

Since the UMAT subroutine employs an implicit scheme for its iteration, both secant and tangent constitutive equations have to be derived and solved in order to compute the next iteration of the strain increment. The derivation of the secant and tangent constitutive equations is presented on the Appendix A.

A viscous regularisation is also implemented into the UMAT subroutine to improve the convergence of the finite element computations. The viscous regularisation is firstly introduced by Lapcyzk and Hurtado [67] by implementing an artificial viscosity on their CDM model based on work by Duvaut and Lion [71] on the regularisation model. For damage variable δ_j, the derivation of the regularisation model is shown in the following equation:

$$\dot{\delta_j} = \frac{1}{\eta}\left[\delta_j - \delta_j^v\right] \text{; with } j = ft, fc, m1t, m1c, m2t, m2c \tag{3.18}$$

The $\dot{\delta_j}$ is the rate of the viscous damage variable, while δ_j is the viscous damage variable and η is the viscosity parameter. Expanding the equation within the time increment $\Delta t = t_{n+1} - t_n$ during the iteration, the viscous damage variable at current time increment $\delta^v_{j,n+1}$ can be obtained using the damage variable at current time $\delta_{j,n+1}$ increment and previous time increment $\delta_{j,n}$ into the following equation:

$$\delta^v_{j,n+1} = \frac{\Delta t}{\eta + \Delta t}\delta_{j,n+1} + \frac{\eta}{\eta + \Delta t}\delta^v_{j,n} \text{; with } j = ft, fc, m1t, m1c, m2t, m2c \tag{3.19}$$

As suggested by Chen et al. [72], the viscosity coefficient $\eta = 0.0002$ is selected when calculating Equation (3.19).

Finally, the complete Fortran code of the UMAT subroutine of the current progressive damage model implementation of 3D Puck failure criterion combined with exponential damage degradation model is presented in Appendix B.

3.2 Validation

To validate the implementation of the progressive damage model in the UMAT subroutine, two finite element models are generated using Abaqus FEA software. The first model, a single-element model, is used to assess the response of the composite material with the exponential damage degradation model under the load and also to evaluate result of the 3D IFF Puck failure criterion result on the orientation angle of the fracture plane. The second model, an open-hole tensile test, is used to compare the material's

load-strain response and ultimate strength predictions from the FEA simulation with actual experimental results.

3.2.1 Single-Element Test

Finite Element Model Setup

In Abaqus FEA software, a cube with $1mm$ length on each side with a reduced integration (1 integration point) hexahedral element, a C3D8R element, is used to model the single-element test. A transverse compressive $0.1mm$ displacement load is applied to one face of the cube, perpendicular to the fibre direction, while a fixed support is applied to the opposite face of the cube where the force is applied, as illustrated in Figure 3.6. The material of the single-element test used in the current FEA simulation is T300/976 Carbon Fibre Reinforced Polymer (CFRP) composite. The details on the material properties of T300/976 CFRP composite is shown on Table 3.2.

Table 3.2: Material properties for T300/976 [66, 73]

E_{11}	E_{22}, E_{33}	G_{12}, G_{13}	ν_{12}, ν_{13}	ν_{23}
139.7 GPa	12.9 GPa	6.9 GPa	0.23	0.4
$R^t_{\parallel}$	$R^c_{\parallel}$	$R^t_{\perp}$	$R^c_{\perp}$	$R_{\perp\parallel}$
1516.8 MPa	1592.7 MPa	44.54 MPa	253 MPa	106.8 MPa
$\mathcal{G}^t_{\parallel}$	$\mathcal{G}^c_{\parallel}$	$\mathcal{G}^t_{\perp}$	$\mathcal{G}^c_{\perp}$	$\mathcal{G}^t_{\perp\parallel}$
91.6 N/mm	79.9 N/mm	0.22 N/mm	0.76 N/mm	0.46 N/mm

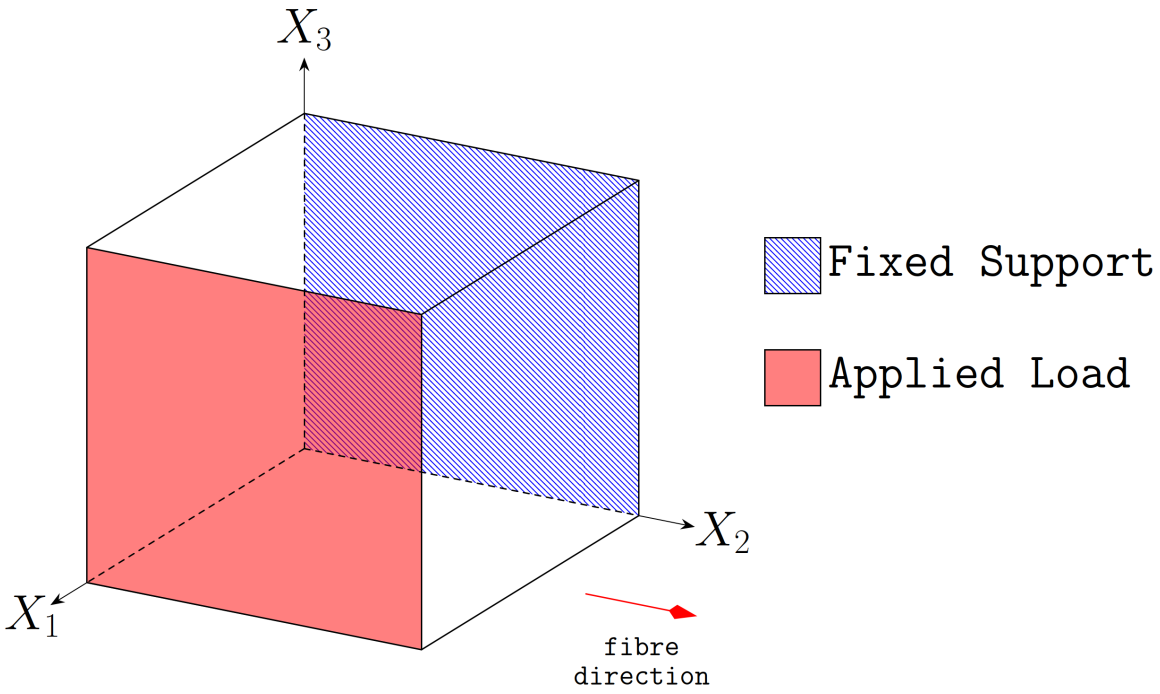


Figure 3.6: Single-element test model

Finite Element Model Simulation Result

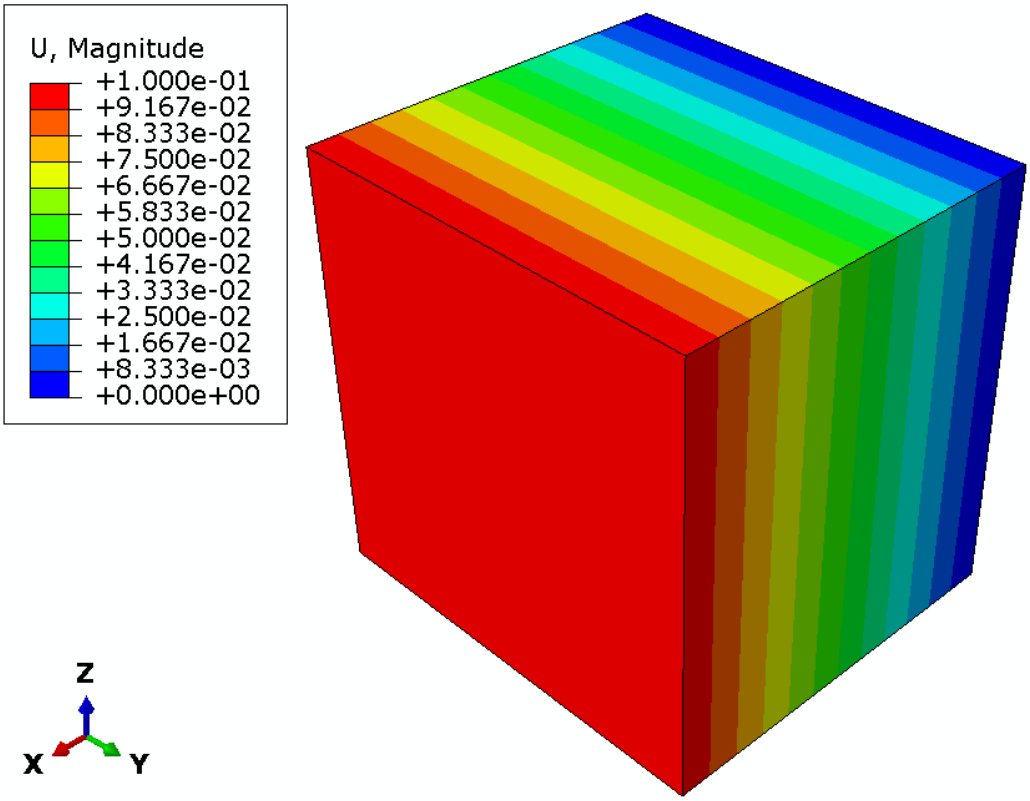


Figure 3.7: Displacement plot of single-element test

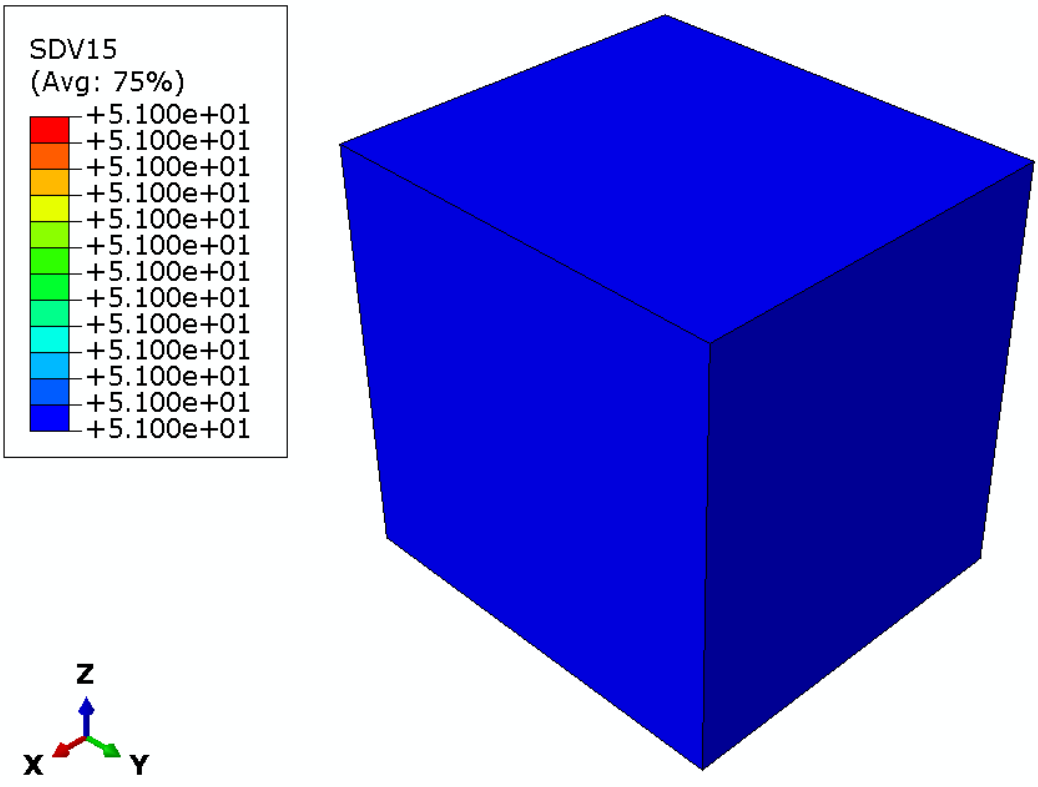


Figure 3.8: The orientation angle of the fracture plane plot on the single-element test

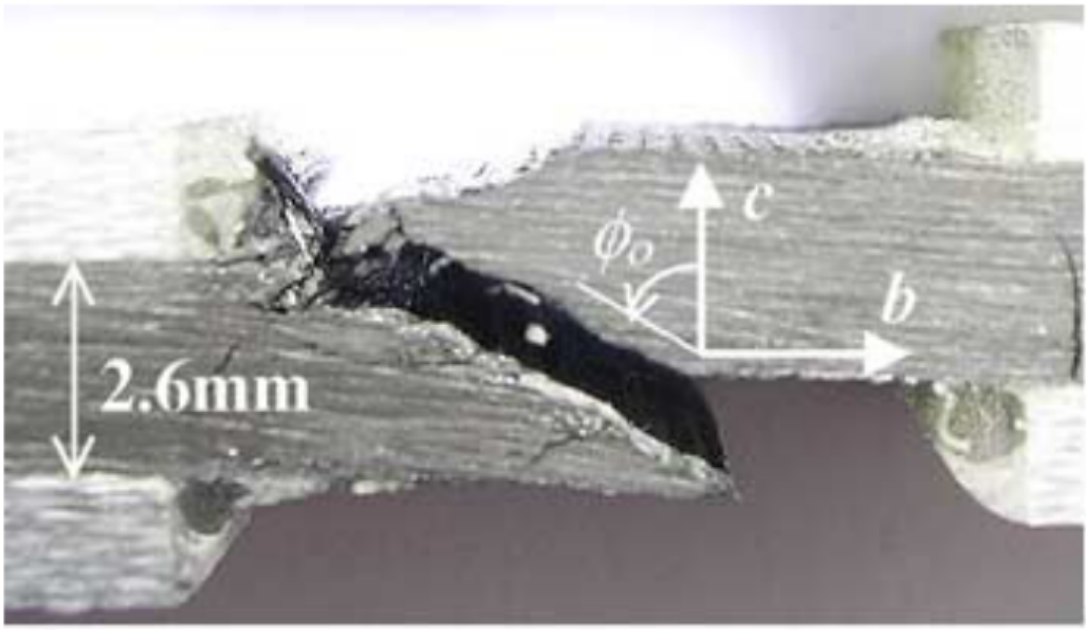


Figure 3.9: The fracture plane angle experimental result[1]

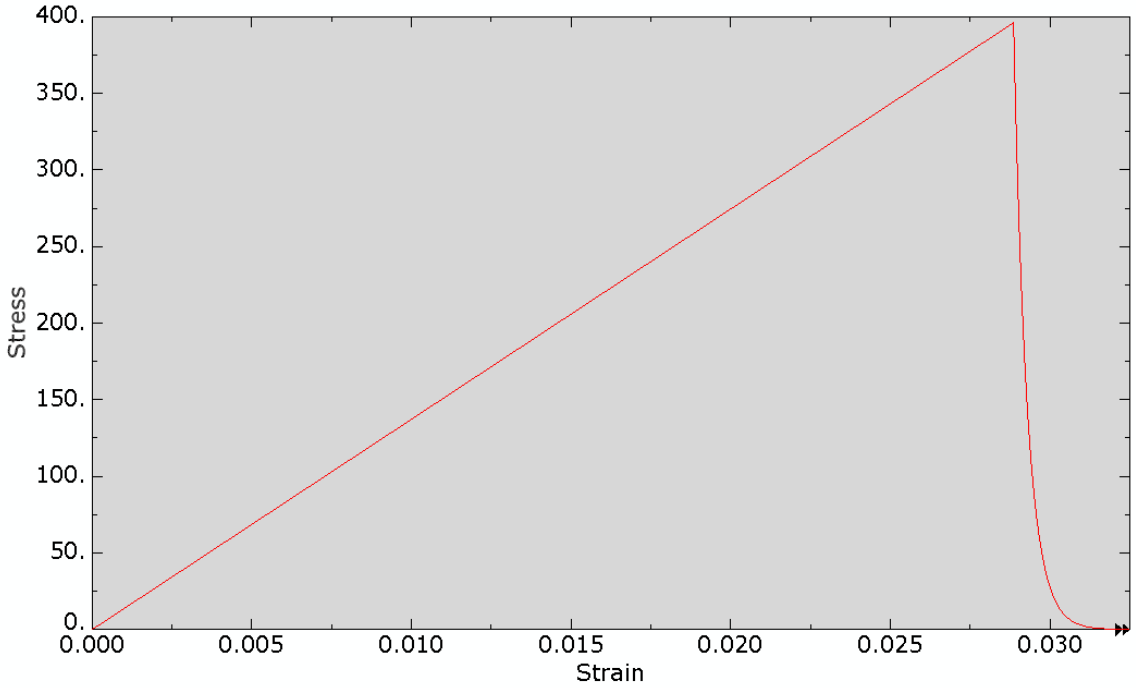


Figure 3.10: The stress-strain plot of the single-element test

Figure 3.7 presents the plot of the displacement, U, when $0.1mm$ transverse compressive displacement is applied into the cube element. As it can be seen from the plot, a $0.1mm$ displacement is observed on one of the element's side, while the opposite side exhibits no displacement due to the fixed support. The simulation result of the

[1]Reprinted from *Composites Part A: Applied Science and Manufacturing*, Vol. 37/1, S.T. Pinho, L. Iannucci, P. Robinson, "Physically-based failure models and criteria for laminated fibre-reinforced composites with emphasis on fibre kinking: Part I: Development," pp. 63–73, Copyright 2006, with permission from Elsevier

single-element finite element model on Figure 3.8 shows that the orientation angle of the fracture plane $SDV15$ is $\theta_{fp,SSM} = 51°$. This simulation result still within the range of the experimental result that has been conducted by Pinho et al. [74, 75] where the orientation angle of the fracture plane $\theta_{fp} = 53° \pm 3°$ as shown on Figure 3.9. Additionally, the stress-strain profile result on Figure 3.10 shows that the behaviour of the single-element finite element model after the element fails is similar with the exponential damage degradation profile as illustrated in the Figure 3.3.

3.2.2 Open-Hole Test

Finite Element Model Setup

Table 3.3: Material properties for AS4/PEEK [72, 76]

E_{11}	E_{22}, E_{33}	G_{12}, G_{13}	ν_{12}, ν_{13}	ν_{23}
127.6 GPa	10.3 GPa	6.0 GPa	0.32	0.49
$R^t_{\parallel}$	$R^c_{\parallel}$	$R^t_{\perp}$	$R^c_{\perp}$	$R_{\perp\parallel}$
2023.0 MPa	1234.0 MPa	92.7 MPa	176.0 MPa	82.6 MPa
	$\mathcal{G}^t_{\parallel}, \mathcal{G}^c_{\parallel}$	$\mathcal{G}^t_{\perp}, \mathcal{G}^c_{\perp}$	$\mathcal{G}^t_{\perp\parallel}$	
	128.0 N/mm	5.6 N/mm	4.93 N/mm	

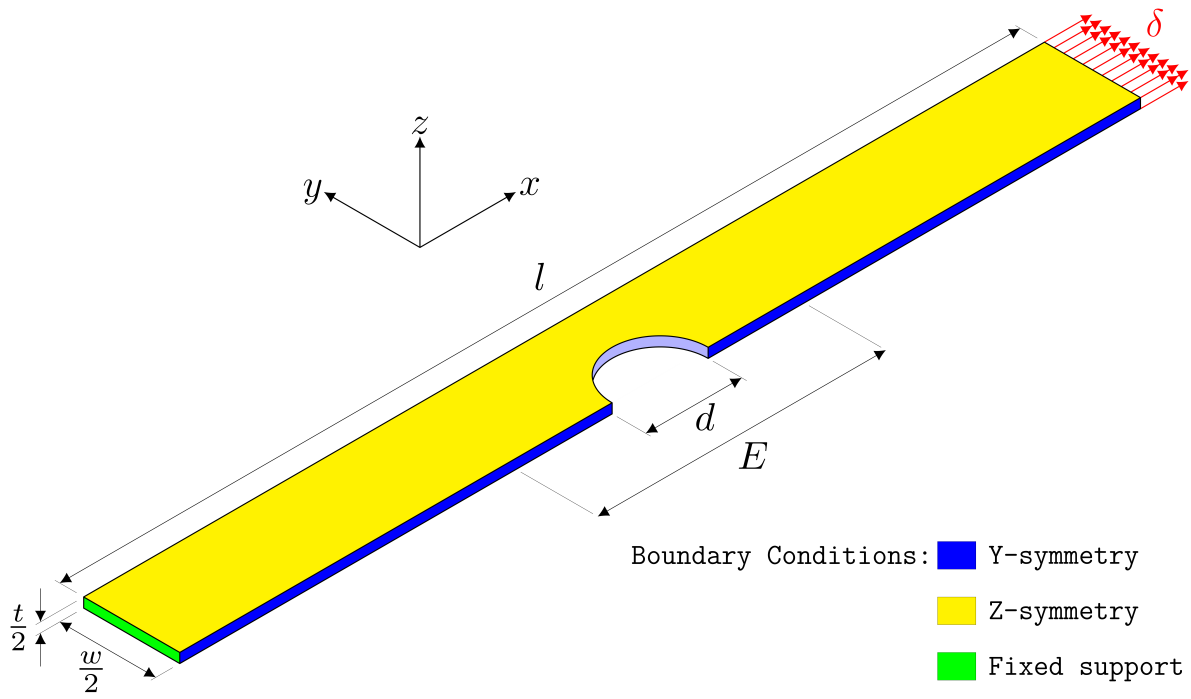


Figure 3.11: The dimension and the boundary conditions of the quarter OHT model

An open-hole test (OHT) using AS4/PEEK CFRP composite material (Table 3.3) with $[0/45/90/-45]_{2s}$ stacking sequence by Maa and Cheng [77] is used to validate the Puck failure criterion with fast FPOA search algorithm subroutine in the FEA simulation. The

OHT specimen dimension has length $l = 100mm$, width $w = 20mm$, thickness $t = 2mm$, hole diameter $d = 5mm$ and loading rate $\delta = 2mm/min$. During the OHT experiment, Maa and Cheng put a clip gauge at the centre of the specimen with $E = 30mm$ spacing to monitor the deformation of the specimen. For the FEA simulation, instead of using the full OHT model, a quarter OHT model with symmetric boundary condition applied at the xy-plane and xz-plane of the specimen is used to reduce the computation time [78], as shown in Figure 3.11.

Finite Element Model Simulation Result

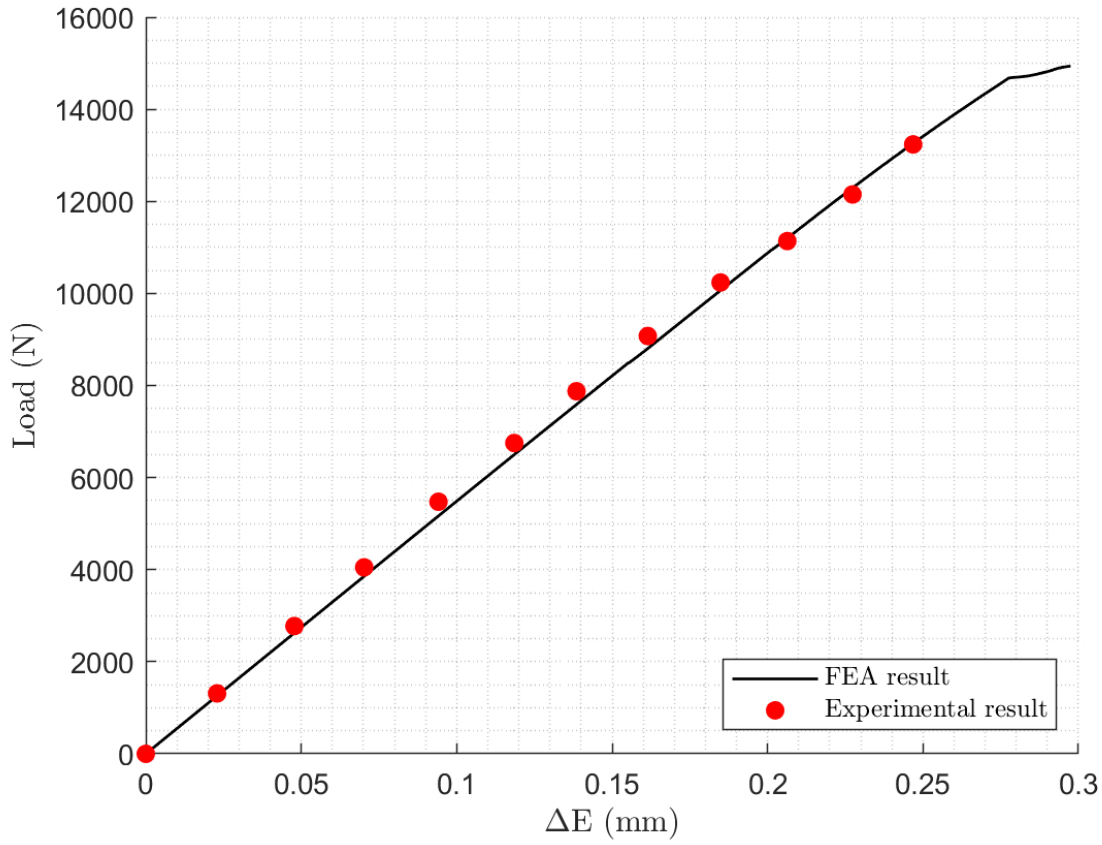
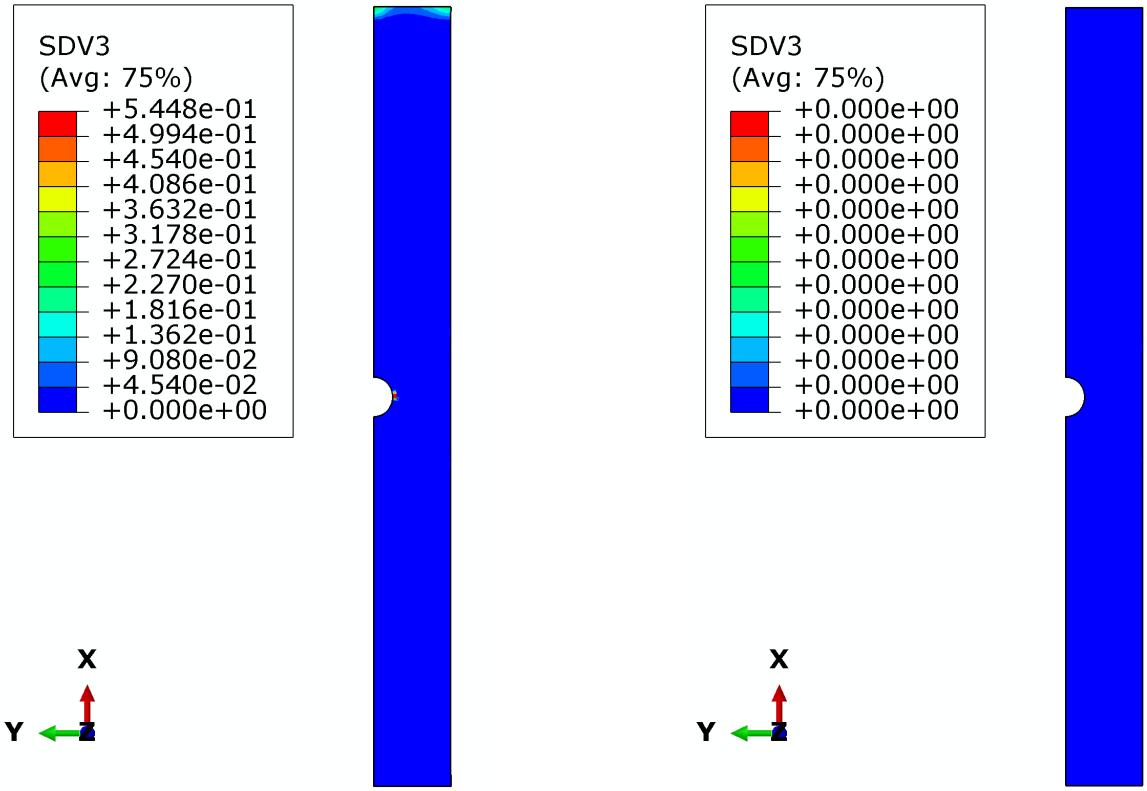


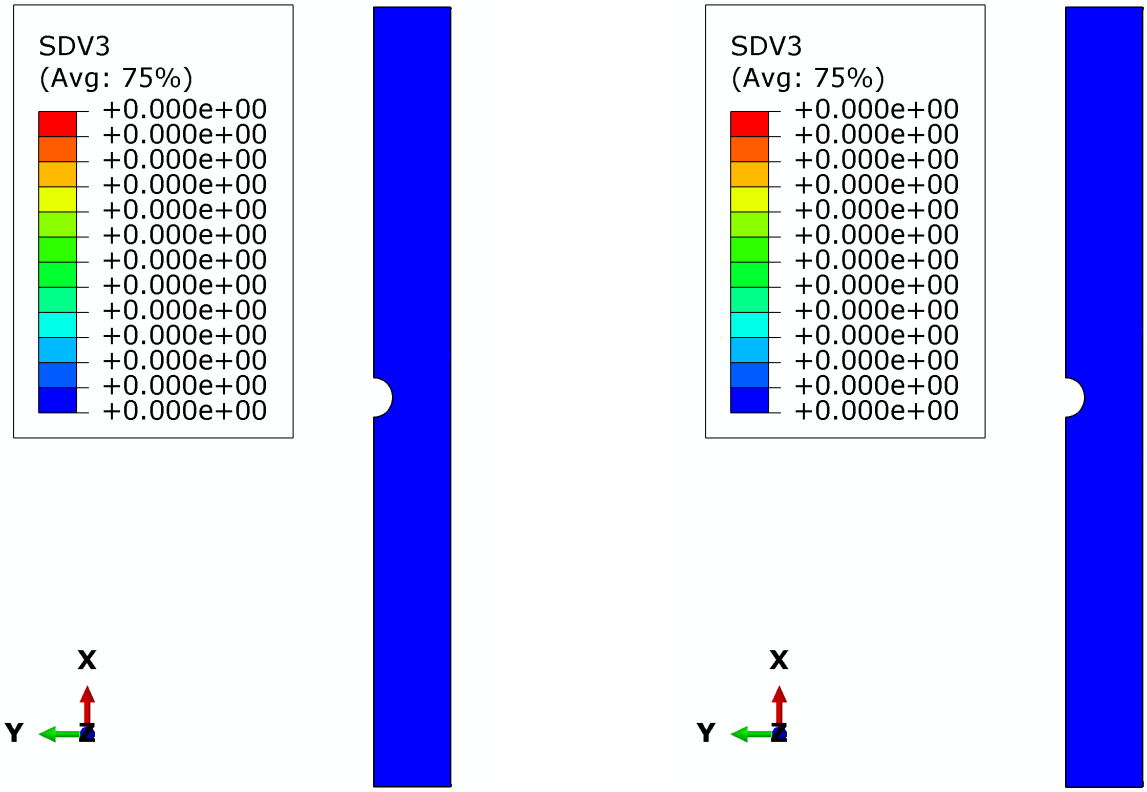
Figure 3.12: The comparison between the experimental and FEA simulation results for the OHT specimen

The FEA simulation result demonstrates a good agreement with the experimental result, as shown in Figure 3.12. In addition, the ultimate strength of the OHT obtained from the FEA simulation differs by only 2.42% from the experimental result conducted by Maa and Cheng [77]. The FEA simulation produced an ultimate strength of 14.94 kN, whereas the experimental result was 15.31 kN.



(a) $0°$ lamina

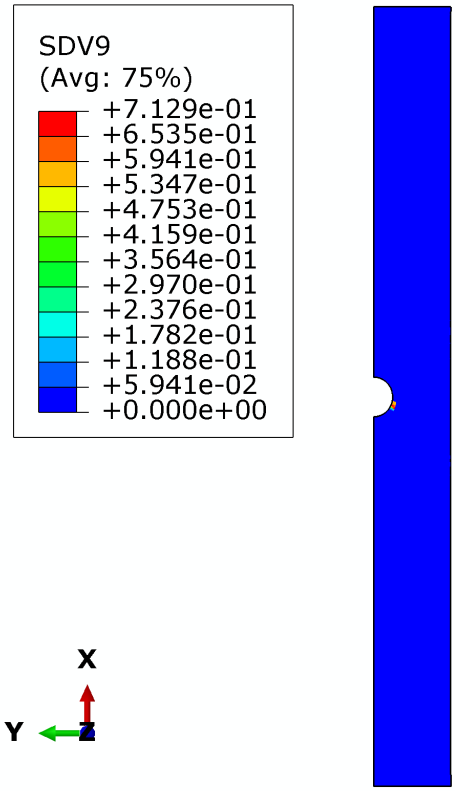
(b) $-45°$ lamina



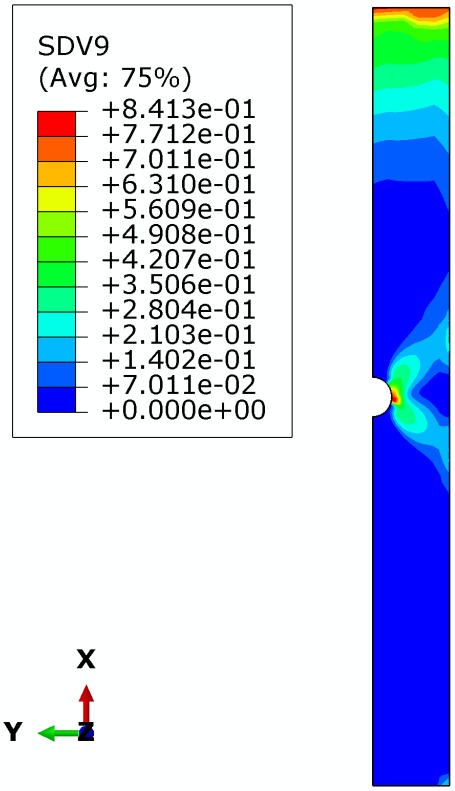
(c) $90°$ lamina

(d) $45°$ lamina

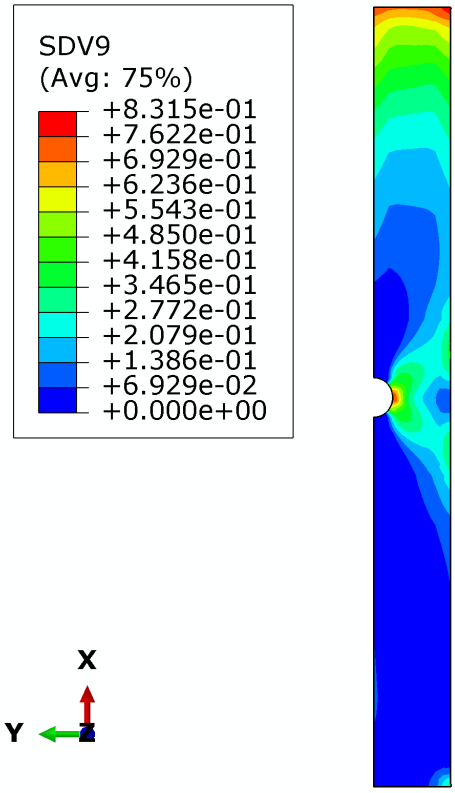
Figure 3.13: The plot of fibre damage in tension on the OHT finite element simulation on different lamina direction



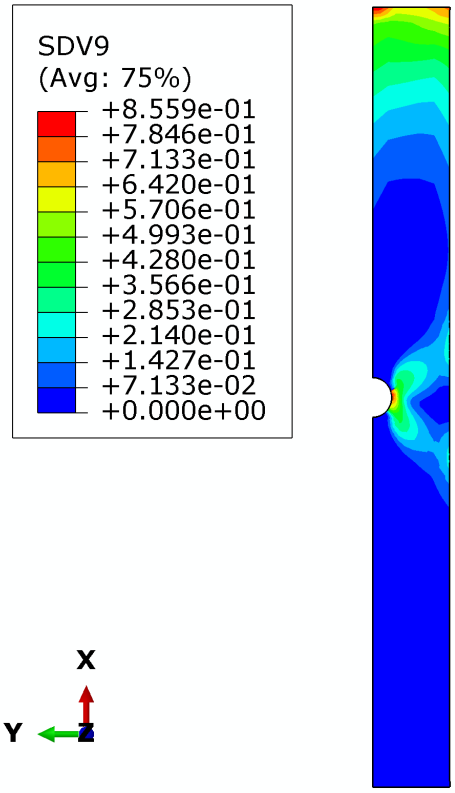
(a) $0°$ lamina



(b) $-45°$ lamina



(c) $90°$ lamina



(d) $45°$ lamina

Figure 3.14: The plot of matrix damage in tension on the OHT finite element simulation on different lamina direction

An example of the FEA simulation result on the OHT can be seen in Figures 3.13 and 3.14. These figures show the plot of fibre damage in tension ($SDV3$) and matrix damage in tension ($SDV9$), respectively at the same iteration time. The damage plot on the OHT shows that the matrix damage dominate the failure. The damage value in the plot was calculated using the Puck failure criterion and updated on each time increment on each Gauss/integration point.

3.3 Summary

In this chapter, a custom composite material model using 3D Puck failure criterion combined with the exponential damage degradation model is implemented using progressive damage model is proposed to simulate a more accurate behaviour of composite materials under load. Thus, an UMAT subroutine containing the custom composite material model is developed to address the limited built-in material model in the commercial FEA software. Two finite element models have been generated to validate the composite material model developed in the UMAT subroutine. The results indicate that the FEA simulation on the finite element model using 3D Puck failure criterion combined with the exponential damage degradation model could accurately represent the experimental results.

Chapter 4

Fast Fracture Plane Orientation Angle (FPOA) Search Algorithm

In the past few decades, the usage of composite material has increased significantly, particularly in the automotive and aerospace industries, due to its high specific strength characteristic. The composite material is also highly customisable. It could be tailored by using different combinations of fibre/matrix and different ply stacks or orientations to conform to the loading condition requirement. As a consequence, failure modes on composite material are becoming more complex and cannot accurately be represented by simple failure criterion evaluation such as Tresca or Von Misses failure criterion.

Initially, Tsai and Wu [40] developed a failure criterion using the failure strength of the material in each direction. In the Tsai-Wu failure criterion, the damage does not differentiate between fibre or matrix failure. Hashin [51] then improved the composite material failure evaluation by calculating fibre failure and matrix failure separately. Later, In the World-Wide Failure Exercise (WWFE) [52], a novel failure criterion developed by Puck [10, 11, 79], which is based on the Hashin failure criterion, was able to reproduce the failure under several stress conditions accurately compared to other failure criteria. Thus, the Puck failure criterion becomes one of the options for evaluating composite failure.

Puck employs fibre failure (FF) stress exposure ($f_{E,FF}$) and inter-fibre failure (IFF) stress exposure ($f_{E,IFF}$) to determine the damage that occurs along the fibre and within the matrix, respectively. In 3D case stress, a potential IFF fracture plane is required to determine the $f_{E,IFF}$. The potential IFF fracture plane is obtained using the Stepwise Search Method (SSM) algorithm by comparing the highest value of $f_{E,IFF}$ from all of the action planes parallel to the fibre. Puck and VDI [80] propose that the action planes are inclined at 1° intervals. As a result, 180 iterations of Equations (3.9)

Part of this chapter has been published in the European Journal of Computational Mechanics, https://doi.org/10.13052/ejcm2642-2085.3332

to (3.13) are required to acquire the IFF fracture plane for each stress case, which leads to high computational costs.

Due to its high computational cost, several fast FPOA search algorithms were developed to reduce the computational cost of the SSM algorithm. Weigand [81] proposes the Extended Golden Section Search (EGSS) algorithm, which utilises the Golden Section Search (GSS) algorithm [82] and successive parabolic interpolation technique [83] to obtain the IFF fracture plane angle orientation (FPOA). However, for several stress conditions, the EGSS algorithm incorrectly identify the location of $f_{E,IFF}$ global maximum value.

Schirmaier [84] improves the EGSS algorithm's accuracy into the Selective Range Golden Section Search (SRGSS) algorithm using the selective range (SR) method that divides the iteration range into several smaller blocks and then applies the GSS algorithm into blocks that contain local maximum value of $f_{E,IFF}$. Rezasefat [85] later proposes the Simple Parabolic Interpolation Search (SPIS) algorithm. Instead of using the GSS algorithm, Rezasefat employs successive parabolic interpolation technique into the SR method.

The fast FPOA search algorithm works by reducing the total amount of iteration required to obtain IFF FPOA. Since the data points from the iteration are less than the SSM algorithm, the accuracy of the fast FPOA search algorithm is also affected. In this paper, a robust fast FPOA search algorithm is proposed. Several new methods are implemented into the algorithm to alleviate the accuracy problem without increasing the computational time significantly. The result is then compared with the previous fast FPOA search algorithm and then applied as a subroutine in finite element analysis software.

4.1 FPOA Search Algorithm

In this section, the original fracture plane angle orientation (FPOA) search algorithm by Puck and the current fast FPOA search algorithms are explained. An IM7/8552 carbon fibre reinforced polymer (CFRP) material (Table 4.1) with varying load cases (Table 4.2) is used as the test case for the FPOA search algorithm. The result from each FPOA search algorithm is plotted into a fracture angle – $f_{E,IFF}$ value graph as comparison.

Three different load cases (Table 4.2) are selected to demonstrate the most common FPOA calculation result. Load case 1 is an example of one maximum value of $f_{E,IFF}$ iteration result. Load case 2 is an example of two local maximum values of $f_{E,IFF}$ results, with the global maximum value of $f_{E,IFF}$ nowhere near 90° or -90° angle orientation. Load case 3 is a specific case for two local maximum values of $f_{E,IFF}$ where the global maximum value of $f_{E,IFF}$ is located near 90° or -90° angle orientation. The load case 3

is used to demonstrate that other fast FPOAs have difficulties in accurately determining the global maximum value of $f_{E,IFF}$.

Table 4.1: Material properties for IM7/8552 [69, 70, 86, 87]

E_{11}	E_{22}, E_{33}	G_{12}, G_{13}	ν_{12}, ν_{13}	ν_{23}
171.42 GPa	9.08 GPa	5.39 GPa	0.32	0.52
$R^t_{\parallel}$	$R^c_{\parallel}$	$R^t_{\perp}$	$R^c_{\perp}$	$R_{\perp\parallel}$
2323.5 MPa	1200.1 MPa	62.3 MPa	199.8 MPa	92.3 MPa
$\mathcal{G}^t_{\parallel}$	$\mathcal{G}^c_{\parallel}$	$\mathcal{G}^t_{\perp}$	$\mathcal{G}^c_{\perp}$	$\mathcal{G}^t_{\perp\parallel}$
81.5 N/mm	106.3 N/mm	0.2774 N/mm	1.3092 N/mm	0.7879 N/mm

Table 4.2: Stress state examples to calculate failure exposure on 3D Puck IFF

Load Case	σ_{22} (MPa)	σ_{33} (MPa)	σ_{12} (MPa)	σ_{13} (MPa)	σ_{23} (MPa)	Notes
1	0	30	5	10	-40	Max value of $f_{E,IFF}$ at ($-56°$, 0.929)
2	4	-38	20	-33	46	Two local maximum values of $f_{E,IFF}$ at ($-7°$, 0.668) and ($72°$, 0.6746)
3	-66	-55	-4	-13	70	Two local max values of $f_{E,IFF}$ with the global max value of $f_{E,IFF}$ located near $90°/-90°$ at ($85°$, 0.745)

4.1.1 Existing FPOA Search Algorithm

Load case 1 – one $f_{E,IFF}$ maximum value

In order to find the maximum value of the failure exposure and the corresponding action plane for the 3D Puck IFF, Puck utilises Stepwise Search Method (SSM) algorithm. Puck SSM algorithm calculates all the failure exposure value that is obtained from $-90° \leq \theta < 90°$ iteration range angle. All the failure exposure values were then compared to obtain the maximum failure exposure value. The SSM algorithm requires 180 supporting points for a $1°$ inclination angle to predict the corresponding FPOA for the maximum failure exposure value for the given load, which requires a considerable amount of time to compute.

Figure 4.1 shows the value of $f_{E,IFF}$ and its corresponding inclination angle, totalling 180 supporting points. The inclination angle with the highest $f_{E,IFF}$ value will be used as the FPOA for the current stress state.

Weigand [81] proposes the EGSS (Extended Golden Section Search) algorithm to reduce the computation time when obtaining the maximum failure exposure value and its corresponding FPOA on 3D Puck IFF. The EGSS algorithm utilises the golden section search (GSS) algorithm [82] to reduce the number of supporting points required to find the maximum failure exposure value.

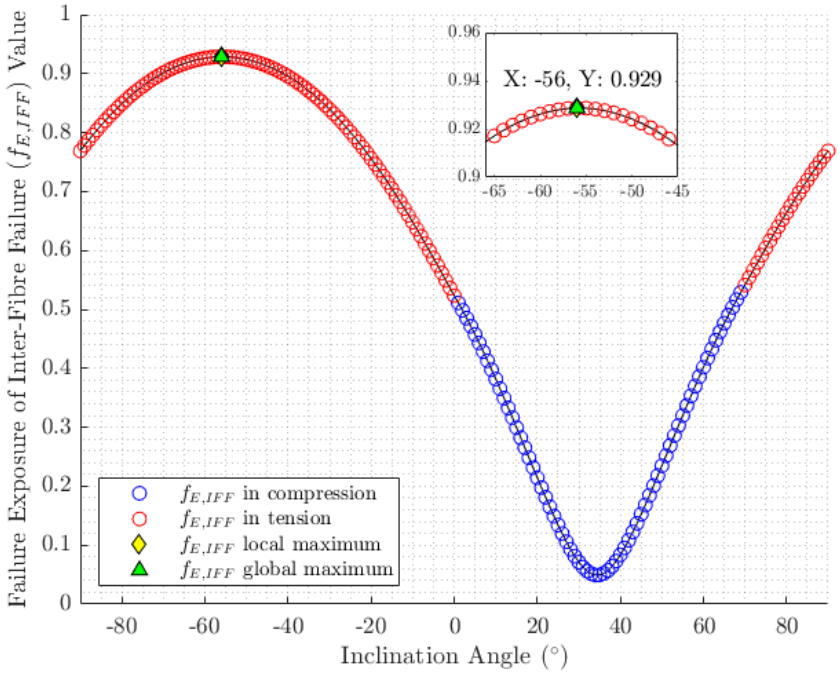


Figure 4.1: SSM algorithm applied on load case 1 yielded a maximum value of $f_{E,IFF}$ 0.929, with corresponding inclination angle $-56°$

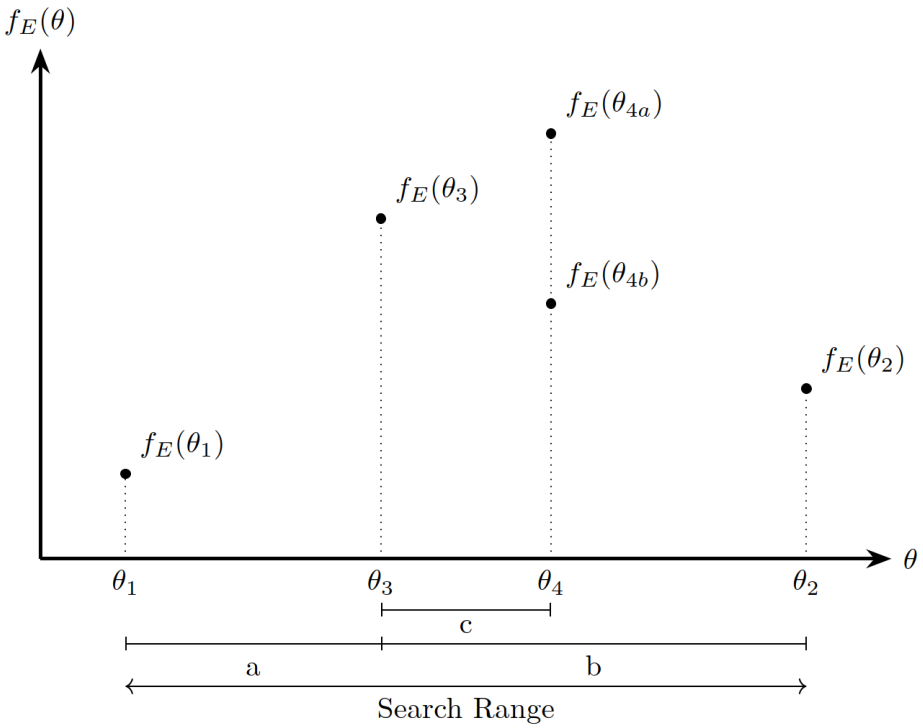


Figure 4.2: Illustration on golden search ratio algorithm

The GSS is an algorithm designed to identify an extremum point (either a maximum or minimum) within a specified interval. This algorithm works by consecutively nar-

rowing the search range interval using three supporting points. The ratio of the interval widths of these supporting points form a golden ratio ϕ (Eqns. (4.1) and (4.2)).

$$\frac{b}{a} = \phi \tag{4.1}$$

where,

$$\phi = \frac{1+\sqrt{5}}{2} \tag{4.2}$$

An additional fourth data point is calculated using Equation (4.3) to establish a symmetric search range interval.

$$\frac{b}{a} = \frac{a}{c} \tag{4.3}$$

In order to narrow the search range interval for the next iteration, the failure exposure calculation result from the third data point θ_3 has to be compared with the failure exposure calculation result from the fourth data point θ_4. If $f_E(\theta_4) \geq f_E(\theta_3)$ (the failure calculation result for the fourth data point θ_4 is $f_E(\theta_{4a})$ as illustrated in Figure 4.2) the first and second supporting points of the new search range interval will be θ_3 and θ_2. Otherwise (the failure calculation result for the fourth data point θ_4 is $f_E(\theta_{4b})$ as illustrated in Figure 4.2) the first and second supporting points of the subsequent search range interval will be θ_1 and θ_4. The GSS algorithm iteration will continue until the outer bounds of the search range interval $k = \theta_2 - \theta_1$ reaches the prescribed tolerance value $k = k_{exit}$.

By using the GSS algorithm, a more precise search range containing the maximum failure exposure value is generated. Next, a successive parabolic interpolation (Equation 4.4) [83] is applied to the search range obtained from the GSS algorithm to determine the maximum failure exposure value. The subscripts in Equation 4.4 refer to the inclination angle and the corresponding $f_{E,IFF}$ value results of the last three supporting points, which are iterated using the GSS algorithm.

$$\theta_{fp} = \theta_2 - \frac{1}{2}\frac{(\theta_2-\theta_1)^2(f_{E,IFF(2)}-f_{E,IFF(3)})(\theta_2-\theta_3)^2(f_{E,IFF(2)}-f_{E,IFF(1)})}{(\theta_2-\theta_1)(f_{E,IFF(2)}-f_{E,IFF(3)})(\theta_2-\theta_3)(f_{E,IFF(2)}-f_{E,IFF(1)})} \tag{4.4}$$

Figure 4.3 shows the same load case 1 calculation but using the EGSS algorithm. Compared to the SSM algorithm, the EGSS algorithm requires fewer supporting points calculation to obtain a similar result. As a consequence, the EGSS algorithm is faster than the SSM algorithm.

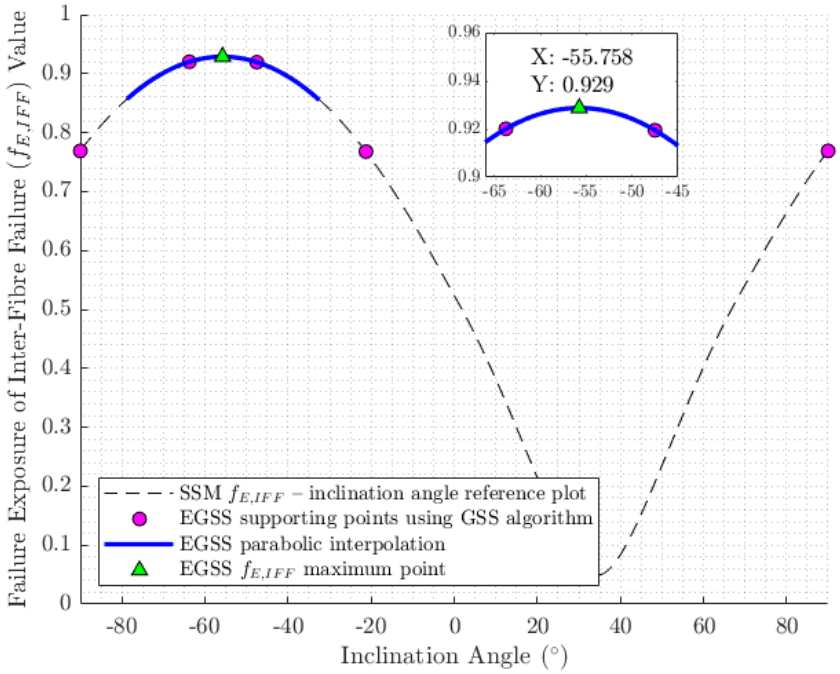


Figure 4.3: On load case 1, EGSS algorithm and SSM algorithm yielded similar result with maximum value of $f_{E,IFF}$ 0.929 and corresponding inclination angle $-55.758°$

Load case 2 – two local maximum values of $f_{E,IFF}$

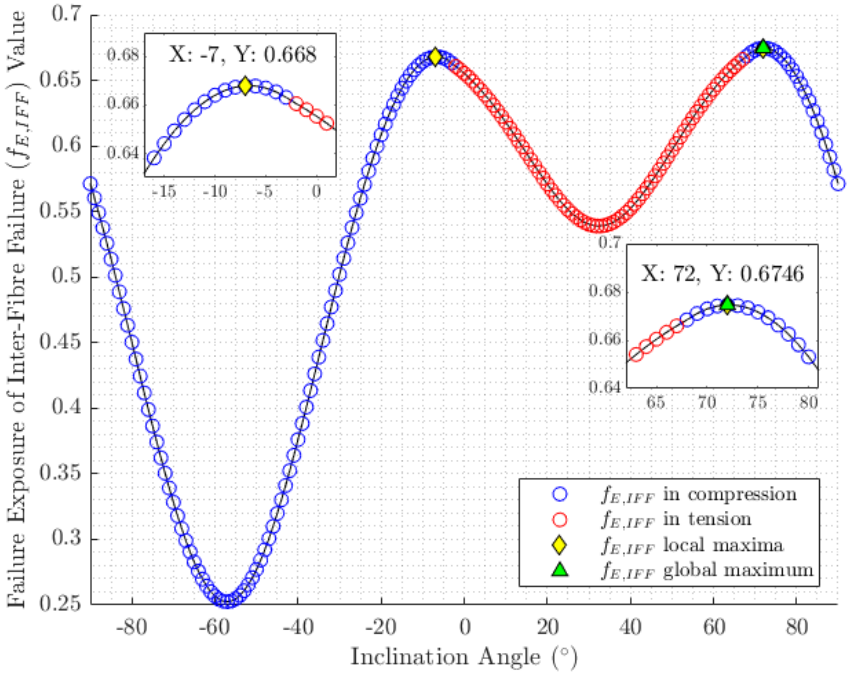


Figure 4.4: SSM algorithm applied on load case 2 yielded a maximum value of $f_{E,IFF}$ 0.6746, with corresponding inclination angle $72°$

However, the EGSS algorithm tends to incorrectly identify the global maximum values of $f_{E,IFF}$ when there are two or more local maximum values of $f_{E,IFF}$. As an example, on load case 2, the SSM algorithm identifies two local max $f_{E,IFF}$ coordinates at ($-7°$, 0.668) and ($72°$, 0.6746). The SSM algorithm then determines that the second coordinate point is the global maximum values of $f_{E,IFF}$ (Figure 4.4) since the second coordinate has a higher $f_{E,IFF}$ value compared to the first coordinate point.

As shown in Figure 4.5, the EGSS algorithm incorrectly determines the global maximum values of $f_{E,IFF}$ at ($-6,4625°$, 0.668) The result shows that the EGSS's global maximum value of $f_{E,IFF}$ is closer to the SSM algorithm's forst local maximum value of $f_{E,IFF}$ coordinate than the SSM algorithm's second local maximum value of $f_{E,IFF}$, which is the true global maximum value of $f_{E,IFF}$.

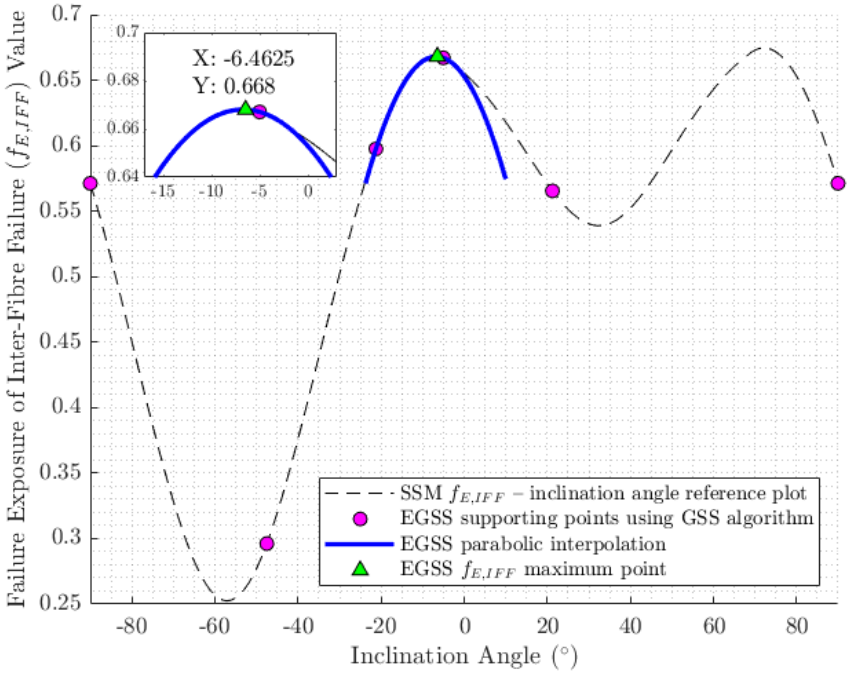


Figure 4.5: EGSS algorithm failed to identify the correct global maximum value of $f_{E,IFF}$ on two local maximum values of $f_{E,IFF}$ on load case 2. On load case 2, EGSS algorithm yielded a maximum value of $f_{E,IFF}$ 0.668, with corresponding inclination angle $-6.4625°$

Schirmaier [84] develops a novel search algorithm called the Selective Range Golden Section Search (SRGSS) algorithm to improve the EGSS algorithm by combining the Selective Range (SR) method and the GSS algorithm. According to Schirmaier, 20% of the 1×10^5 randomised stress cases evaluated using the EGSS algorithm have a $5°$ deviation on its FPOA. Schirmaier observes that the $f_{E,IFF}$ – inclination angle result plot has distinct characteristics, such as smooth, may have up to three local maxima and the minimum distance between two local maxima is always larger than $25°$.

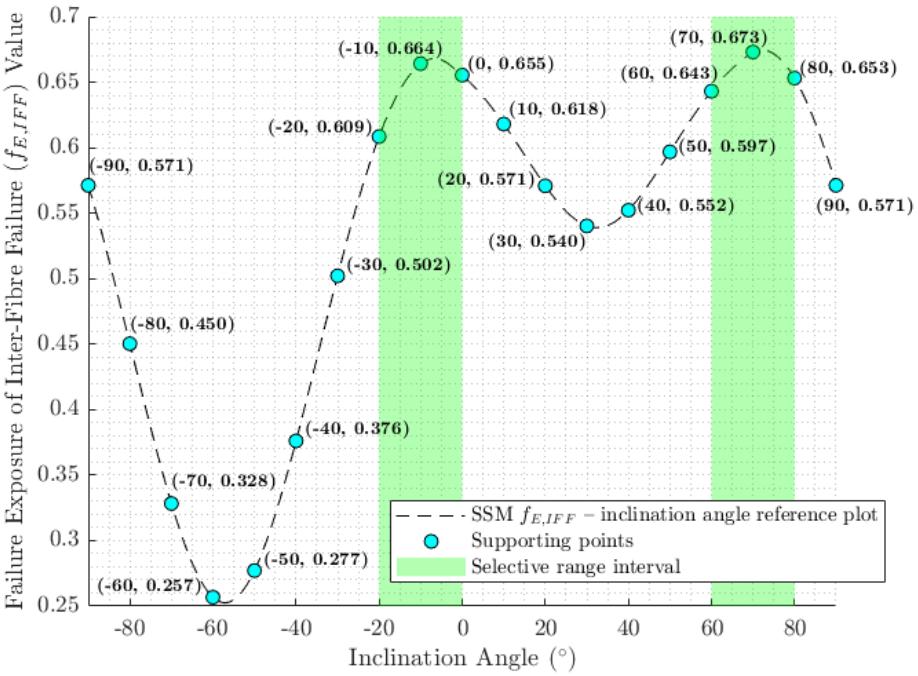


Figure 4.6: A localised search range interval (in green) was established using the Selective Range (SR) method. This method is obtained by comparing the neighbouring supporting points and its corresponding $f_{E,IFF}$ values

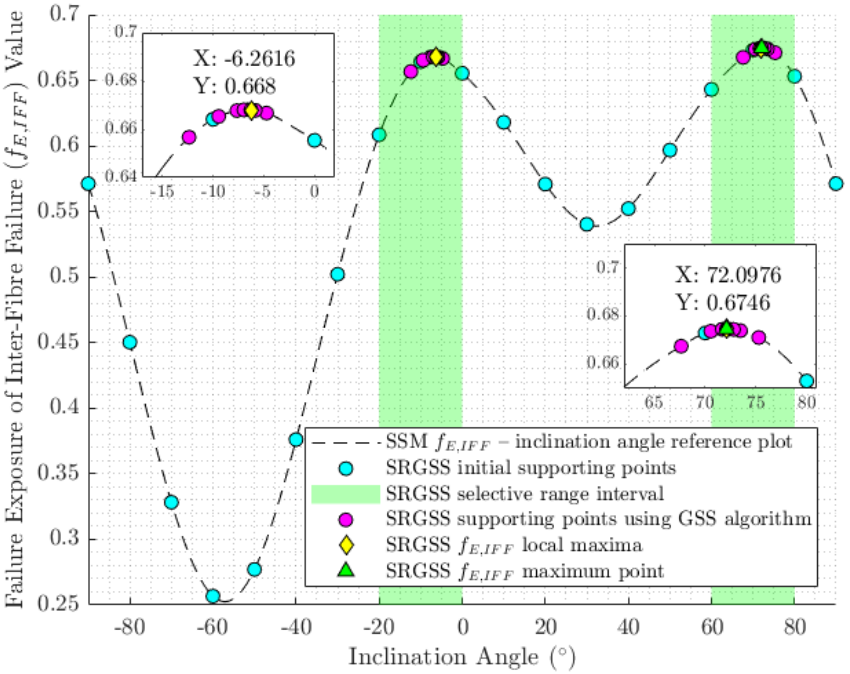


Figure 4.7: SRGSS algorithm identify the correct global maximum value of $f_{E,IFF}$ on two local maximum values of $f_{E,IFF}$ on load case 2. On load case 2, SRGSS algorithm yielded a maximum value of $f_{E,IFF}$ 0.6746, with corresponding inclination angle $72.0976°$

Schirmaier proposes the SR method, which reduces the wide $180°$ search range interval into one or more narrow $20°$ search range intervals to find the possibility of multiple local maxima on the $f_{E,IFF}$ – inclination angle result plot. The $20°$ search range interval is determined by calculating the supporting points on every $10°$ interval inclination angle. If the current supporting point has higher $f_{E,IFF}$ values than the previous and the next supporting point, then the span of inclination angle between the previous and the next supporting point is designated as the SR interval (Figure 4.6). After the SR interval is obtained, the GSS algorithm is applied to obtain the maximum $f_{E,IFF}$ value in each SR interval. If there are two or more SR intervals, the local maximum value of $f_{E,IFF}$ on each SR interval is compared to obtain the global maximum value of $f_{E,IFF}$ (Figure 4.7). This method greatly improves the accuracy of the GSS algorithm while maintaining less computational resources since it uses fewer supporting points to obtain the global maximum value of $f_{E,IFF}$ compared to the SSM algorithm.

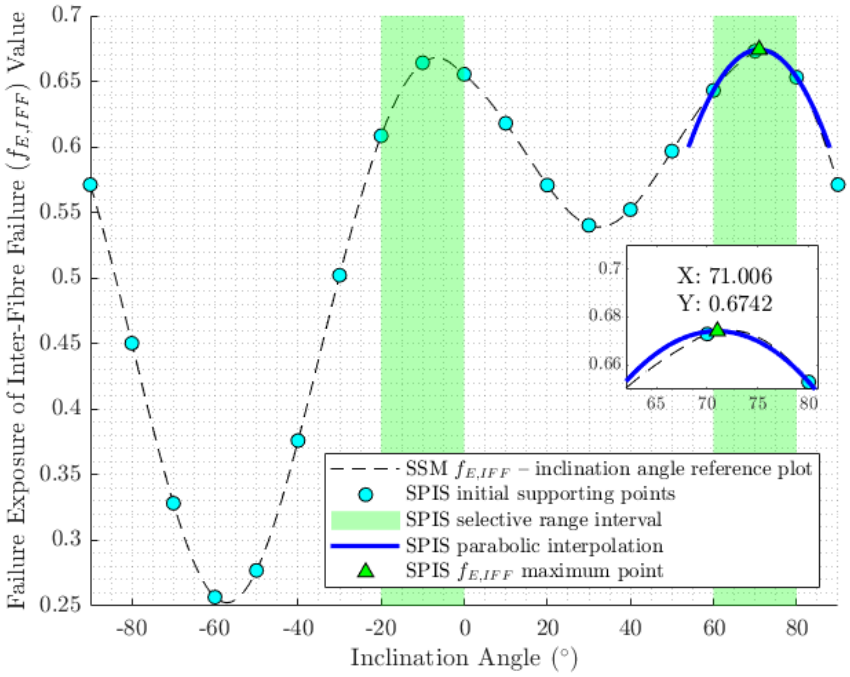


Figure 4.8: SPIS algorithm identify the correct global maximum values of $f_{E,IFF}$ on two local maximum values of $f_{E,IFF}$ on load case 2. On load case 2, SPIS algorithm yielded a maximum value of $f_{E,IFF}$ 0.6742 and corresponding inclination angle $71.006°$

Rezasefat [85] later improves the SRGSS algorithm by implementing the Simple Parabolic Interpolation Search (SPIS) algorithm (Figure 4.8). SPIS aims for less computation time compared to the SRGSS. SPIS uses parabolic interpolation (Equation 4.4) instead of the GSS algorithm to reduce the required supporting points to obtain the global maximum value of $f_{E,IFF}$. The parabolic interpolation is only implemented into the SR interval range that has the highest average of two smaller $f_{E,IFF}$ value neighbouring

supporting points, further reducing the total amount of the total calculation time. In the final step, SPIS compares the value obtained from the parabolic interpolation with the $f_{E,IFF}$ value at the inclination angle $-90°$ and $90°$.

Load case 3 – two local maximum values of $f_{E,IFF}$ with the global $f_{E,IFF}$ maximum value located near $-90°/90°$

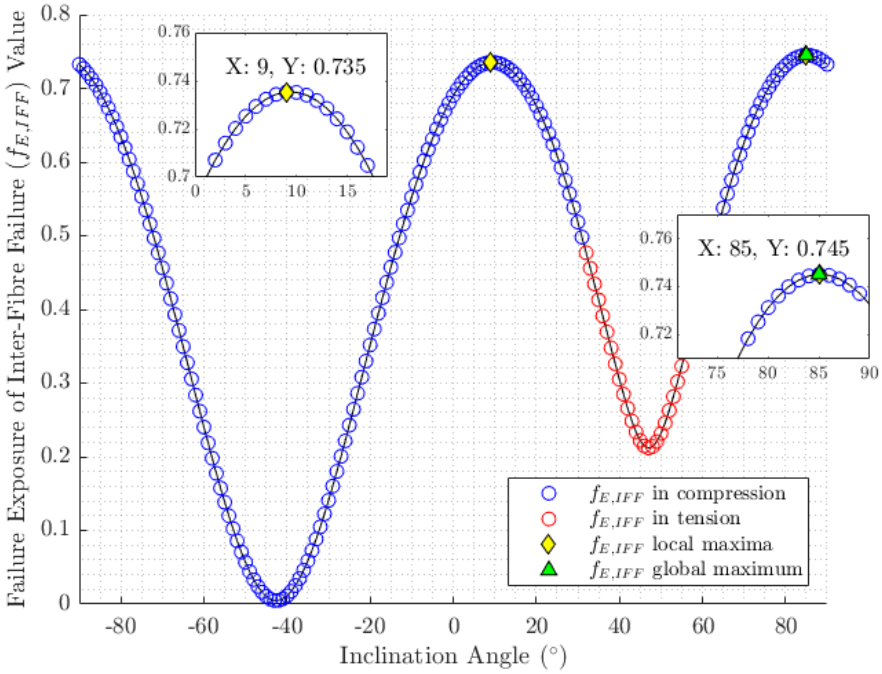
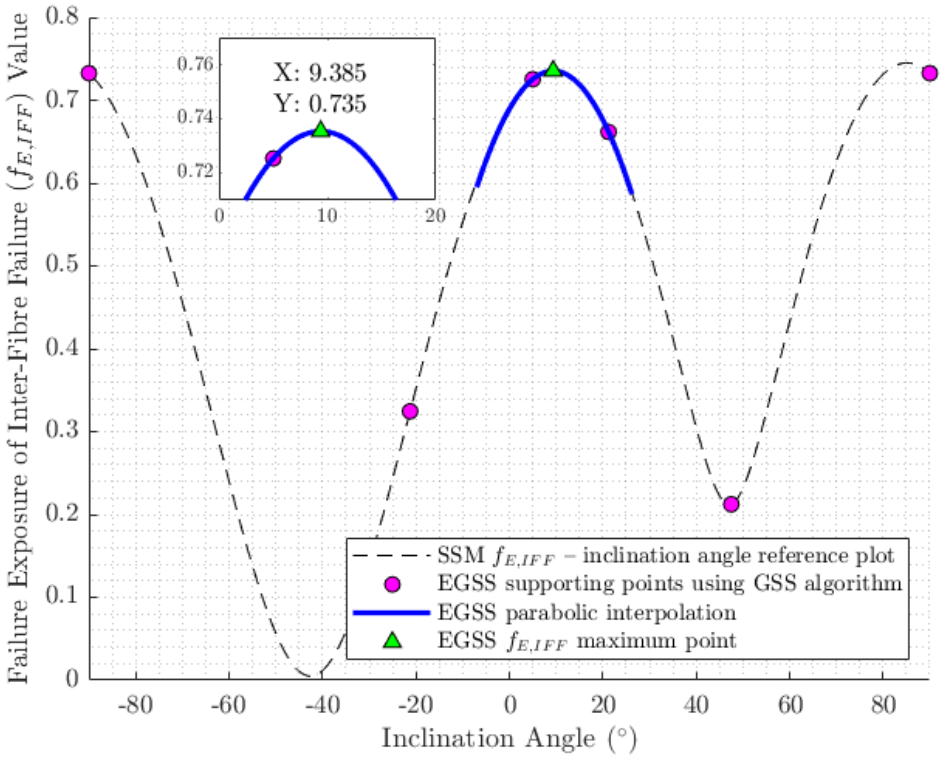


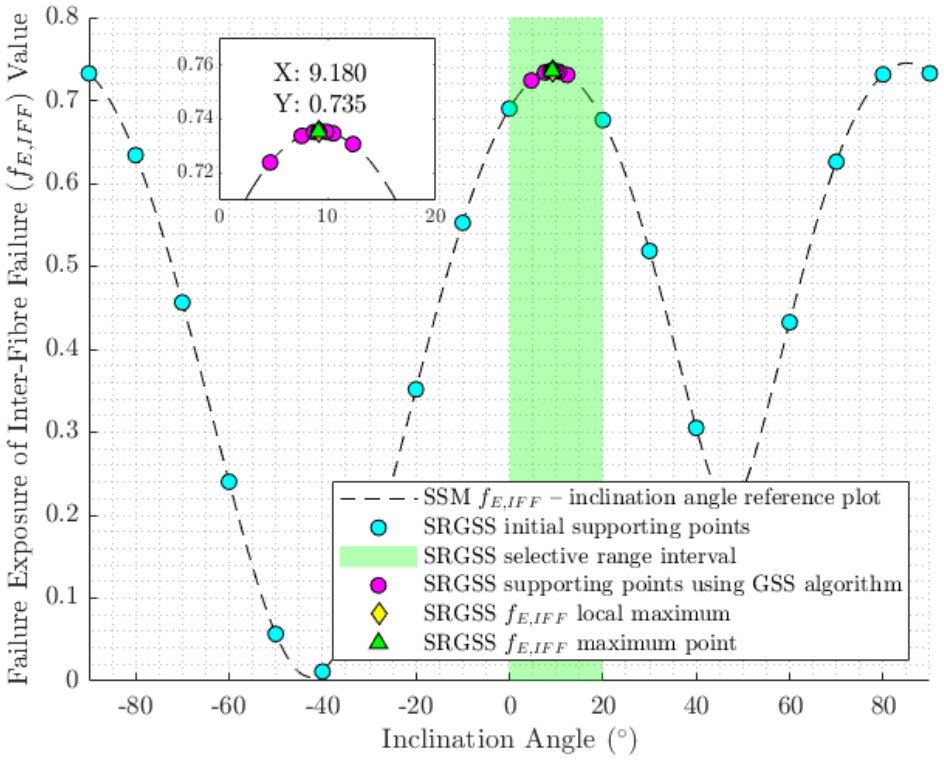
Figure 4.9: SSM algorithm applied on load case 3 yielded a maximum value of $f_{E,IFF}$ 0.745, with corresponding inclination angle $85°$

On load case 3, two local maximum values of $f_{E,IFF}$ coordinates at ($9°$, 0.735) and ($85°$, 0.745) are identified using the SSM algorithm. The SSM algorithm determines the second local maximum values of $f_{E,IFF}$ as the global maximum value of $f_{E,IFF}$ since it has higher $f_{E,IFF}$ value compared to the other point. The location of the global maximum value of $f_{E,IFF}$ is near the edge of the Puck search range interval ($-90° \leq \theta \leq 90°$). The result of the fracture angle – $f_{E,IFF}$ value plot using the SSM algorithm is shown in Figure 4.9.

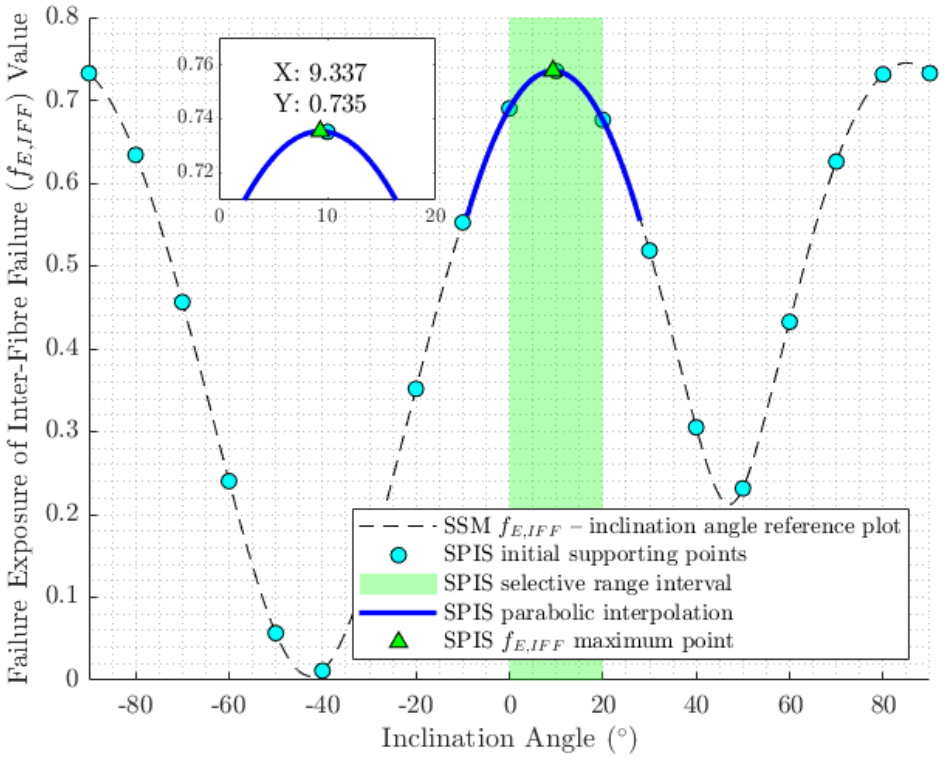
Even though load case 3 is similar to load case 2 (two local maximum values of $f_{E,IFF}$), the current fast FPOA search algorithm (EGSS, SRGSS and SPIS) incorrectly determines the global maximum value of $f_{E,IFF}$, as shown in Figure 4.10.



(a) EGSS



(b) SRGSS



(c) SPIS

Figure 4.10: Current fast algorithm, i.e., (**a**) EGSS, (**b**) SRGSS and (**c**) SPIS) failed to identify the correct global maximum value of $f_{E,IFF}$

4.1.2 Proposed Fast FPOA Search Algorithm

A novel fast FPOA search algorithm is proposed to improve the reliability of the current fast FPOA search algorithm in finding the global maximum value of $f_{E,IFF}$ and its corresponding fracture angle. The proposed fast FPOA search algorithm is named the Improved Selective Range Brent Method (ISRBM).

In the ISRBM search algorithm, an improved version of the SR method is implemented to rectify the unintentional bug in the SRGSS or SPIS algorithm. The SR method itself works by comparing the current supporting point with two points on both preceding and following supporting points. If the current supporting point has higher $f_{E,IFF}$ value, the preceding and the following supporting points are regarded as the starting and the stopping points for the SR interval.

On the SRGSS and SPIS algorithm, the calculation of the supporting points for the SR method is limited within $-90° \leq \theta \leq 90°$ inclination angle [84, 85], as shown in Figure 4.11. Since the SR interval evaluation requires both preceding and following points, the SR interval evaluation will never initialise at the edge points ($-90°/90°$ inclination angle) since there is only one point to compare (either preceding or following point, not both).

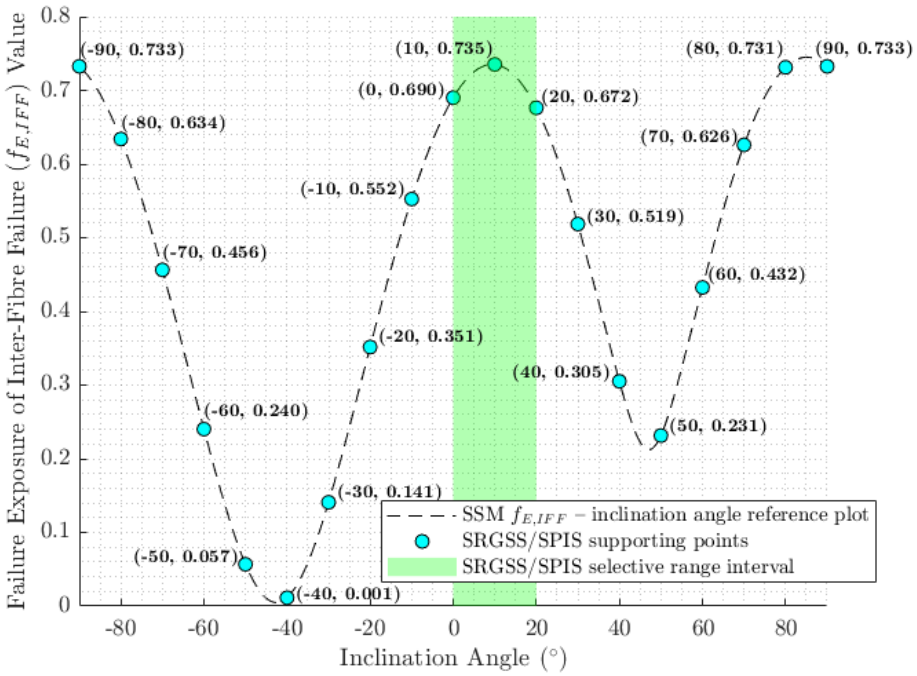


Figure 4.11: Current search range interval on SRGSS or SPIS only covers inclination angle interval $-90° \leq \theta \leq 90°$

The ISRBM search algorithm improves the SR method by appending an additional supporting point to ensure that the SR interval evaluation can be initialised even at the edge point ($-90°/90°$ inclination angle). The additional supporting point is calculated at a $100°$ inclination angle. Since $f_{E,IFF}$ is periodic for every $180°$ (the $f_{E,IFF}$ value at θ and $\theta + 180°$ will be identical), the $f_{E,IFF}$ calculation result that is obtained at $-90°$ and $-90°$ inclination angle can be reused as the $f_{E,IFF}$ result at $90°$ and $100°$ inclination angle respectively, thus further reducing unnecessary computation time. As a result, the SR method is implemented into a $-90° \leq \theta \leq 100°$ inclination angle on the ISRBM search algorithm (Figure 4.12) instead of the usual $-90° \leq \theta \leq 90°$ inclination angle.

The ISRBM search algorithm obtains the maximum values of $f_{E,IFF}$ on each SR interval using the Brent Method (Figure 4.13). Brent method [88] combines the more reliable GSS algorithm and the fast successive parabolic interpolation technique. Successive parabolic interpolation technique has a superlinear convergence rate but is prone to fail when the evaluated points are collinear. On the contrary, the GSS algorithm has a linear convergence rate but is more robust. In the Brent method, the default calculation uses successive parabolic interpolation, but when the evaluated points are collinear, the calculation will fall back to the GSS algorithm.

In the final step, when the corresponding fracture angle of the global maximum value of $f_{E,IFF}$ is between $90° \leq \theta \leq 100°$ inclination angle, the fracture angle needs to be substracted by $180°$ due to $f_{E,IFF}$ periodicity characteristic. The pseudocode of the ISRBM search algorithm is summarised in algorithm 2.

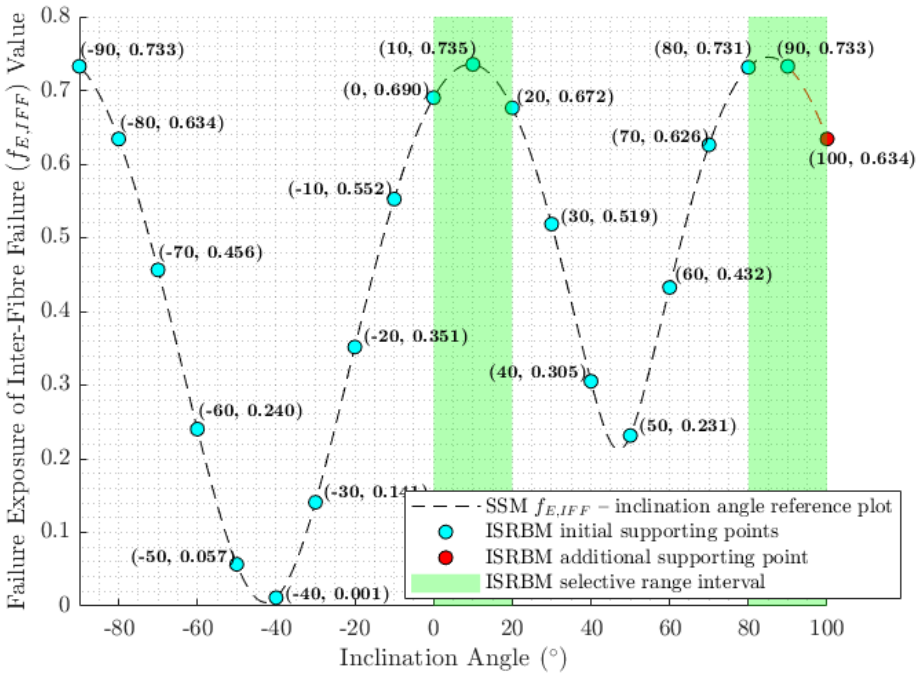


Figure 4.12: Extended search range interval on ISRBM covers inclination angle interval $-90° \leq \theta \leq 100°$. Note the additional calculation at $100°$ inclination angle

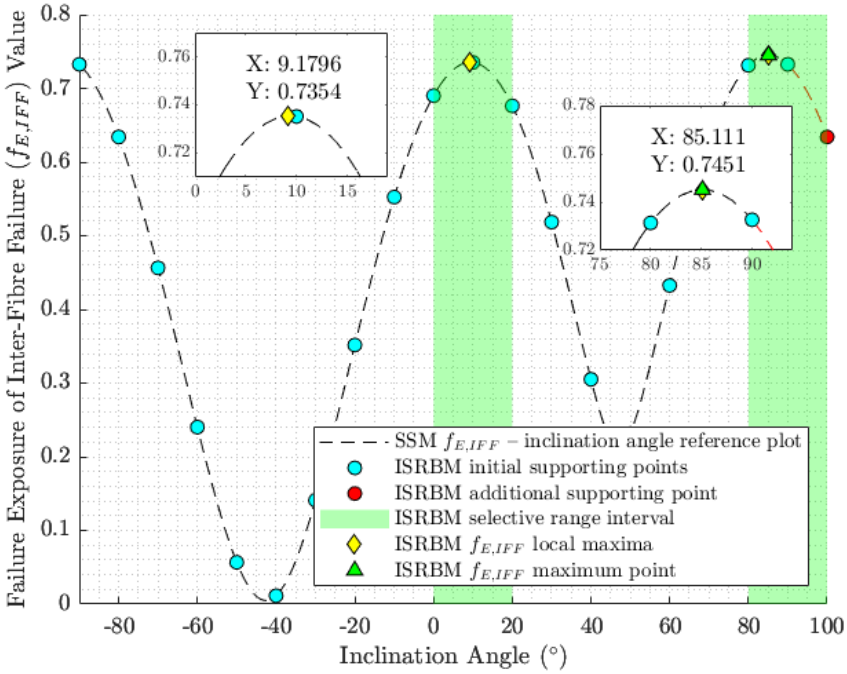


Figure 4.13: ISRBM algorithm identify the correct global maximum values of $f_{E,IFF}$ on two local maximum values of $f_{E,IFF}$ on load case 3. On load case 3, ISRBM algorithm yielded a maximum value of $f_{E,IFF}$ 0.7541 and corresponding inclination angle $85.111°$

Algorithm 2 Pseudocode of the ISRBM search algorithm

1: Calculate the $f_{E,IFF}$ at supporting points from $-90^\circ \leq \theta \leq 100^\circ$ inclination angle with 10° interval. Reuse the $f_{E,IFF}$ calculation result of the supporting points to reduce the computation time since the $f_{E,IFF}$ result is periodic for every 180° inclination angle
2: **IF** inclination angle between $-90^\circ \leq \theta \leq 90^\circ$ **THEN**
3: Calculate failure exposure at corresponding inclination angle $f_{E,IFF}(\theta)$
4: **ELSE IF** inclination angle is equal or more than -90° **THEN**
5: Assign $f_{E,IFF}(\theta - 180^\circ)$ result into $f_{E,IFF}(\theta)$
6: **END IF**
7: Find the appropriate selective range interval that contains local maximum point
8: Applying Brent Method into the selective range interval
9: Compare the local maximum values of $f_{E,IFF}$ obtained from the previous step
10: Assign the highest local maximum values of $f_{E,IFF}$ into global maximum value of $f_{E,IFF}$
11: Assign the corresponding inclination angle of global maximum value of $f_{E,IFF}$ as the FPOA, θ_{fp}
12: Check the θ_{fp} result
13: **IF** θ_{fp} is more than 90° **THEN**
14: Substract the θ_{fp} by 180°
15: **END IF**

4.2 Performance and Result Evaluation

In order to evaluate the performance of the FPOA search algorithm, 1×10^5 randomised stress cases are tested. IM7/8552 CFRP material (Table 4.1) is used as the material input. Each FPOA search algorithm is evaluated in terms of the calculation time and reliability to obtain the correct calculation result. Table 4.3 shows the average calculation time to obtain the global maximum value of $f_{E,IFF}$ and its corresponding FPOA θ_{fp} for each stress case. In this table, the performance increase of each fast FPOA is also compared with the standard Puck SSM algorithm.

Table 4.3: Performance comparison of each fast FPOA search algorithm to the Puck SSM with a 1° increment search

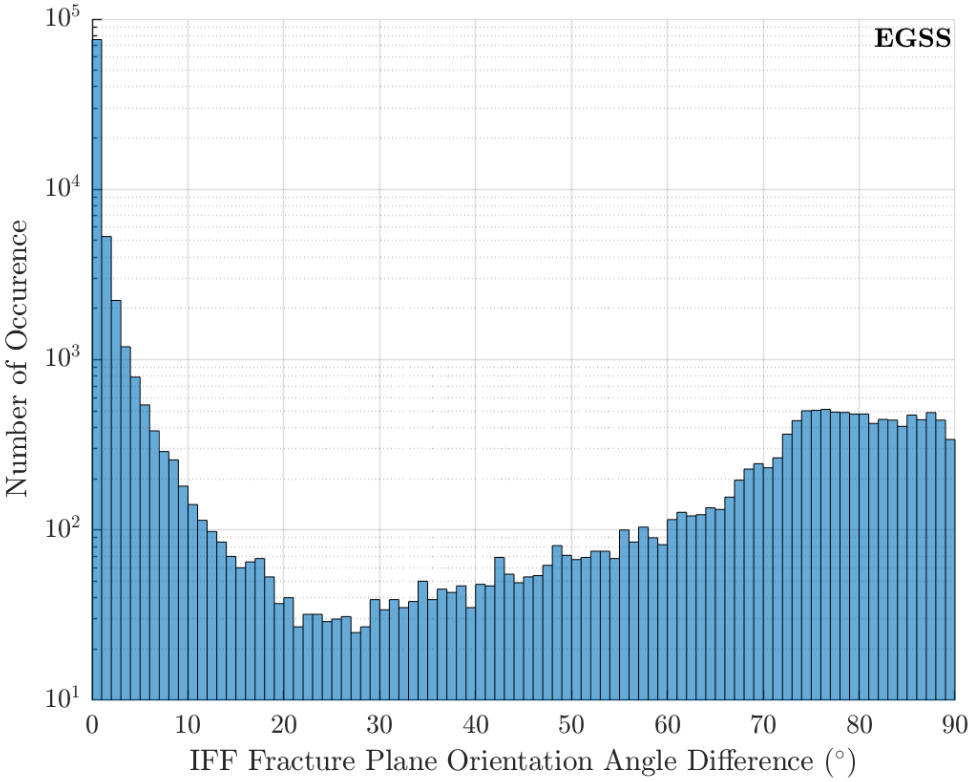
Search Algorithm	Average Calculation Time (millisecond)	Average Calculation Time Decrease Compared to SSM (%)
SSM	0.59	-
EGSS	0.029	1934.48
SRGSS	0.079	646.84
SPIS	0.071	730.99
ISRBM	0.099	495.96

Next, the reliability of each fast FPOA search algorithm is also evaluated by comparing the global maximum values of $f_{E,IFF}$ and its corresponding FPOA θ_{fp} using the Puck SSM algorithm. However, instead of 1°, a more precise 0.01° increment search is

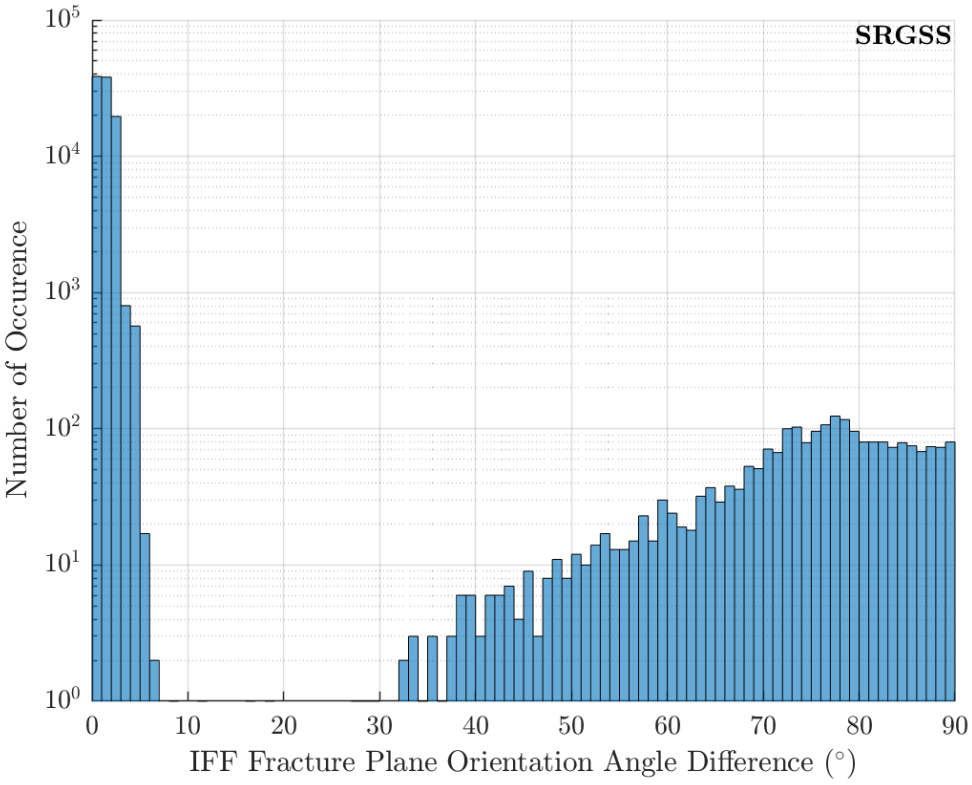
used to iterate the interval angle. The average difference result obtained from each fast FPOA search algorithm with the Puck SSM algorithm is shown in Table 4.4. In addition, the distribution of the FPOA result difference on each fast FPOA search algorithm for all 1×10^5 randomised stress cases is visualised in Figure 4.1. The FPOA result difference is obtained by taking the absolute value of the difference between the fast FPOA search algorithm result and the FPOA result from SSM algorithm with $0.01°$ increment. Since $f_{E,IFF}$ value is periodic every $180°$, the maximum FPOA difference between the fast FPOA search algorithm and the Puck SSM algorithm is always less or equal to $90°$.

Table 4.4: Result comparison of each fast FPOA search algorithm to the Puck SSM with a $0.01°$ increment search

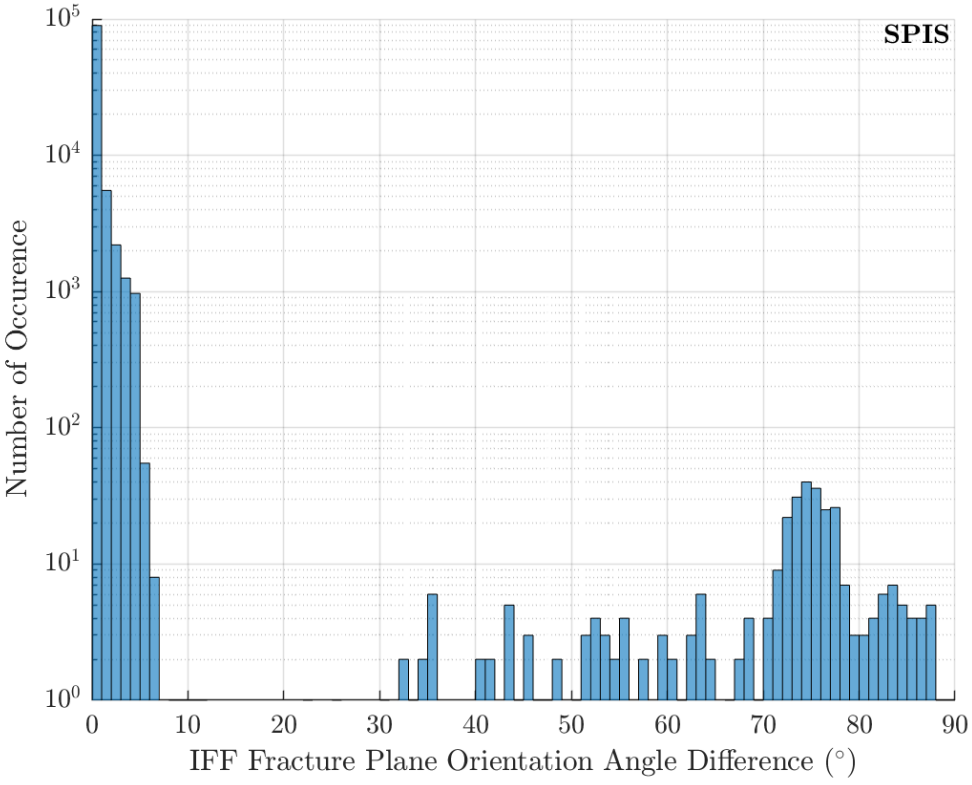
Search Algorithm	Average Failure Exposure Value Difference	Average Fracture Plane Angle Difference (°)
EGSS	0.0156	9.485
SRGSS	0.0027	2.991
SPIS	1.3322e-04	0.5801
ISRBM	1.0875e-07	0.0048



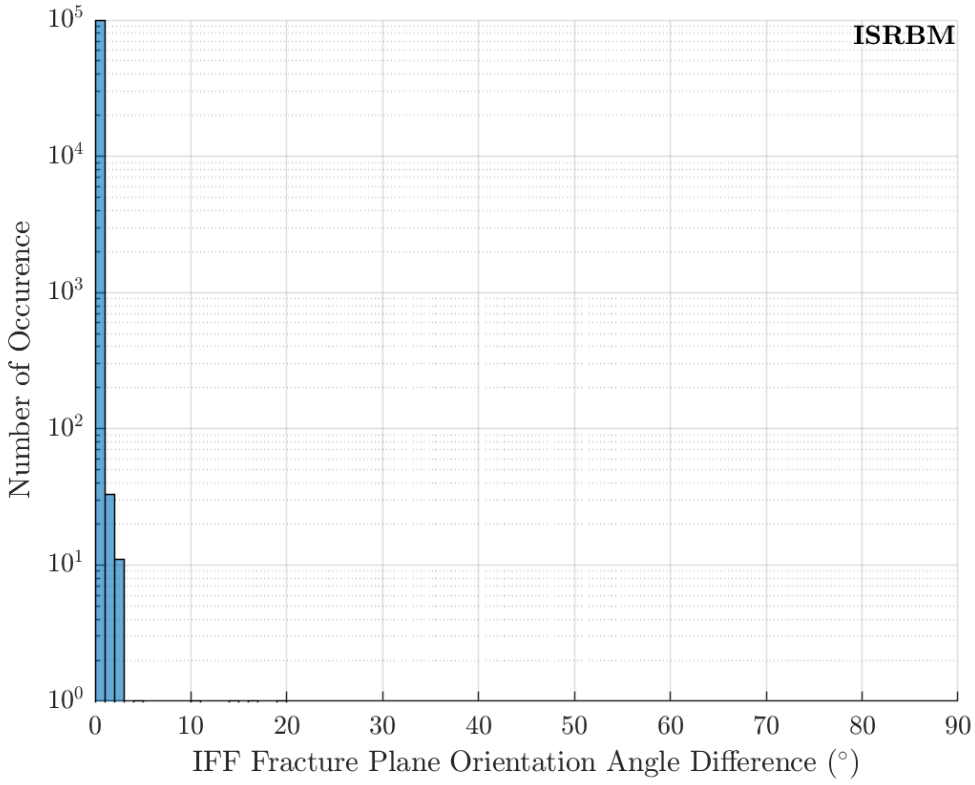
(a) EGSS



(b) SRGSS



(c) SPIS



(d) ISRBM

Figure 4.14: Distribution of 1×10^5 randomised stress cases on the FPOA difference for (**a**) EGSS, (**b**) SRGSS, (**c**) SPIS and (**d**) ISRBM to the Puck SSM $0.01°$ increment search

According to Table 4.3, the ISRBM search algorithm has the slowest computation time among the other fast FPOA search algorithms. However, Table 4.3 shows that the ISRBM search algorithm has the most reliable and consistent FPOA and failure exposure result since it has the average lowest difference result compared to the other FPOA search algorithms. Figure 4.14 captures the consistency of the ISRBM search algorithm to obtain FPOA. The figure shows that most of the ISRBM search algorithm calculation result has less than $3°$ FPOA difference when compared to the Puck SSM algorithm with $0.01°$ increment. In contrast, the other FPOA search algorithms have more scattered results. The slower computation time in the ISRBM search algorithm is the result of a more reliable and rigorous error-checking algorithm.

Another validation is conducted by comparing the stress tensor at the material failure point between the Puck failure criterion and the experimental test. For this validation, the biaxial failure envelope $\sigma_{22} - \tau_{12}$ test result on e-glass/LY556 [89] is used as a benchmark. For comparison, both standard Puck SSM and ISRBM search algorithms are used alternatively to calculate the IFF failure exposure. At the IFF failure exposure $f_{E,IFF} = 1$ (the irreversible damage on the material is started), the $\sigma_{22} - \tau_{12}$ result values are overlaid on top of the experimental test result. Figure 4.15 shows that the both

search algorithm Puck failure criterion is able to predict the result from the experimental result accurately.

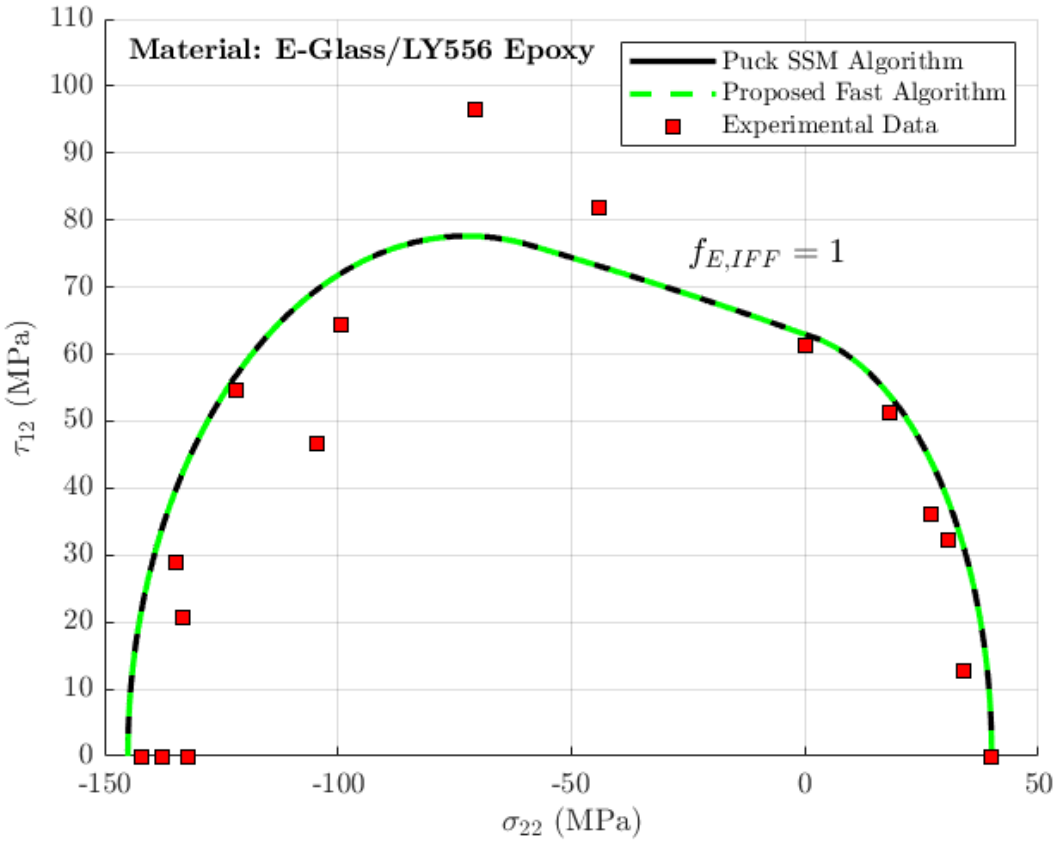


Figure 4.15: The failure envelope $\sigma_{22} - \theta_{12}$ of 0° uniderectional e-glass/LY566 epoxy biaxial test result

To implement the proposed fast FPOA search algorithm into the Abaqus FEA software, the UMAT subroutine previously developed to model the 3D Puck failure criterion with exponential damage degradation model is modified. The standard FPOA search algorithm, SSM search algorithm, in the original UMAT subroutine is replaced with the proposed FPOA search algorithm, ISRBM search algorithm, which is more reliable and faster. The flowchart of the modified UMAT subroutine with the proposed fast FPOA search algorithm subroutine is illustrated in Figure 4.16 and the search algorithm process is highlighted in red. Inside the modified UMAT subroutine, a new calling function along with several additional functions, has been introduced to replace the original FPOA search algorithm. Details of the UMAT subroutine modifications are provided in Appendix C.

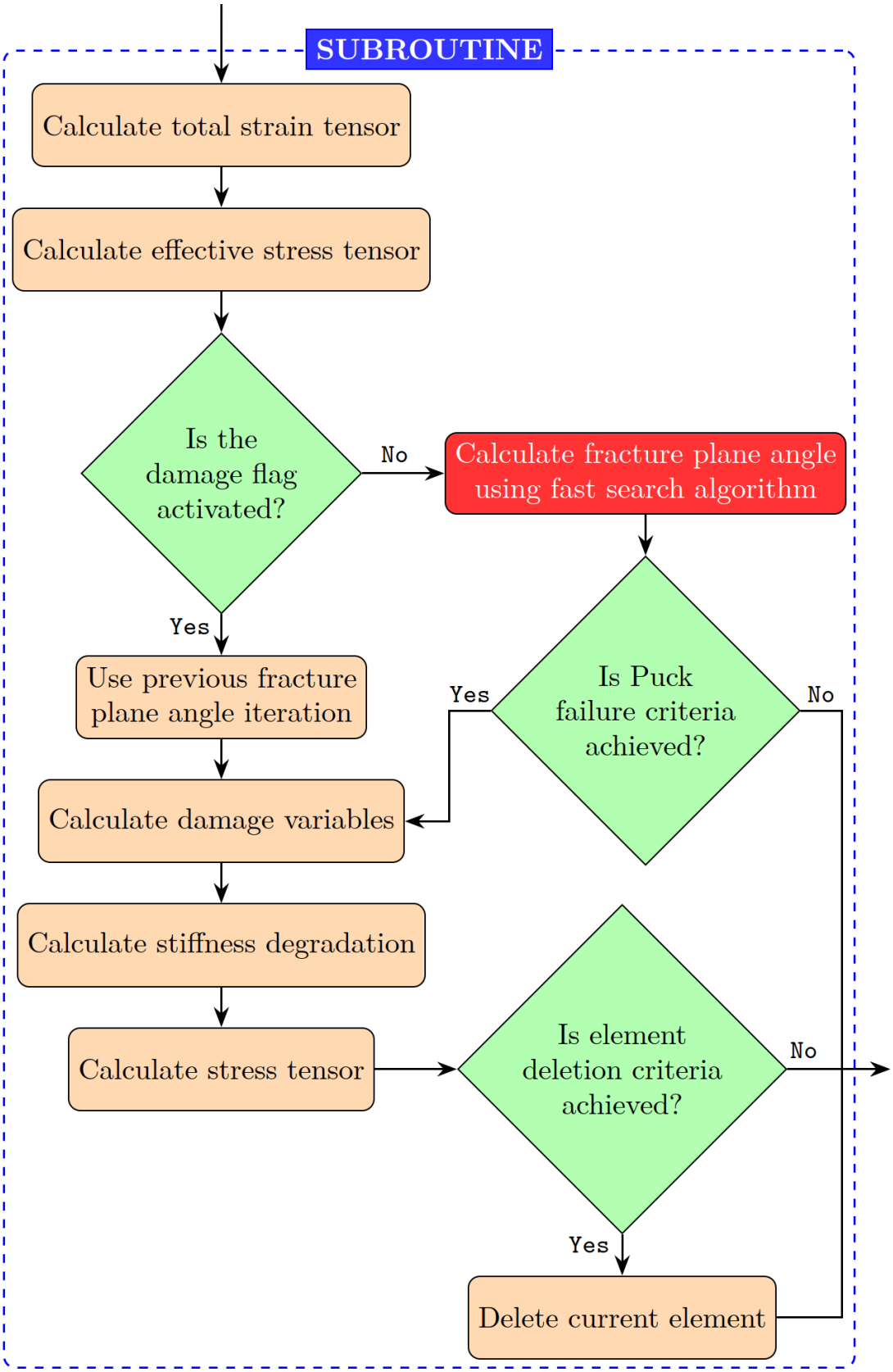


Figure 4.16: Flowchart of the Puck failure criterion with proposed fast FPOA search algorithm subroutine

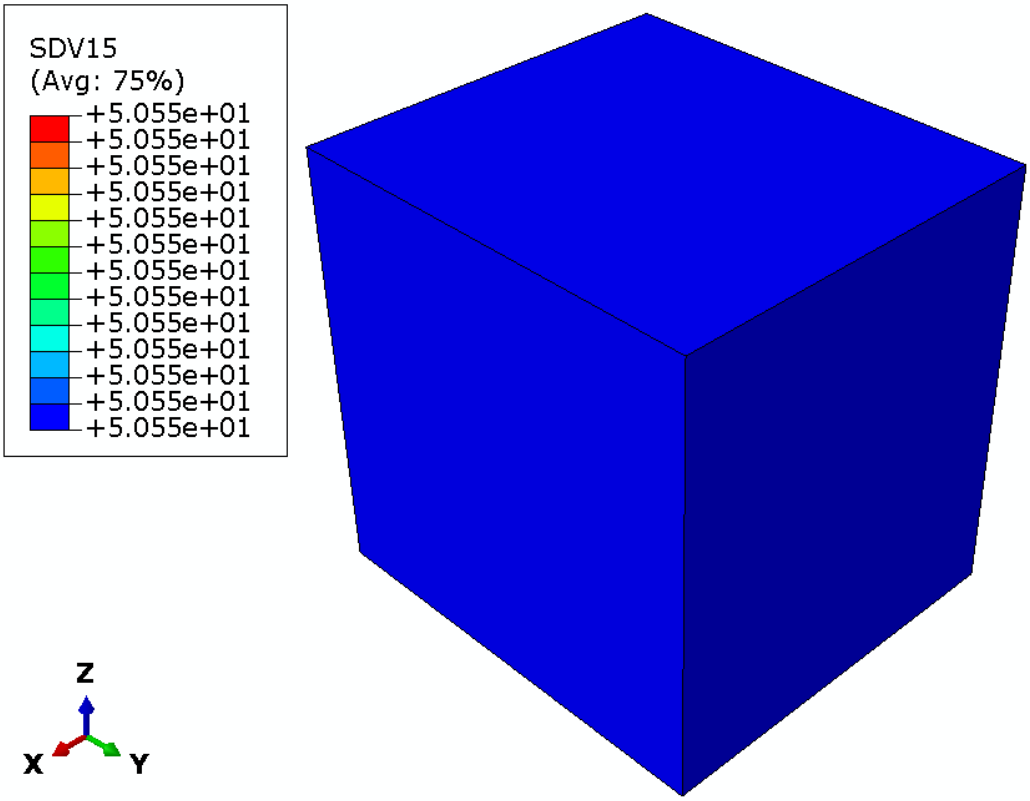


Figure 4.17: The orientation angle of the fracture plane on the single-element test using ISRBM search algorithm

Using a similar single-element test FEA simulation setup and material properties as described in **subsection 3.2.1**, the orientation angle of the fracture plane calculated with the ISRBM search algorithm ($SDV15$ in Figure 4.17) is $\theta_{fp,ISRBM} = 50.55°$. This current result is similar to the result obtained using standard SSM search algorithm, $\theta_{fp,SSM} = 51°$ ($SDV15$ in Figure 3.8) and still within the range of the experimental result, $\theta_{fp} = 53° \pm 3°$, reported by Pinho et al. [74].

4.3 Summary

In this chapter, a more reliable implementation of fast FPOA search algorithm, the ISRBM search algorithm is proposed to substitute the standard Puck SSM algorithm. Several improvements have been implemented into the ISRBM search algorithm, such as additional supporting points for more robust selective range interval testing, particularly at the FPOA near $-90°/90°$, and a more robust Brent's method for finding the failure exposure maximum value within selective range interval. Albeit the computational performance of the ISRBM search algorithm is slower compared to other fast FPOA search algorithms, the ISRBM search algorithm calculation result is exceptionally reliable.

Chapter 5

Custom Failure Criterion on XFEM Frameworks

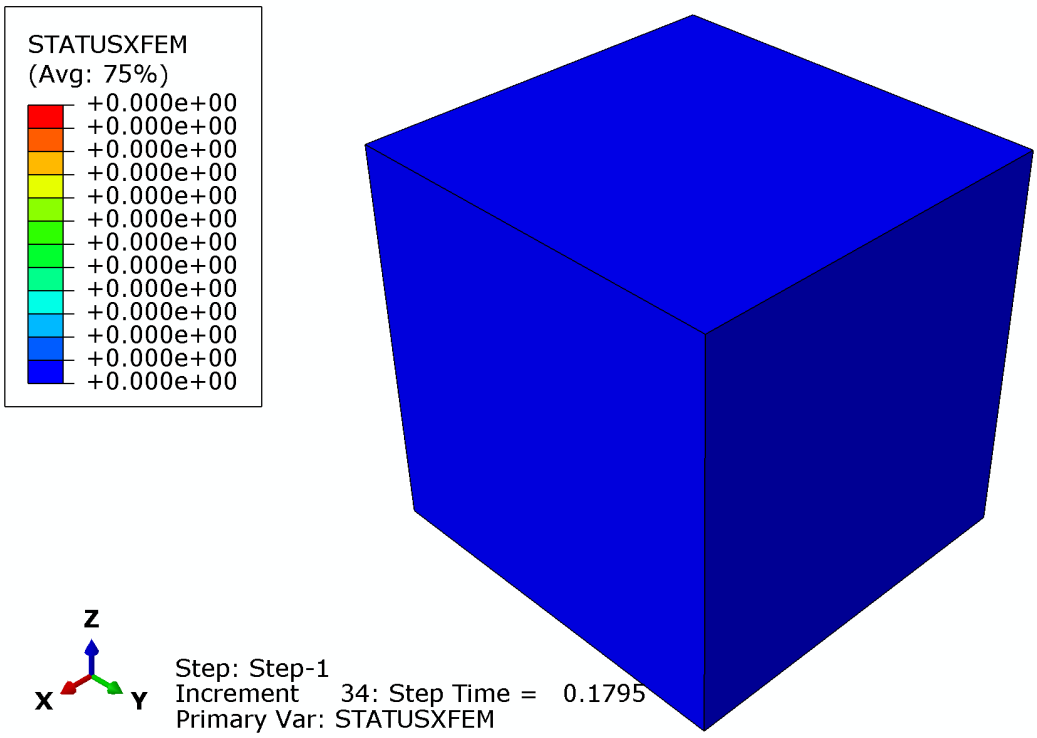
In this chapter, an Extended Finite Element Method (XFEM) with custom failure criterion using 3D Puck failure criterion is presented. XFEM is an advance finite element method that employs an enrichment function to the elements that specifically affected by discontinuities, making it highly effective for simulating crack propagation within the finite element model. In XFEM, a damage or failure criterion is required to evaluate the crack initiation before crack could propagate. Therefore, an appropriate failure criterion is essential for achieving accurate finite element simulations.

However, the built-in failure criterion library in the commercial FEA software is limited. As a result, a custom failure criterion has to be developed from the ground up to facilitate a more advance failure criterion model In Abaqus FEA software, the custom failure criterion is implemented into the UDMGINI subroutine. For this simulation, a 3D Puck failure criterion is developed and implemented into the UDMGINI subroutine to simulate the damage initiation on the composite structures, reusing syntax and formulas from the previous UMAT subroutine designed for a progressive damage model.

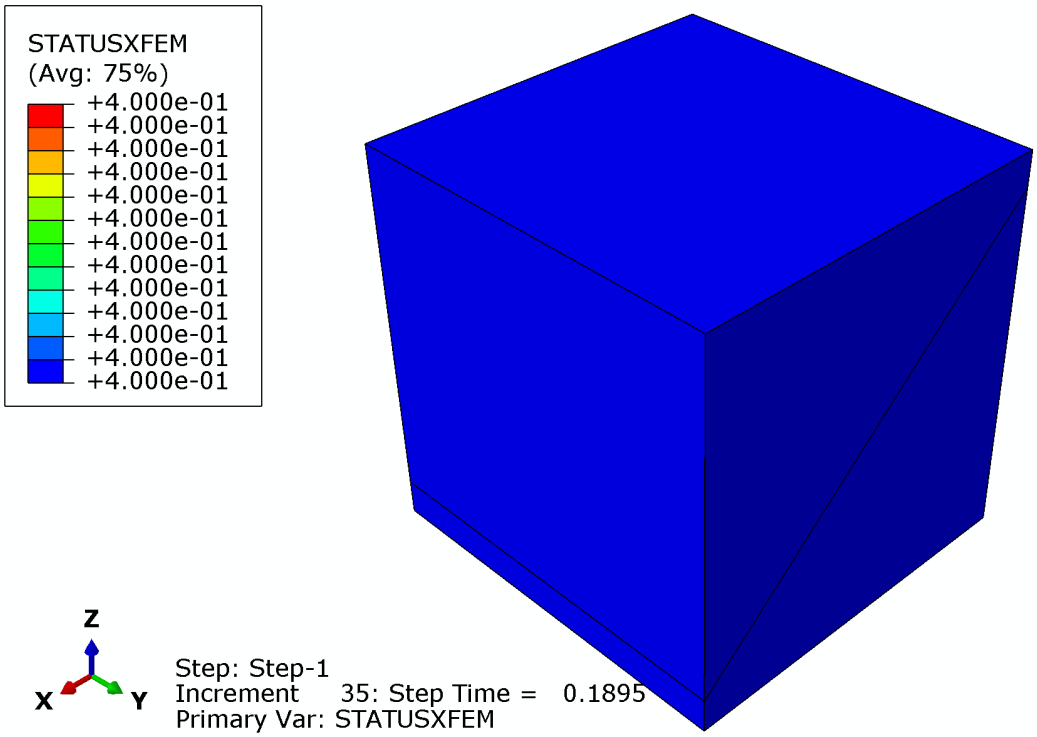
In the UDMGINI subroutine, each failure modes and its corresponding failure plane orientation angle has to be defined. The current UDMGINI subroutine includes a total of four distinct failure modes: the 3D Puck Fibre-Failure (FF) criterion in both tension and compression, as well as the 3D Puck Inter-Fibre Failure (IFF) criterion, also in tension and compression.

For 3D Puck FF criterion, the corresponding failure plane orientation angle is perpendicular to the fibre direction for both tensile and compressive load. Meanwhile, for 3D Puck IFF criterion, the corresponding failure plane orientation angle is calculated using formula as shown on the Equations (3.9) to (3.13). The final UDMGINI subroutine containing the 3D Puck failure criterion and its corresponding fracture plane orientation angle for each failure modes is presented in Appendix D.

5.1 Validation on the XFEM Implementation

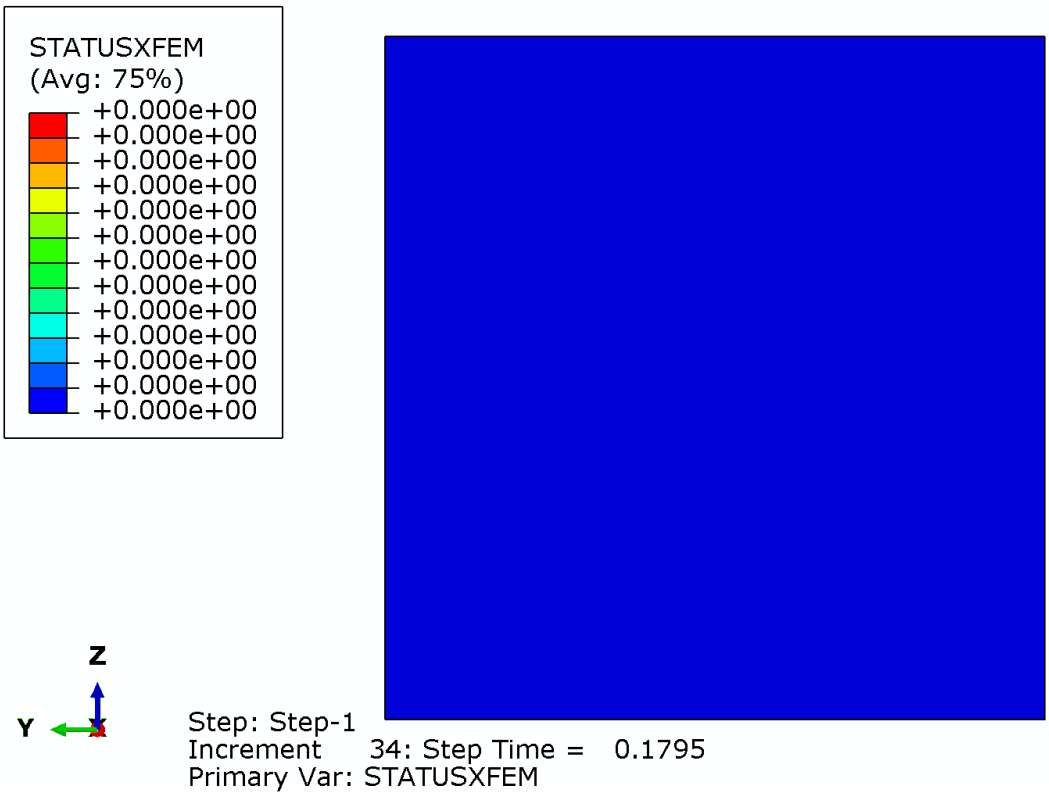


(a) Cube element before crack initiates

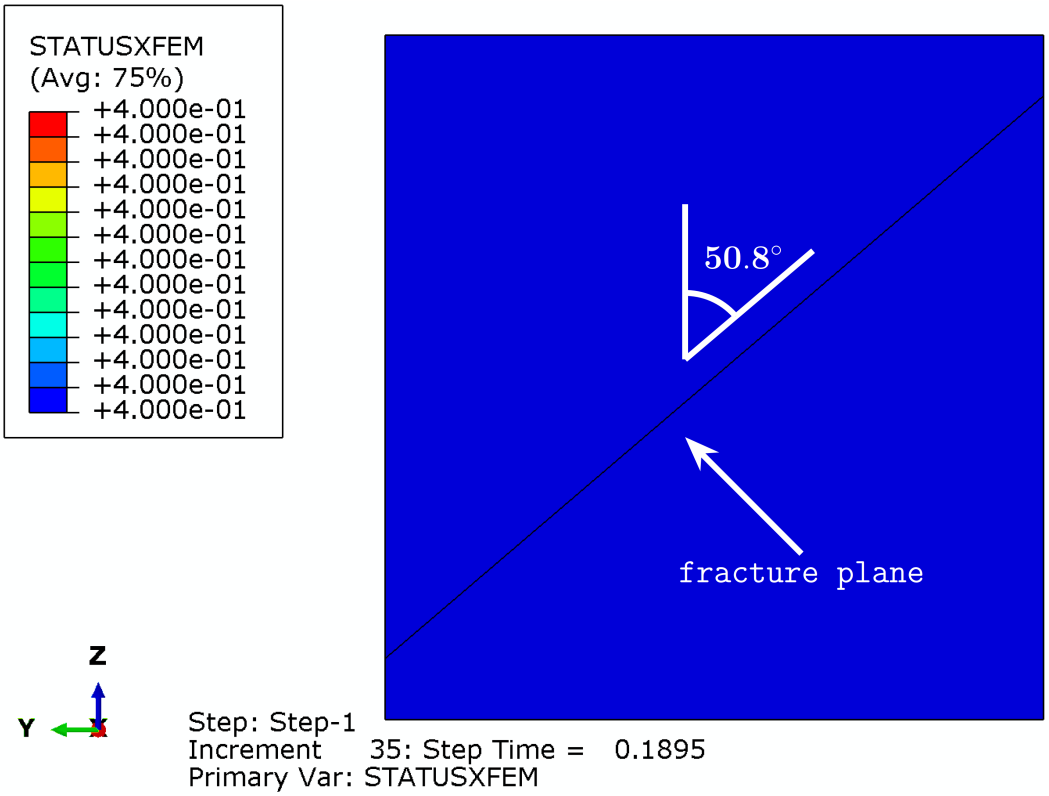


(b) Cube element at crack initiation

Figure 5.1: Isometric view of the cube element under compression load on single-element test



(a) Cube element before crack initiates



(b) Cube element at crack initiation

Figure 5.2: The orientation angle of the fracture plane on the single-element test using ISRBM search algorithm in the XFEM framework

A similar single-element test, as previously described in **subsection 3.2.1**, is employed to validate the implementation of the UDMGINI subroutine within the Abaqus FEA software. A compressive load is applied to one side of the cube element, while a fixed support is established on the opposite side. The details of the test are illustrated in Figure 3.6. IM7/8552 CFRP is used as the material input for the single-element test, as outlined in Table 4.1.

The results of the single-element test are presented in Figure 5.1, which illustrates the crack progression in the cube element. The STATUSXFEM shows the severity of the damage on the element, with 0 is for undamaged element and 1 is for fully damaged element. During increment 34 (Figure 5.1a), the cube element remains intact. However, on the next increment, increment 35 (Figure 5.1b), the cube is divided into two regions by a fracture plane.

Figure 5.2 provides a detailed view of the cube element before (Figure 5.2a) and after (Figure 5.2b) crack initiation. As shown in the aforementioned figures, the orientation angle of the fracture plane is $\theta_{fp,XFEM} = 50.8°$, which is still within the tolerance of the experimental results reported by Pinho et al. [74] at $\theta_{fp} = 53° \pm 3°$.

In contrast to the progressive damage model discussed in the previous chapter, the XFEM approach allows for the fracture physically propagates through the element. Therefore, the orientation angle of the fracture plane could be directly observed on the finite element model.

5.2 Composite Wing Subjected to the Gust Load

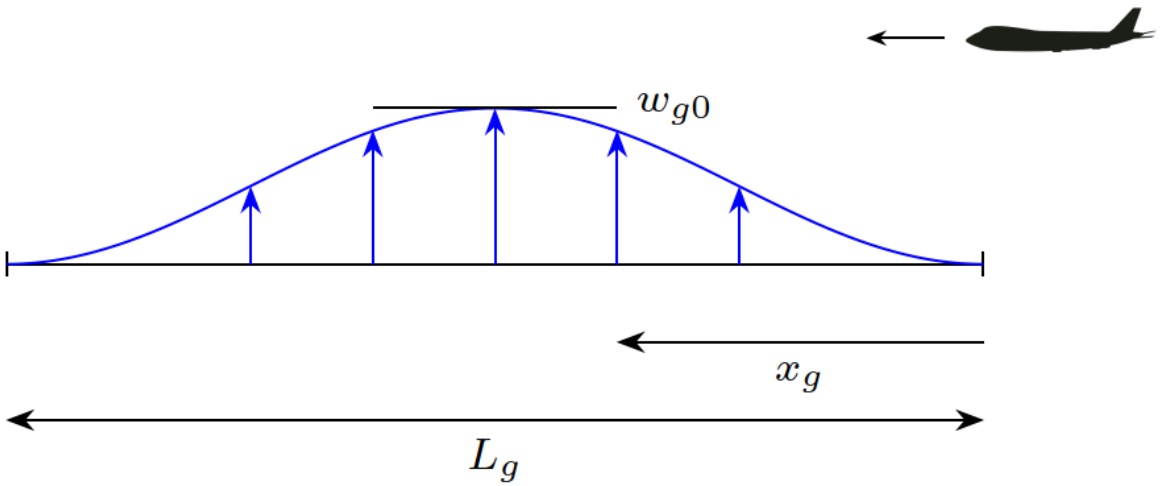


Figure 5.3: Vertical wind gust load

In this section, the implementation of XFEM with 3D Puck failure criterion for a lifting structure, i.e., wing, is discussed. For a wing structure exerted by aerodynamic loads,

the failure modes critically influenced by the vibration modes. As the structure exposed to the airflow and disturbance, the vibration response is a function of various modes, i.e., bending and torsion, so-called aeroelastic phenomenon. Therefore, a study on the fracture of wing structure is an interesting case to be investigated. Figure 5.3 shows the illustration of an aircraft flying through a region where a vertical wind gust occurs. In this case, the gust load becomes an external disturbance which excite dynamic response of the aircraft structure.

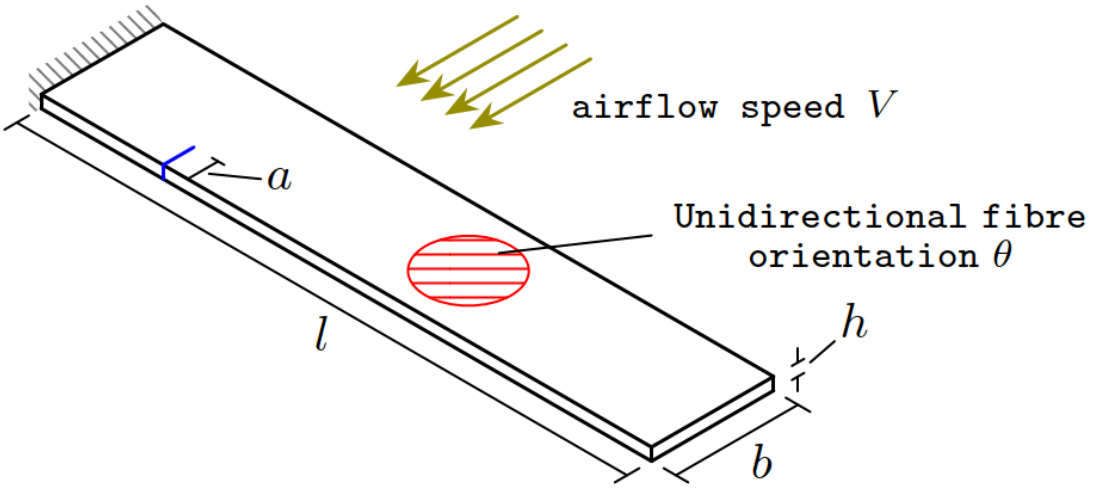


Figure 5.4: Planar unidirectional composite lifting surface [27]

In the current work, a planar lifting structure made of a 0° unidirectional composite by Wang et al. [27] is evaluated. The dimensions of the composite lifting structure are as follow: half-span $l = 500mm$, chord length $b = 100mm$, and chord thickness $h = 5mm$. In this model, the fibre direction of the unidirectional composite is parallel to the span of the planar lifting surface.

The composite lifting structure is exposed to the airflow as depicted in Figure 5.4. As the airspeed increases, an aeroelastic instability problem may occur, i.e., flutter, and could lead to a structural failure. Abdullah et al. [8] discussed that the existence of damage, i.e., crack, could potentially alter the instability boundary. Moreover, for a composite structure which has multiple failure modes, the behaviour of damage with aeroelasticity effect is not easily predicted.

In the present study, two distinct models are examined: a damaged composite lifting surface featuring a crack initiation at 20% of the chord length, situated 20% along the half-span from the root (Figure 5.5), and an undamaged composite lifting surface without crack initiation (Figure 5.6). Both composite lifting structures are evaluated at its cruise airspeed, far from the flutter instability.

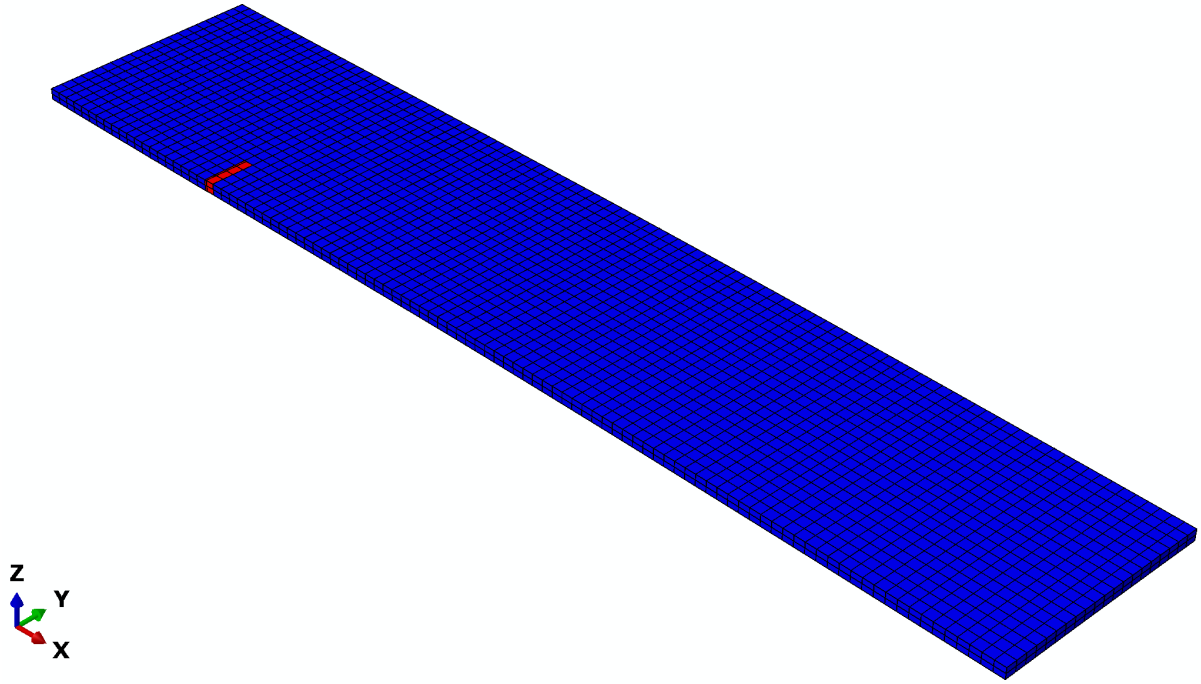


Figure 5.5: Finite element model of the damaged composite lifting surface with crack initiation located near the root

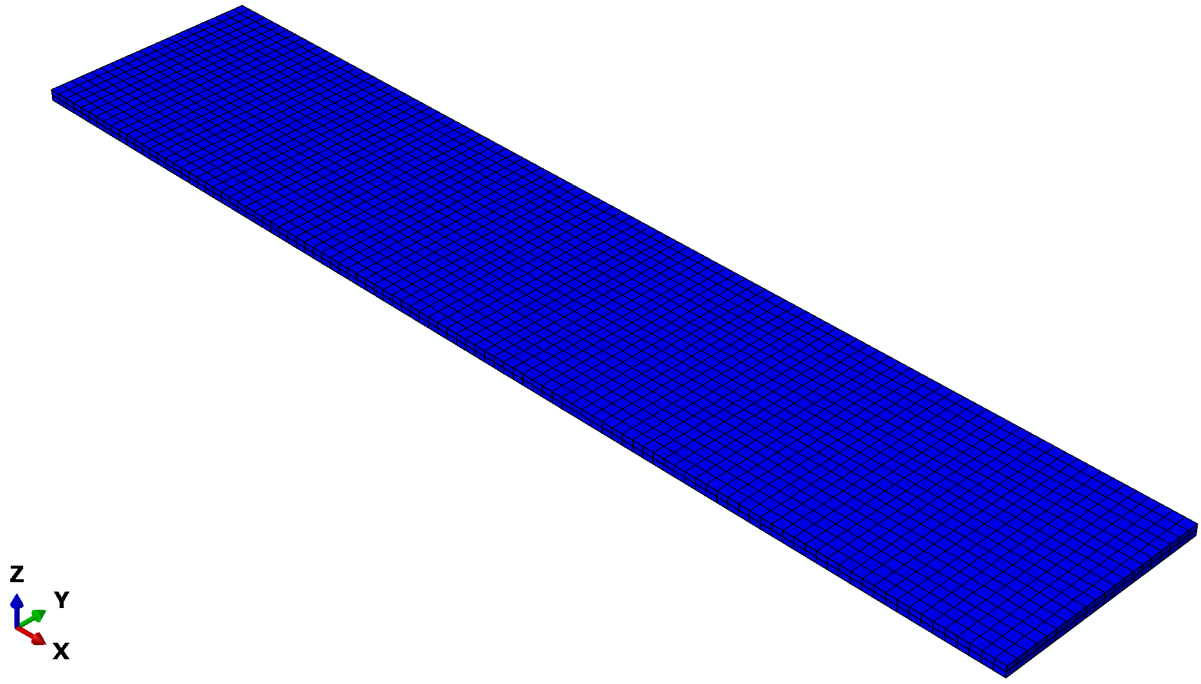


Figure 5.6: Finite element model of the undamaged composite lifting surface

A gust disturbance is introduced to excite vibration on the structure. In this case, the effect of the dynamic response to the structural damage is observed. Abdullah et al. [8] proposed a computational scheme to evaluate crack propagation of composite

lifting structure by means of XFEM. However, in their computational scheme, a standard failure criterion in commercial software, i.e., maximum principal stress, was used in conjunction with XFEM. Despite they were able to evaluate the crack propagation due to aeroelastic vibration, the use of maximum principal stress as a failure criterion for composite may yield to a different perception of damage behaviour.

Table 5.1: Fibre and matrix properties of the composite lifting surface [27]

Material Properties	Matrix	Fibre	Unit
Modulus of Elasticity	2.76	275	GPa
Poisson's Ratio	0.32	0.2	-
Shear Modulus	1.036	114.8	GPa
Mass Density	1600	1900	kg/m^3
Volume fraction	0.5	0.5	-

Wang et al. provide only the fibre and matrix properties of the composite material, as presented in Table 5.1. Hence, Mean Field Homogenisation (MFH) is employed using the Mori-Tanaka method [90, 91] to estimate the effective properties of the composite material based on the fibre and matrix composition. However, for the application of a 3D Puck failure criterion, additional data—such as material strength properties and fracture energies—are required, while the material data available from Wang et al. is rather limited. Therefore, in this work, different composite materials with more comprehensive data such as: T300/976, AS4/PEEK and IM7/8552 CFRP are proposed to replace the material used for the composite lifting surface from Wang et al.

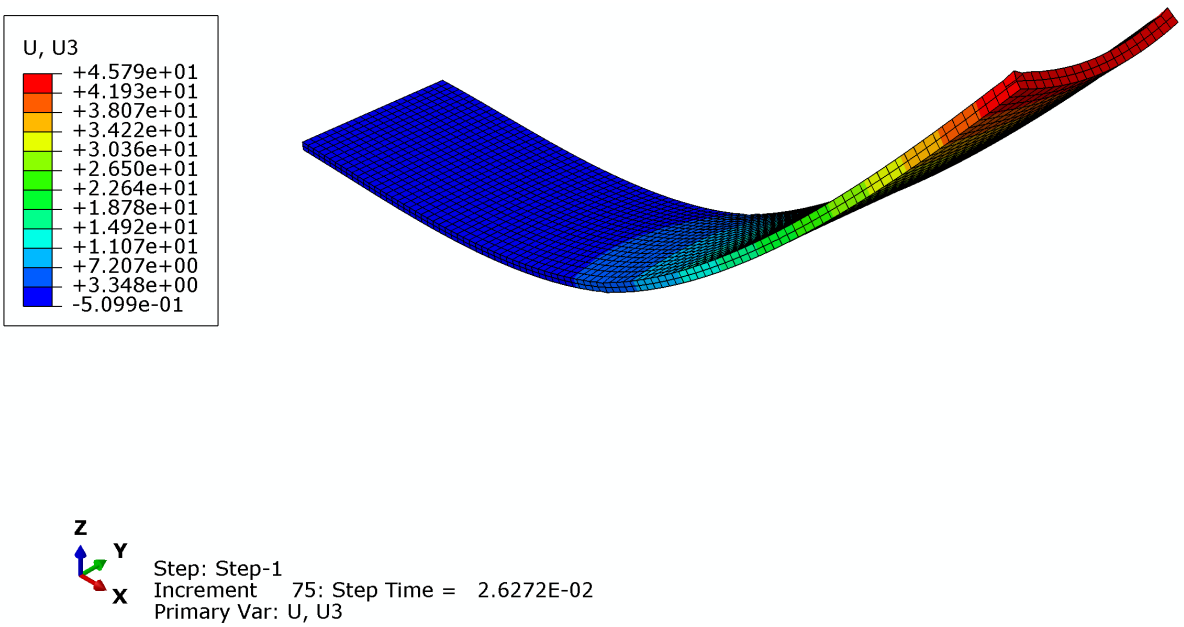
Table 5.2: Comparison of different composite material properties

	Mori-Tanaka	T300/976	Diff(%)	AS4/PEEK	Diff (%)	IM7/8552	Diff (%)
E_{11}	139.19 GPa	139.7 GPa	0.37	127.6 GPa	9.08	171.42 GPa	18.80
E_{22}	7.65 GPa	12.9 GPa	68.63	10.3 GPa	25.73	9.08 GPa	15.75
E_{33}	7.65 GPa	12.9 GPa	68.63	10.3 GPa	25.73	9.08 GPa	15.75
G_{12}	4.9 GPa	6.9 GPa	40.81	6 GPa	18.33	5.39 GPa	9.09
Average Diff.			44.61		19.72		14.85

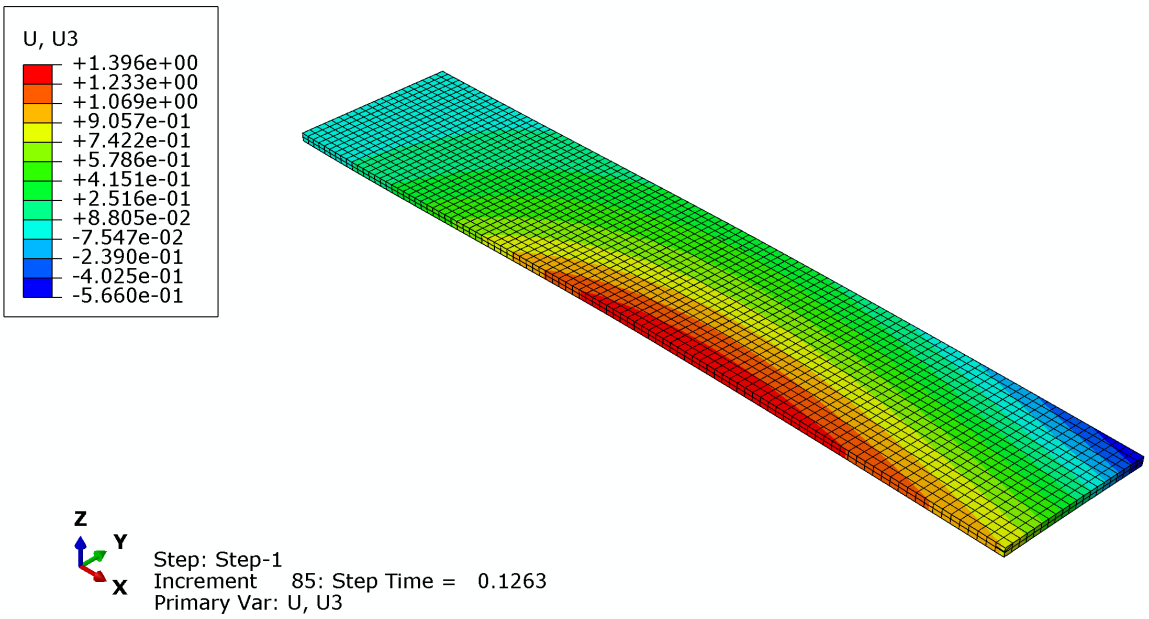
According to Table 5.2, IM7/8552 CFRP has the least average difference in material properties value compared to other composite material. For that reason, IM7/8552 CFRP is selected as the replacement material for the composite lifting surface from Wang et al.

Table 4.1 depicts the material properties of the composite structure used in the present case. As it can be seen, the elasticities, strengths and fracture energies are different for each direction. Hence, if a single value of maximum principal stress criterion is used will not represent the overall behaviour of the composite at its failure limit.

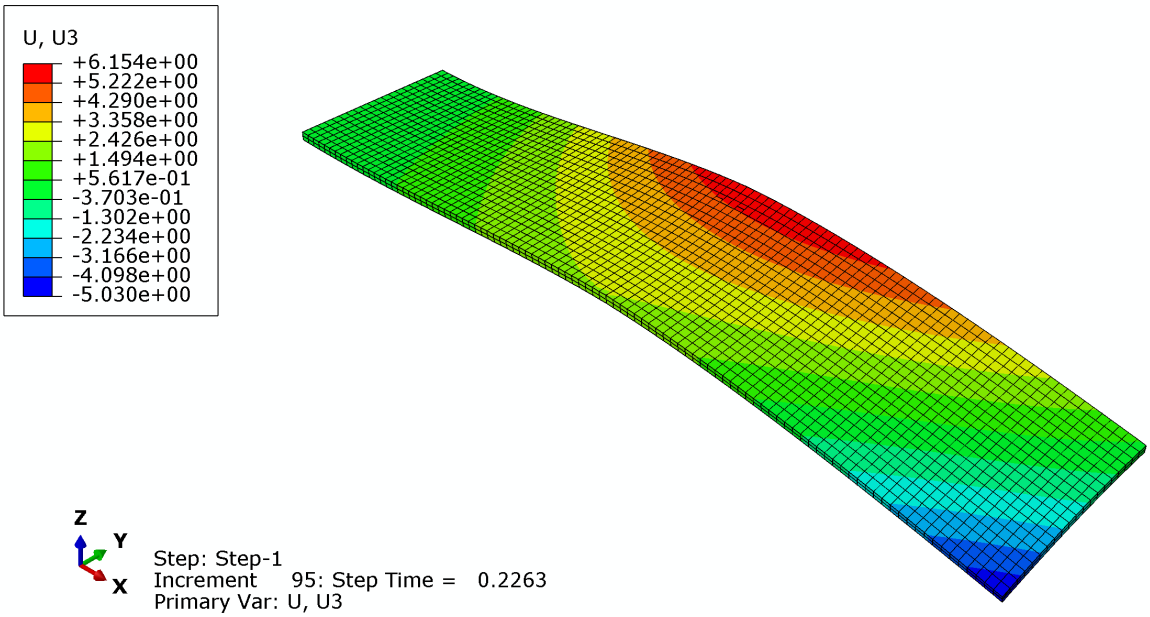
Therefore, in the present case, the 3D Puck failure criterion is implemented to accommodate the effect of strengths and fracture energies in different directions. Figure 5.7 shows the displacement (in millimeter) due to the vibration response (scaled by a factor of 5 for improved visibility) of the composite plate due to aeroelastic loads. In this case, the composite structure is exposed to a uniform airflow of 72 m/s and a sharp gust with amplitude of 3.6 m/s. In accordance with ESDU69023 [92], the gust amplitude is relatively small; however, it has 60% probability of being encountered during a flight.



(a) Composite lifting surface displacement plot at 0.026 s

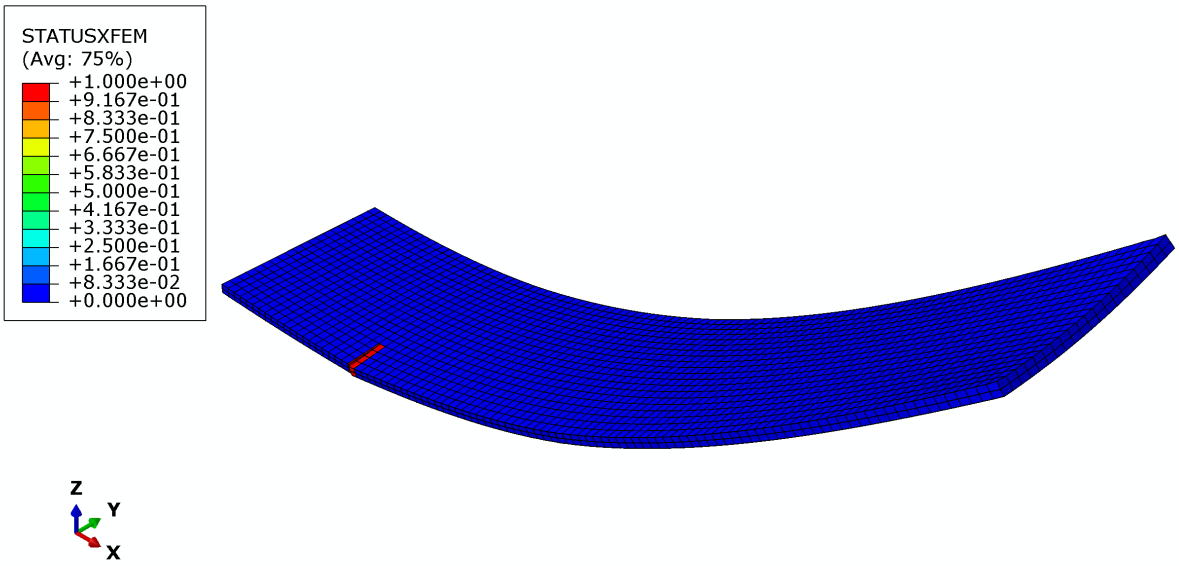


(b) Composite lifting surface displacement plot at 0.126 s

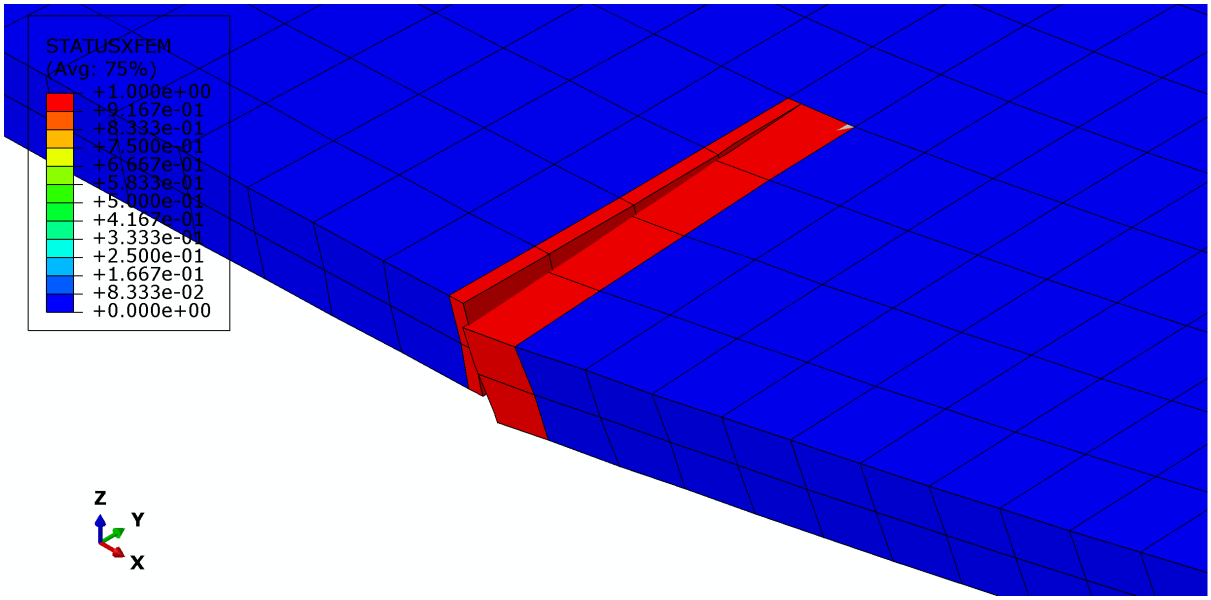


(c) Composite lifting surface displacement plot at 0.226 s

Figure 5.7: Composite lifting surface displacement progression plot. The displacement (U) is in millimeter



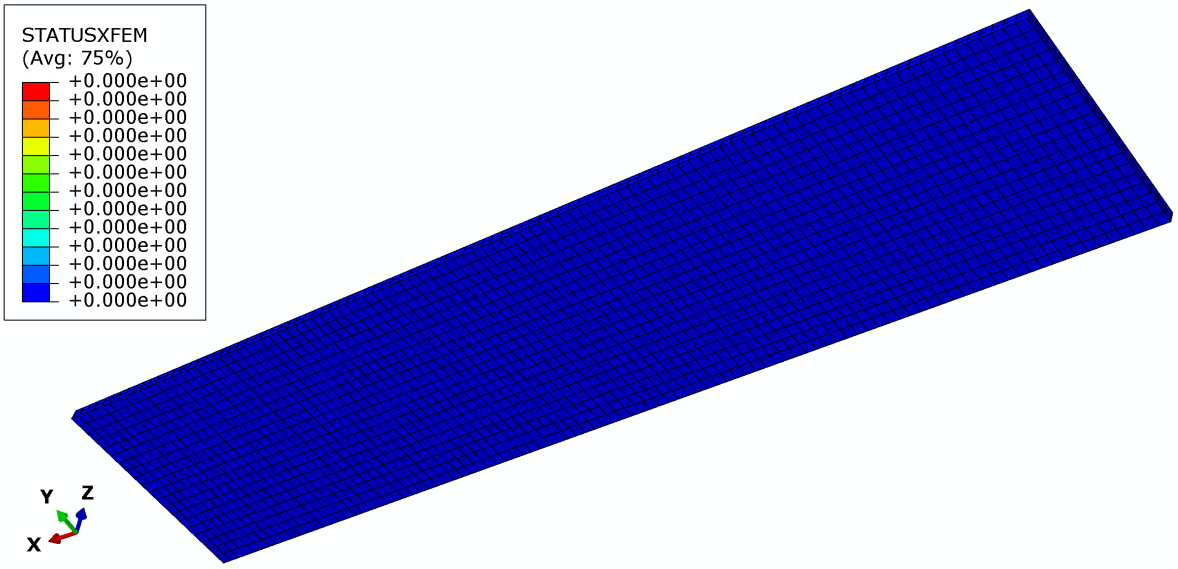
(a) Damage plot on the damaged composite lifting surface after subjected to gust load



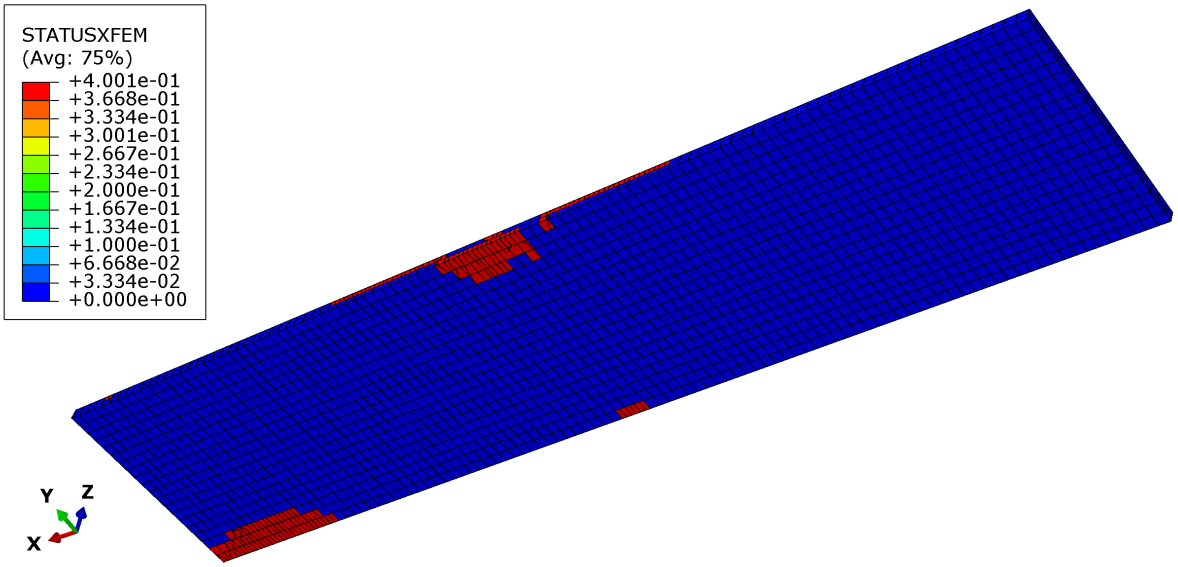
(b) Damage plot near crack initiation

Figure 5.8: Damage plot on the damaged composite lifting surface

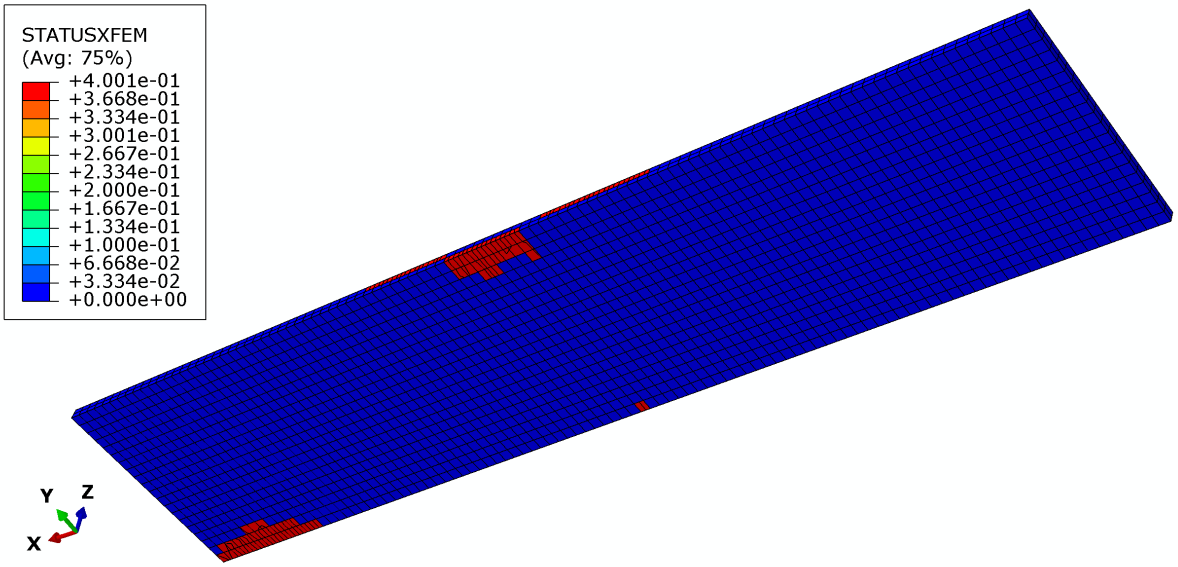
The comparison of XFEM with maximum principal stress and 3D Puck failure criterion model is investigated. In the case of maximum principal stress failure criterion, only a single value of strength and fracture energy could be used in a simulation; thus, different simulations using strength and fracture energy from each direction, i.e., longitudinal, transversal and in-plane shear, are conducted.



(a) Damage plot using maximum principal stress failure criterion with longitudinal strength value



(b) Damage plot using maximum principal stress failure criterion with transverse strength value



(c) Damage plot using maximum principal stress failure criterion with shear strength value

Figure 5.9: Damage plot at the lower surface of the undamaged composite lifting surface using maximum principal stress failure criterion.

Despite the fact that 3D Puck failure criterion may provide a more realistic fracture scenario, in the case of composite structure with crack initiation, there is no noticable difference between 3D Puck failure criterion model and the one with maximum principal stress failure criterion model (Figure 5.8a), with no further propagation on the crack initiation (Figure 5.8b). However, for undamaged composite structure, Figure 5.9 show how the crack(s) may occur in the composite lifting structure for different strength and fracture energy. In this case, the cracks on the structure occur due to inability of the material to withstand the load.

As shown in Figure 5.9a, as the longitudinal strength is the highest value compared to the strengths in other directions, there is no crack on the structure when it is exposed to the gust load. In contrast, as seen in Figure 5.9b and 5.9c, the cracks occur on the lower surface as the transversal or in-plane shear strength is used as the maximum principal stress. Figure 5.9b and 5.9c show almost a similar cracked area, especially at around the edge of mid-span as the vibration response is dominated by higher degree modes, i.e., 2nd and 3rd bending modes with a bit mix of torsional mode. In this case, the strength and fracture energy are assumed to be "isotropic" as it applies the same value to all direction. However, in the current study, 3D Puck failure criterion model is applied to represent the strengths and fracture energies in all different directions of the composite.

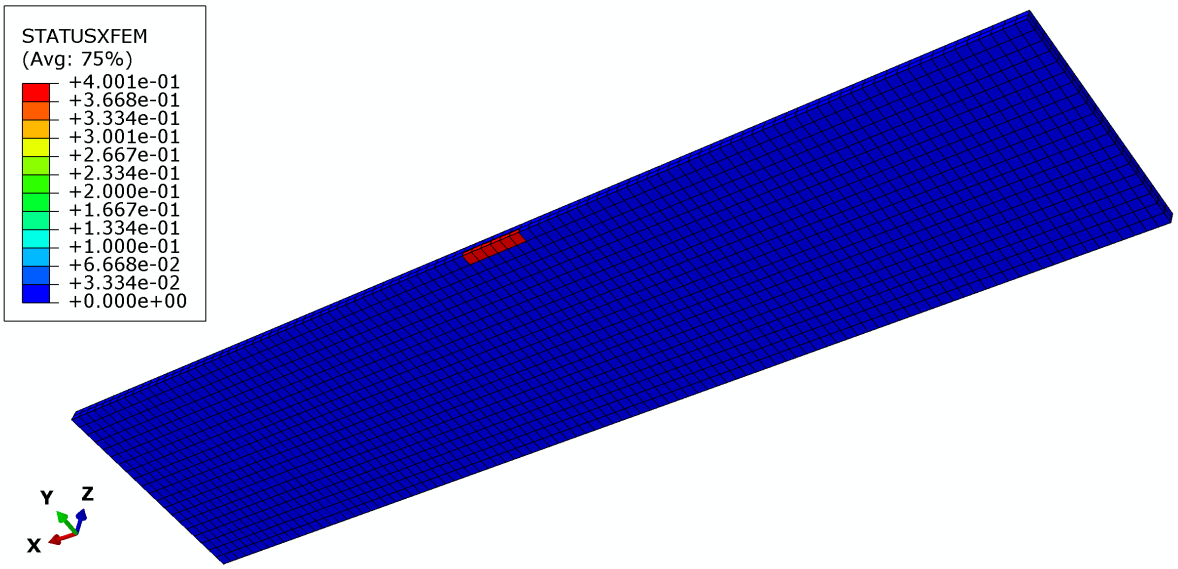


Figure 5.10: Damage plot at the lower surface of the undamaged composite lifting surface using 3D Puck failure criterion

Figure 5.10 shows the cracked area as the 3D Puck failure criterion model is implemented. As it can be seen, at the lower surface, cracks also occur at the edge of the mid-span. However, the affected area is smaller compared to the ones with only transversal or in-plane shear strength as the maximum principal stress criteria. In this case, by using 3D Puck failure criterion model, the composite ability to withstand the load is influenced by the strengths and fracture energies in all directions. As bending modes are the dominant shapes in the dynamic response of this structure, the normal stresses on the longitudinal direction still able to be resisted by the longitudinal strength of the composite. However, with torsion mode can also be seen, the transversal and in-plane shear stresses occur in critical area; thus, failure (crack) propagates.

5.3 Summary

The 3D Puck failure criterion has been successfully implemented with the XFEM to evaluate the crack propagation on composite structures. As presented in this chapter, a case study of a composite lifting structure (wing) under aeroelastic vibration has been investigated. By means of 3D Puck failure criterion, the crack location and propagation may define a much more realistic case. In contrast, in the available FEA software, the XFEM module is only considered with a single value of maximum principal stress as the failure criterion. Hence, in the case of the composite structure with various strength value for different direction, 3D Puck failure criterion is more accurate compared to the maximum principal stress failure criterion.

Chapter 6

Conclusions

In the present work, advanced computational methods have been implemented to evaluate structural integrity on composite structures. Various computational schemes have been applied to different case studies, including a composite lifting structure (wing) under aeroelastic vibration.

Part of this work focused on the development of a scheme to accelerate computational process of damage evolution in composite structures. A computational code based on the 3D Puck failure criterion has been built and employed to evaluate damage evolution on several benchmark cases. An improved fast fracture plane orientation angle search algorithm augmented to the code and has been proven to increase reliability while maintaining relatively fast computing time.

In a commercial finite element analysis (FEA) software, currently, only a single value of maximum principal stress criteria can be applied in conjunction with the Extended Finite Element Method (XFEM). Therefore, it is not possible to evaluate the case of a composite structure with different strengths and fracture energies in each direction, i.e., longitudinal and transversal. Therefore, 3D Puck failure criterion model is applied to accommodate different characteristics of composite in each direction.

Lastly, the computational scheme with 3D Puck, XFEM and fast fracture plane orientation angle search algorithm has been applied to evaluate crack propagation on a composite lifting structure. In this case, aeroelastic vibration due to cruise load and gust disturbance was applied as the main source of excitation, causing deformation to the structure. The dynamic responses constructed from different mode shapes, i.e., bending and torsion modes, provided complexity to the evaluation of damage (crack).

The comparisons of the results obtained from the implementation of maximum principal stress and 3D Puck failure criterion depicted different fracture scenarios could occurred. The main difficulty in using a single value of maximum principal stress is the selection of the material strength. In the composite material, it is known that at one direction, the strength could be much higher than other directions. Thus, by only selecting a single value of strength could lead to inaccurate results, i.e., no damage /

crack occurred when the highest strength value is applied. In contrast, by means of 3D Puck failure criterion, different strengths on each direction could be represented; thus, could display a much more realistic fracture scenario.

Future Works

Based on the present work, several improvements could be implemented for similar research topic in the future to name some of few:

Progressive Damage Model

- Use different combination of degradation damage model such as linear or B-K criterion on the progressive damage model and compare the results.

Fast Fracture Plane Orientation Search Algorithm

- Apply different minimisation/maximisation algorithm to substitute Brent's method.

XFEM Implementation on the Composite Wing

- Perform XFEM simulation on various unidirectional composite fibre orientation.
- Compare different combination of stacking sequence of the composite laminate on the composite wing structure.

Higher Fidelity Fluid-Structure Interaction (FSI) Evaluation

- Apply a higher fidelity computational fluid dynamics (CFD) scheme, such as, Reynolds-Averaged Navier Stokes (RANS).
- Implement a two-ways FSI computational scheme to accommodate changes in structure and fluid domains due to the occurrence and/or propagation of crack.

Appendix A

Constitutive Equation Derivation

The derivation of the constitutive equation for the progressive damage model, utilising the three-dimensional Puck failure criterion, is presented in the equations below. The subscript i denotes the damage variables associated with fibre (f) and matrix failures ($m1, m2$), while the subscripts t and c refer to tensile and compressive loads, respectively. The secant constitutive equation is derived from the undamaged segment of the material's elastic constitutive equation. Consequently, the secant constitutive equation for the material can be expressed as follows:

$$\begin{aligned}\mathbb{C}^{sec} &= \Sigma_i(1-\delta_{it})(1-\delta_{ic})\mathbb{C}_i^e \\ &= (1-\delta_{ft})(1-\delta_{fc})\mathbb{C}_f^e + (1-\delta_{m1t})(1-\delta_{m1c})\mathbb{C}_{m1}^e + (1-\delta_{m2t})(1-\delta_{m2c})\mathbb{C}_{m2}^e\end{aligned}$$

The tangent constitutive equation is subsequently derived as follows:

$$\begin{aligned}\mathbb{C}^{tangent} &= \mathbb{C}^{sec} - \Sigma_i(1-\delta_{ic})\frac{\partial\delta_{it}}{\partial\epsilon}\otimes\bar{\sigma}_i - \Sigma_i(1-\delta_{it})\frac{\partial\delta_{ic}}{\partial\epsilon}\otimes\bar{\sigma}_i \\ &= \Sigma_i(1-\delta_{it})(1-\delta_{ic})\mathbb{C}_i^e - \Sigma_i(1-\delta_{ic})\frac{\partial\delta_{it}}{\partial\epsilon}\otimes\bar{\sigma}_i - \Sigma_i(1-\delta_{it})\frac{\partial\delta_{ic}}{\partial\epsilon}\otimes\bar{\sigma}_i \\ &= (1-\delta_{ft})(1-\delta_{fc})\mathbb{C}_f^e + (1-\delta_{m1t})(1-\delta_{m1c})\mathbb{C}_{m1}^e + (1-\delta_{m2t})(1-\delta_{m2c})\mathbb{C}_{m2}^e \\ &- (1-\delta_{fc})\frac{\partial\delta_{ft}}{\partial\epsilon}\otimes\bar{\sigma}_f - (1-\delta_{ft})\frac{\partial\delta_{fc}}{\partial\epsilon}\otimes\bar{\sigma}_f \\ &- (1-\delta_{m1c})\frac{\partial\delta_{m1t}}{\partial\epsilon}\otimes\bar{\sigma}_{m1} - (1-\delta_{m1t})\frac{\partial\delta_{m1c}}{\partial\epsilon}\otimes\bar{\sigma}_{m1} \\ &- (1-\delta_{m2c})\frac{\partial\delta_{m2t}}{\partial\epsilon}\otimes\bar{\sigma}_{m2} - (1-\delta_{m2t})\frac{\partial\delta_{m2c}}{\partial\epsilon}\otimes\bar{\sigma}_{m2}\end{aligned}$$

Each term in the tensor product on the right-hand side of the tangent constitutive equation can be further expressed as:

$$\begin{aligned}
\frac{\partial \delta_{ft}}{\partial \epsilon} &= \frac{\partial \delta_{ft}}{\partial f_{E,FF+}} \left(\frac{\partial f_{E,FF+}}{\partial \bar{\sigma}} : \frac{\partial \bar{\sigma}}{\partial \epsilon} \right); \\
\frac{\partial \delta_{fc}}{\partial \epsilon} &= \frac{\partial \delta_{fc}}{\partial f_{E,FF-}} \left(\frac{\partial f_{E,FF-}}{\partial \bar{\sigma}} : \frac{\partial \bar{\sigma}}{\partial \epsilon} \right); \\
\frac{\partial \delta_{m1t}}{\partial \epsilon} &= \frac{\partial \delta_{m1t}}{\partial f_{E,IFF+}} \left(\frac{\partial f_{E,IFF+}}{\partial \bar{\sigma}} : \frac{\partial \bar{\sigma}}{\partial \epsilon} \right); \\
\frac{\partial \delta_{m1c}}{\partial \epsilon} &= \frac{\partial \delta_{m1c}}{\partial f_{E,IFF-}} \left(\frac{\partial f_{E,IFF-}}{\partial \bar{\sigma}} : \frac{\partial \bar{\sigma}}{\partial \epsilon} \right); \\
\frac{\partial \delta_{m2t}}{\partial \epsilon} &= \frac{\partial \delta_{m2t}}{\partial f_{E,IFF+}} \left(\frac{\partial f_{E,IFF+}}{\partial \bar{\sigma}} : \frac{\partial \bar{\sigma}}{\partial \epsilon} \right); \\
\frac{\partial \delta_{m2c}}{\partial \epsilon} &= \frac{\partial \delta_{m2c}}{\partial f_{E,IFF-}} \left(\frac{\partial f_{E,IFF-}}{\partial \bar{\sigma}} : \frac{\partial \bar{\sigma}}{\partial \epsilon} \right)
\end{aligned}$$

where,

$$\frac{\partial \delta_{fj}}{\partial f_{E,FF}} = \frac{1}{f_{E,FF}^2} \left(1 + A_f^j f_{E,FF}\right) \left(A_f^j (1 - f_{E,FF})\right); \text{ with } j = t, c$$

$$\frac{\partial \delta_{mij}}{\partial f_{E,IFF}} = \frac{1}{f_{E,IFF}^2} \left(1 + A_{mi}^j f_{E,IFF}\right) \left(A_{mi}^j (1 - f_{E,IFF})\right); \text{ with } i = 1, 2;\ j = t, c$$

The inner product of the two tensors within the brackets for the fibre failure yields:

$$\begin{aligned}
\frac{\partial f_{E,FF\pm}}{\partial \epsilon_{11}} &= \frac{1}{R_{\parallel}^{t,c}} \left[\mathbb{C}_{11}^e - \left(\nu_{\perp\parallel} - \frac{E_{\parallel}}{E_{\parallel f}} \right) (\mathbb{C}_{21}^e + \mathbb{C}_{31}^e) \right] \\
\frac{\partial f_{E,FF\pm}}{\partial \epsilon_{22}} &= \frac{1}{R_{\parallel}^{t,c}} \left[\mathbb{C}_{12}^e - \left(\nu_{\perp\parallel} - \frac{E_{\parallel}}{E_{\parallel f}} \right) (\mathbb{C}_{22}^e + \mathbb{C}_{32}^e) \right] \\
\frac{\partial f_{E,FF\pm}}{\partial \epsilon_{33}} &= \frac{1}{R_{\parallel}^{t,c}} \left[\mathbb{C}_{13}^e - \left(\nu_{\perp\parallel} - \frac{E_{\parallel}}{E_{\parallel f}} \right) (\mathbb{C}_{23}^e + \mathbb{C}_{33}^e) \right] \\
\frac{\partial f_{E,FF\pm}}{\partial \epsilon_{12}} &= 0 \\
\frac{\partial f_{E,FF\pm}}{\partial \epsilon_{13}} &= 0 \\
\frac{\partial f_{E,FF\pm}}{\partial \epsilon_{23}} &= 0
\end{aligned}$$

However, for inter-fibre failure, additional calculations must be performed. The following equation represents a compacted form of the failure exposure equation for inter-fibre failure under both tension and compression.

$$f_{E,IFF\pm} = \sqrt{\mathcal{A}\bar{\sigma}_n^2(\theta_{fp}) + \mathcal{B}\bar{\tau}_{nt}^2(\theta_{fp}) + \mathcal{C}\bar{\tau}_{n1}^2(\theta_{fp})} + \mathcal{D}\bar{\sigma}_(\theta_{fp})$$

where $mathcal A$, $mathcal B$, $mathcal C$ and $mathcal D$ are material parameters that have different values depending on the loading mode. The material parameters for each variable are presented in the following table.

Material Parameters	Tension	Compression
$\mathcal{A}$	$\left(\frac{1}{R_{\perp}^{At}} - \frac{p_{\perp\psi}^{t}}{R_{\perp\psi}^{A}}\right)^2$	$\left(\frac{p_{\perp\psi}^{c}}{R_{\perp\psi}^{A}}\right)^2$
$\mathcal{B}$	$\left(\frac{1}{R_{\perp\perp}^{A}}\right)^2$	$\left(\frac{1}{R_{\perp\perp}^{A}}\right)^2$
$\mathcal{C}$	$\left(\frac{1}{R_{\parallel\perp}^{A}}\right)^2$	$\left(\frac{1}{R_{\parallel\perp}^{A}}\right)^2$
$\mathcal{D}$	$\frac{p_{\perp\psi}^{t}}{R_{\perp\psi}^{A}}$	$\frac{p_{\perp\psi}^{c}}{R_{\perp\psi}^{A}}$

The derivation of the stress tensor calculation on the potential fracture plane yields:

$$\sigma_n(\theta) = \epsilon_{11}\left(\mathbb{C}_{21}^{e}c^2 + \mathbb{C}_{31}^{e}s^2\right) + \epsilon_{22}\left(\mathbb{C}_{22}^{e}c^2 + \mathbb{C}_{32}^{e}s^2\right) + \epsilon_{33}\left(\mathbb{C}_{23}^{e}c^2 + \mathbb{C}_{33}^{e}s^2\right) + \gamma_{23}2cs\mathbb{C}_{66}^{e}$$
$$\tau_{nt}(\theta) = \epsilon_{11}\left(-\mathbb{C}_{21}^{e}c^2 + \mathbb{C}_{31}^{e}s^2\right) + \epsilon_{22}\left(-\mathbb{C}_{22}^{e}c^2 + \mathbb{C}_{32}^{e}s^2\right) + \epsilon_{33}\left(-\mathbb{C}_{23}^{e}c^2 + \mathbb{C}_{33}^{e}s^2\right) + \gamma_{23}\left(c^2 - s^2\right)\mathbb{C}_{66}^{e}$$
$$\tau_{n1}(\theta) = \gamma_{31}s\mathbb{C}_{55}^{e} + \gamma_{21}c\mathbb{C}_{44}^{e}$$

with,

$$\mathcal{R} = \mathcal{A}\bar{\sigma}_n^2(\theta_{fp}) + \mathcal{B}\bar{\tau}_{nt}^2(\theta_{fp}) + \mathcal{C}\bar{\tau}_{n1}^2(\theta_{fp})$$

The final derivations of the inner product of the failure exposure equation for inter-fibre failure under tension and compression are as follows:

$$\frac{\partial f_{E,IFF\pm}}{\partial \epsilon_{11}} = \frac{1}{2\sqrt{\mathcal{R}}}\left[2\mathcal{A}\left(\mathbb{C}_{21}^{e}c^2 + \mathbb{C}_{31}^{e}s^2\right)\bar{\sigma}_n + 2\mathcal{B}\left(-\mathbb{C}_{21}^{e}sc + \mathbb{C}_{31}^{e}sc\right)\bar{\tau}_{nt}\right] + \mathcal{D}\left(\mathbb{C}_{21}^{e}c^2 + \mathbb{C}_{31}^{e}s^2\right)$$
$$\frac{\partial f_{E,IFF\pm}}{\partial \epsilon_{22}} = \frac{1}{2\sqrt{\mathcal{R}}}\left[2\mathcal{A}\left(\mathbb{C}_{22}^{e}c^2 + \mathbb{C}_{32}^{e}s^2\right)\bar{\sigma}_n + 2\mathcal{B}\left(-\mathbb{C}_{22}^{e}sc + \mathbb{C}_{32}^{e}sc\right)\bar{\tau}_{nt}\right] + \mathcal{D}\left(\mathbb{C}_{22}^{e}c^2 + \mathbb{C}_{32}^{e}s^2\right)$$
$$\frac{\partial f_{E,IFF\pm}}{\partial \epsilon_{33}} = \frac{1}{2\sqrt{\mathcal{R}}}\left[2\mathcal{A}\left(\mathbb{C}_{23}^{e}c^2 + \mathbb{C}_{33}^{e}s^2\right)\bar{\sigma}_n + 2\mathcal{B}\left(-\mathbb{C}_{23}^{e}sc + \mathbb{C}_{33}^{e}sc\right)\bar{\tau}_{nt}\right] + \mathcal{D}\left(\mathbb{C}_{23}^{e}c^2 + \mathbb{C}_{33}^{e}s^2\right)$$
$$\frac{\partial f_{E,IFF\pm}}{\partial \epsilon_{12}} = \frac{1}{\sqrt{\mathcal{R}}}\left[2c\mathbb{C}_{44}^{e}\bar{\tau}_{nt}\right]$$
$$\frac{\partial f_{E,IFF\pm}}{\partial \epsilon_{13}} = \frac{1}{\sqrt{\mathcal{R}}}\left[2c\mathbb{C}_{55}^{e}\bar{\tau}_{nt}\right]$$
$$\frac{\partial f_{E,IFF\pm}}{\partial \epsilon_{23}} = \frac{1}{\sqrt{\mathcal{R}}}\left[4cs\mathcal{A}\mathbb{C}_{66}^{e}\bar{\sigma}_n + 2\left(c^2 - s^2\right)\mathcal{B}\mathbb{C}_{66}^{e}\bar{\tau_{nt}}\right] + 4cs\mathcal{D}\mathbb{C}_{66}^{e}$$

Appendix B

3D Puck Failure Criterion Subroutine

```
! SUBROUTINE HEADER
SUBROUTINE umat(stress,statev,ddsdde,sse,spd,scd,rpl,ddsddt,drplde,drpldt,&
stran,dstran,time,dtime,temp,dtemp,predef,dpred,cmname,ndi,nshr,ntens,     &
nstatv,props,nprops,coords,drot,pnewdt,celent,dfgrd0,dfgrd1,noel,npt,      &
layer,kspt,jstep,kinc)

INCLUDE 'aba_param.inc'

CHARACTER(80) :: cmname

DIMENSION stress(ntens),statev(nstatv),ddsdde(ntens,ntens),ddsddt(ntens), &
drplde(ntens),stran(ntens),dstran(ntens),time(2),predef(1),dpred(1),       &
props(nprops),coords(3),drot(3,3),dfgrd0(3,3),dfgrd1(3,3)

! USER SUBROUTINE HEADER
DIMENSION strant(ntens),streff(ntens),celast(ntens,ntens),                 &
csec(ntens,ntens),dftde(ntens),dfcde(ntens),ditde(ntens),dicde(ntens),     &
dfftde(ntens),dffcde(ntens),dm1tde(ntens),dm1cde(ntens),dm2tde(ntens),     &
dm2cde(ntens),efstfi(ntens),efstm1(ntens),efstm2(ntens)

! SUBROUTINE PARAMETER
INTEGER, PARAMETER :: dp = KIND(0.0d0)
REAL(dp), PARAMETER :: zero = 0.0d0, one = 1.0d0, two = 2.0d0, four = 4.0d0
! SUBROUTINE TEMPORARY VARIABLES
REAL(dp) :: outiff(8)

! ====================
! INITIALIZE VARIABLE
! ====================
! MECHANICAL PROPERTIES
e11 = props(1)
e22 = props(2)
e33 = e22
u12 = props(3)
u13 = u12
u23 = props(4)
g12 = props(5)
g13 = g12
g23 = (e22/(two*(one + u23))) ! assumes e22 = e33
u21 = (e22/e11) * u12
u31 = (e33/e11) * u13
u32 = (e33/e22) * u23
```

```
43
44  ! MATERIAL STRENGTH PROPERTIES
45  xt  = props(6)
46  xc  = props(7)
47  yt  = props(8)
48  yc  = props(9)
49  s12 = props(10)
50
51  ! FIBRE AND MATRIX FRACTURE ENERGY VALUES
52  gft = props(11)
53  gfc = props(12)
54  gmt = props(13)
55  gmc = props(14)
56  gfm = props(15)
57
58  ! FIBRE VOLUME FRACTION
59  ! e11/e11f substituted with volume fraction, Deuschle et al. (2003)
60  volfib = 0.591 ! Camanho et al. (2007)
61  u12f = 0.27 ! CFRP average Poisson's Ratio
62
63  ! MAGNIFICATION FACTOR FOR CFRP
64  fm = 1.1
65
66  ! INCLINATION PARAMETER FOR CFRP
67  p21c = 0.30
68  p21t = 0.25
69  p23c = 0.30
70  p23t = 0.35
71
72  ! INITIALIZE VARIABLES
73  fft    = zero ! failure calculation
74  ffc    = zero ! failure calculation
75  fift   = zero ! failure calculation
76  fifc   = zero ! failure calculation
77  affft  = zero ! activation function
78  afffc  = zero ! activation function
79  affift = zero ! activation function
80  affifc = zero ! activation function
81  dft    = statev(3)  ! damage accumulation
82  dfc    = statev(6)  ! damage accumulation
83  dm1t   = statev(9)  ! damage accumulation
84  dm2t   = statev(10) ! damage accumulation
85  dm1c   = statev(13) ! damage accumulation
86  dm2c   = statev(14) ! damage accumulation
87
88  ! FAILURE EXPOSURE THRESHOLD FROM MAIMI ET AL. (2007)
89  rfft  = MAX(one, statev(2), statev(5))
90  rffc  = MAX(one, statev(5))
91  rfift = MAX(one, statev(8), statev(12))
92  rfifc = MAX(one, statev(12))
93
94  ! =============
95  ! STRAIN TENSOR
96  ! =============
97  DO i = 1, ntens
98      strant(i) = stran(i) + dstran(i)
99  END DO
100
```

```
! =========================
! EFFECTIVE STIFFNESS MATRIX
! =========================
! INITIALIZE EFFECTIVE STIFFNESS MATRIX
DO i = 1, ntens
    DO j = 1, ntens
        celast(i,j) = zero
    END DO
END DO

! EFFECTIVE STIFFNESS MATRIX CALCULATION
delta = one - u32**2 - 2 * u31 * u13 - 2 * u31 * u32 * u13
omega = e22 * (u23 + u31 * u13) / delta
alpha = e22 * (u31 * (one + u32 - u13) - u32) / delta
beta = e11 * (one - u32 * u23) / delta - &
        e22 * (one - u21 * (u12 + two * (one + u23))) / delta - &
        four * g12

celast(1,1) = omega + two * alpha + beta + four * g12 - two * g23
celast(1,2) = omega + alpha
celast(1,3) = omega + alpha
celast(2,1) = omega + alpha
celast(2,2) = omega + two * g23
celast(2,3) = omega
celast(3,1) = omega + alpha
celast(3,2) = omega
celast(3,3) = omega + two * g23
celast(4,4) = g12
celast(5,5) = g12
celast(6,6) = g23

! ======================
! EFFECTIVE STRESS TENSOR
! ======================
DO i = 1, ntens
    streff(i) = zero
    DO j = 1, ntens
        streff(i) = streff(i) + (celast(i,j) * strant(j))
    END DO
END DO

! ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
! EXPOSURE FACTOR FOR FIBRE FAILURE CALCULATION IN TENSION
! ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
IF (streff(1) >= zero) THEN
    fft = (1/xt) * (streff(1) - (u12 - (volfib) * u12f * fm) * &
            (streff(2) + streff(3)))
    ffc = zero
    ! WEAKENING FACTOR (CORREA ET AL., 2008) - VARIABLE FOR IFF CALCULATION
    rho = ccweak(fft)

    ! DAMAGE EVOLUTION CALCULATION
    affft = fft - rfft ! activation function
    IF (affft > zero) THEN
        rfft = fft
        aft  = two * celent * xt**2  / (two * e11 * gft - celent * xt**2)
        afc  = zero
        IF (aft >= zero) THEN
```

```
159             dft = one - (EXP(aft*(one-fft))/fft)
160         ELSE
161             dft = one
162         END IF
163     END IF
164
165 ! ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
166 ! EXPOSURE FACTOR FOR FIBRE FAILURE CALCULATION IN COMPRESSION
167 ! ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
168 ELSE
169     ffc = (1/-xc) * (streff(1) - (u12 - (volfib) * u12f * fm) * &
170             (streff(2) + streff(3)))
171     fft = zero
172     ! WEAKENING FACTOR (CORREA ET AL., 2008) - VARIABLE FOR IFF CALCULATION
173     rho = ccweak(ffc)
174
175     ! DAMAGE EVOLUTION CALCULATION
176     afffc = ffc - rffc ! activation function
177     IF (afffc > zero) THEN
178         rffc = ffc
179         afc  = two * celent * xc**2  / (two * e11 * gfc - celent * xc**2)
180         aft  = zero
181         IF (afc >= zero) THEN
182             dfc = one - (EXP(afc*(one-ffc))/ffc)
183         ELSE
184             dfc = one
185         END IF
186     END IF
187 END IF
188
189 ! ------------------------------------------------------------
190 ! FRACTURE PLANE ANGLE CALCULATION FOR INTER-FIBRE FAILURE MODE
191 ! ------------------------------------------------------------
192 ! STRENGTH PARAMETER FOR TRANSVERSE DIRECTION OF THE FRACTURE PLANE
193 r23a = yc/(two*(one + p23c))
194
195 ! FAILURE PLANE ANGLE CALCULATION (SSM METHOD)
196 IF ((rfift == one).AND.(rfifc == one)) THEN
197
198     CALL ssm(p21c,p21t,p23c,p23t,yc,yt,s12,r23a,streff,outiff)
199
200     theta   = outiff(1)
201     failex  = outiff(2)
202     fxcond  = outiff(3)
203     sigman  = outiff(4)
204     taunt   = outiff(5)
205     taun1   = outiff(6)
206     pcr     = outiff(7)
207     ptr     = outiff(8)
208
209     failex = failex/rho
210
211     IF (fxcond < 0.5) THEN
212         fifc = failex
213         fift = zero
214     ELSE
215         fifc = zero
216         fift = failex
```

```
217       END IF
218
219   ! FRACTURE PLANE ANGLE WHEN IT'S ALREADY FRACTURED IN THE PREV. ITERATION
220   ELSE
221       theta = statev(15)
222       CALL cfailx(p21c,p21t,p23c,p23t,yc,yt,s12,r23a,streff, &
223           sigman,taunt,taun1,pcr,ptr,theta,failex,fxcond)
224
225       failex = failex/rho
226
227       IF (fxcond < 0.5) THEN
228           fifc = failex
229           fift = zero
230       ELSE
231           fifc = zero
232           fift = failex
233       END IF
234   END IF
235
236   ! ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
237   ! EXPOSURE FACTOR FOR INTER-FIBRE FAILURE CALCULATION IN TENSION
238   ! ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
239   IF (fxcond > 0.5) THEN
240       ! DAMAGE EVOLUTION CALCULATION
241       affift = fift - rfift ! activation function
242       IF (affift > zero) THEN
243           rfift = fift
244           am1t = two * celent * yt**2  / (two * e22 * gmt - celent * yt**2)
245           am2t = two * celent * s12**2 / (two * g12 * gfm - celent * s12**2)
246           am1c = zero
247           am2c = zero
248           IF (am1t >= zero) THEN
249               dm1t = one - (EXP(am1t*(one-fift))/fift)
250           ELSE
251               dm1t = one
252           END IF
253           IF (am2t >= zero) THEN
254               dm2t = one - (EXP(am2t*(one-fift))/fift)
255           ELSE
256               dm2t = one
257           END IF
258       END IF
259
260   ! ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
261   ! EXPOSURE FACTOR FOR INTER-FIBRE FAILURE CALCULATION IN COMPRESSION
262   ! ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
263   ELSE
264       ! DAMAGE EVOLUTION CALCULATION
265       affifc = fifc - rfifc ! activation function
266       IF (affifc > zero) THEN
267           rfifc = fifc
268           am1c = two * celent * yc**2  / (two * e22 * gmc - celent * yc**2)
269           am2c = two * celent * s12**2 / (two * g12 * gfm - celent * s12**2)
270           am1t = zero
271           am2t = zero
272           IF (am1c >= zero) THEN
273               dm1c = one - (EXP(am1c*(one-fifc))/fifc)
274           ELSE
```

```
            dm1c = one
        END IF
        IF (am2c >= zero) THEN
            dm2c = one - (EXP(am2c*(one-fifc))/fifc)
        ELSE
            dm2c = one
        END IF
    END IF
END IF

! ----------------------
! VISCOUS REGULARIZATION
! ----------------------
! VISCOSITY PARAMETER
zeta = 0.0002 ! CHEN ET AL., 2012

! FIBRE TENSION
dft = (dtime * dft/(zeta + dtime)) + (zeta/(zeta + dtime) * statev(3))

! FIBRE COMPRESSION
dfc = (dtime * dfc/(zeta + dtime)) + (zeta/(zeta + dtime) * statev(6))

! INTER-FIBRE FAILURE TENSION FOR DIRECTION 1
dm1t = (dtime * dm1t/(zeta + dtime)) + (zeta/(zeta + dtime) * statev(9))

! INTER-FIBRE FAILURE COMPRESSION FOR DIRECTION 1
dm2t = (dtime * dm2t/(zeta + dtime)) + (zeta/(zeta + dtime) * statev(10))

! INTER-FIBRE FAILURE TENSION FOR DIRECTION 2
dm1c = (dtime * dm1c/(zeta + dtime)) + (zeta/(zeta + dtime) * statev(13))

! INTER-FIBRE FAILURE COMPRESSION FOR DIRECTION 2
dm2c = (dtime * dm2c/(zeta + dtime)) + (zeta/(zeta + dtime) * statev(14))

! ======
! STATEV
! ======
statev( 1) = fft
statev( 2) = rfft
statev( 3) = dft
statev( 4) = ffc
statev( 5) = rffc
statev( 6) = dfc
statev( 7) = fift
statev( 8) = rfift
statev( 9) = dm1t
statev(10) = dm2t
statev(11) = fifc
statev(12) = rfifc
statev(13) = dm1c
statev(14) = dm2c
statev(15) = theta

! ===============
! SECANT OPERATOR
! ===============
! INITIALIZE SECANT OPERATOR
DO i = 1, ntens
```

```
333     DO j = 1, ntens
334         csec(i,j) = zero
335     END DO
336 END DO
337 ! SECANT OPERATOR CALCULATION
338 csec(1,1) = (one -  dft) * (one -  dfc) * celast(1,1)
339 csec(1,2) = (one - dm1t) * (one - dm1c) * celast(1,2)
340 csec(1,3) = (one - dm1t) * (one - dm1c) * celast(1,3)
341 csec(2,1) = (one - dm1t) * (one - dm1c) * celast(2,1)
342 csec(2,2) = (one - dm1t) * (one - dm1c) * celast(2,2)
343 csec(2,3) = (one - dm1t) * (one - dm1c) * celast(2,3)
344 csec(3,1) = (one - dm1t) * (one - dm1c) * celast(3,1)
345 csec(3,2) = (one - dm1t) * (one - dm1c) * celast(3,2)
346 csec(3,3) = (one - dm1t) * (one - dm1c) * celast(3,3)
347 csec(4,4) = (one - dm2t) * (one - dm2c) * celast(4,4)
348 csec(5,5) = (one - dm2t) * (one - dm2c) * celast(5,5)
349 csec(6,6) = (one - dm1t) * (one - dm1t) * celast(6,6)
350
351 ! ==============
352 ! STRESS STENSOR
353 ! ==============
354 DO i = 1, ntens
355     stress(i) = zero
356     DO j = 1, ntens
357         stress(i) = stress(i) + (csec(i,j) * strant(j))
358     END DO
359 END DO
360
361 ! ================
362 ! TANGENT OPERATOR
363 ! ================
364 ! ------------------------------------------------------------------
365 ! VARIATION OF THE DAMAGE VARIABLES WITH RESPECT TO THE EXPOSURE FACTOR
366 ! ------------------------------------------------------------------
367 IF (fft /= zero) THEN
368     dftdft = ((one + aft * fft) * (aft * (one - fft)))/fft**2
369 ELSE
370     dftdft = zero
371 END IF
372
373 IF (ffc /= zero) THEN
374     dfcdfc = ((one + afc * ffc) * (afc * (one - ffc)))/ffc**2
375 ELSE
376     dfcdfc = zero
377 END IF
378
379 IF (fift /= zero) THEN
380     d1tdit = ((one + am1t * fift) * (am1t * (one - fift)))/fift**2
381     d2tdit = ((one + am2t * fift) * (am2t * (one - fift)))/fift**2
382 ELSE
383     d1tdit = zero
384     d2tdit = zero
385 END IF
386
387 IF (fifc /= zero) THEN
388     d1cdic = ((one + am1c * fifc) * (am1c * (one - fifc)))/fifc**2
389     d2cdic = ((one + am2c * fifc) * (am2c * (one - fifc)))/fifc**2
390 ELSE
```

```
391       d1cdic = zero
392       d2cdic = zero
393   END IF
394
395   ! ------------------------------------------------------------
396   ! DERIVATIVES OF TANGENT OPERATOR FOR FIBRE FAILURE IN TENSION
397   ! ------------------------------------------------------------
398   dftde(1) = (1/xt) * (celast(1,1) - (u12 - (volfib) * u12f * fm) * &
399               (celast(2,1) + celast(3,1)))
400   dftde(2) = (1/xt) * (celast(1,2) - (u12 - (volfib) * u12f * fm) * &
401               (celast(2,2) + celast(3,2)))
402   dftde(3) = (1/xt) * (celast(1,3) - (u12 - (volfib) * u12f * fm) * &
403               (celast(2,3) + celast(3,3)))
404   dftde(4) = zero
405   dftde(5) = zero
406   dftde(6) = zero
407
408   ! ------------------------------------------------------------
409   ! DERIVATIVES OF TANGENT OPERATOR FOR FIBRE FAILURE IN COMPRESSION
410   ! ------------------------------------------------------------
411   dfcde(1) = (1/-xc) * (celast(1,1) - (u12 - (volfib) * u12f * fm) * &
412               (celast(2,1) + celast(3,1)))
413   dfcde(2) = (1/-xc) * (celast(1,2) - (u12 - (volfib) * u12f * fm) * &
414               (celast(2,2) + celast(3,2)))
415   dfcde(3) = (1/-xc) * (celast(1,3) - (u12 - (volfib) * u12f * fm) * &
416               (celast(2,3) + celast(3,3)))
417   dfcde(4) = zero
418   dfcde(5) = zero
419   dfcde(6) = zero
420
421   ! ------------------------------------------------------------
422   ! DERIVATIVES OF TANGENT OPERATOR FOR INTER-FIBRE FAILURE IN TENSION
423   ! ------------------------------------------------------------
424   bcap = (1/r23a)**2
425   ccap = (1/s12)**2
426
427   acap = ((1/yt) - ptr)**2
428   dcap = ptr
429   rcap = acap*sigman**2 + bcap*taunt**2 + ccap*taun1**2
430
431   IF (rcap > zero) THEN
432       ditde(1) = 1/(two * SQRT(rcap)) * (two * acap * sigman *(  &
433                   celast(2,1) * COSD(theta)**2 +                 &
434                   celast(3,1) * SIND(theta)**2)) +               &
435                  two * bcap * taunt *(-                          &
436                   celast(2,1) * SIND(theta) * COSD(theta) +      &
437                   celast(3,1) * SIND(theta) * COSD(theta)) +     &
438                   dcap *(                                        &
439                   celast(2,1) * COSD(theta)**2 +                 &
440                   celast(3,1) * SIND(theta)**2)
441       ditde(2) = 1/(two * SQRT(rcap)) * (two * acap * sigman *(  &
442                   celast(2,2) * COSD(theta)**2 +                 &
443                   celast(3,2) * SIND(theta)**2)) +               &
444                  two * bcap * taunt *(-                          &
445                   celast(2,2) * SIND(theta) * COSD(theta) +      &
446                   celast(3,2) * SIND(theta) * COSD(theta)) +     &
447                   dcap *(                                        &
448                   celast(2,2) * COSD(theta)**2 +                 &
```

```
                celast(3,2) * SIND(theta)**2)
    ditde(3) = 1/(two * SQRT(rcap)) * (two * acap * sigman *(  &
                celast(2,3) * COSD(theta)**2 +                 &
                celast(3,3) * SIND(theta)**2)) +               &
                two * bcap * taunt *(-                         &
                celast(2,3) * SIND(theta) * COSD(theta) +      &
                celast(3,3) * SIND(theta) * COSD(theta)) +     &
                dcap *(                                        &
                celast(2,3) * COSD(theta)**2 +                 &
                celast(3,3) * SIND(theta)**2)
    ditde(4) = 1/SQRT(rcap) * (2*COSD(theta) *ccap *celast(4,4) * taun1)
    ditde(5) = 1/SQRT(rcap) * (2*SIND(theta) *ccap *celast(5,5) * taun1)
    ditde(6) = 1/SQRT(rcap) * (4*COSD(theta) * SIND(theta) * &
                acap * sigman * celast(6,6) + &
                2 * (COSD(theta)**2 - SIND(theta)**2) * &
                bcap * celast(6,6) * taunt) + &
                4 * COSD(theta) * SIND(theta) * dcap * celast(6,6)
ELSE
    DO i = 1, ntens
        ditde(i) = zero
    END DO
END IF

! ---------------------------------------------------------------------
! DERIVATIVES OF TANGENT OPERATOR FOR INTER-FIBRE FAILURE IN COMPRESSION
! ---------------------------------------------------------------------
acap = pcr**2
dcap = pcr
rcap = acap*sigman**2 + bcap*taunt**2 + ccap*taun1**2

IF (rcap > zero) THEN
    dicde(1) = 1/(two * SQRT(rcap)) * (two * acap * sigman *( &
                celast(2,1) * COSD(theta)**2 +                 &
                celast(3,1) * SIND(theta)**2)) +               &
                two * bcap * taunt *(-                         &
                celast(2,1) * SIND(theta) * COSD(theta) +      &
                celast(3,1) * SIND(theta) * COSD(theta)) +     &
                dcap *(                                        &
                celast(2,1) * COSD(theta)**2 +                 &
                celast(3,1) * SIND(theta)**2)
    dicde(2) = 1/(two * SQRT(rcap)) * (two * acap * sigman *( &
                celast(2,2) * COSD(theta)**2 +                 &
                celast(3,2) * SIND(theta)**2)) +               &
                two * bcap * taunt *(-                         &
                celast(2,2) * SIND(theta) * COSD(theta) +      &
                celast(3,2) * SIND(theta) * COSD(theta)) +     &
                dcap *(                                        &
                celast(2,2) * COSD(theta)**2 +                 &
                celast(3,2) * SIND(theta)**2)
    dicde(3) = 1/(two * SQRT(rcap)) * (two * acap * sigman *( &
                celast(2,3) * COSD(theta)**2 +                 &
                celast(3,3) * SIND(theta)**2)) +               &
                two * bcap * taunt *(-                         &
                celast(2,3) * SIND(theta) * COSD(theta) +      &
                celast(3,3) * SIND(theta) * COSD(theta)) +     &
                dcap *(                                        &
                celast(2,3) * COSD(theta)**2 +                 &
                celast(3,3) * SIND(theta)**2)
```

```
    dicde(4) = 1/SQRT(rcap) * (2*COSD(theta) *ccap * celast(4,4) * taun1)
    dicde(5) = 1/SQRT(rcap) * (2*SIND(theta) *ccap * celast(5,5) * taun1)
    dicde(6) = 1/SQRT(rcap) * (4*COSD(theta) * SIND(theta) * &
                acap * sigman * celast(6,6) + &
                2 * (COSD(theta)**2 - SIND(theta)**2) * &
                bcap * celast(6,6) * taunt) + &
                4 * COSD(theta) * SIND(theta) * dcap * celast(6,6)
ELSE
    DO i = 1, ntens
        dicde(i) = zero
    END DO
END IF

! ------------------------------------------------------------------
! DERIVATIVES OF THE DAMAGE VARIABLES WITH RESPECT TO THE STRAIN TENSOR
! ------------------------------------------------------------------
DO i = 1, ntens
    dfftde(i) = dftdft * dftde(i)
    dffcde(i) = dfcdfc * dfcde(i)
    dm1tde(i) = d1tdit * ditde(i)
    dm1cde(i) = d1cdic * dicde(i)
    dm2tde(i) = d2tdit * ditde(i)
    dm2cde(i) = d2cdic * dicde(i)
END DO

! --------------------------------
! DECOUPLED EFFECTIVE STRESS TENSOR
! --------------------------------
DO i = 1, ntens
    efstfi(i) = zero
    efstm1(i) = zero
    efstm2(i) = zero
    DO j = 1, ntens
        IF ((i <= 3).AND.(j <= 3)) THEN
        IF ((i == 1).AND.(j == 1)) THEN
            efstfi(i) = efstfi(i) + (celast(i,j) * strant(j))
        ELSE
            efstm1(i) = efstm1(i) + (celast(i,j) * strant(j))
        END IF
        ELSE IF ((i == 4).AND.(j == 4)) THEN
            efstm2(i) = efstm2(i) + (celast(i,j) * strant(j))
        ELSE IF ((i == 5).AND.(j == 5)) THEN
            efstm2(i) = efstm2(i) + (celast(i,j) * strant(j))
        ELSE IF ((i == 6).AND.(j == 6)) THEN
            efstm1(i) = efstm1(i) + (celast(i,j) * strant(j))
        END IF
    END DO
END DO

! ------------------------
! TANGENT STIFFNESS MATRIX
! ------------------------
DO i = 1, ntens
    DO j = 1, ntens
        ddsdde(i,j) = csec(i,j) -                &
        (one -  dfc) * dfftde(i) * efstfi(j) - &
        (one -  dft) * dffcde(i) * efstfi(j) - &
        (one - dm1c) * dm1tde(i) * efstm1(j) - &
```

```
565           (one - dm1t) * dm1cde(i) * efstm1(j) - &
566           (one - dm2c) * dm2tde(i) * efstm2(j) - &
567           (one - dm2t) * dm2cde(i) * efstm2(j)
568       END DO
569   END DO
570
571   ! ================
572   ! ELEMENT DELETION
573   ! ================
574   IF (statev(3).ge.0.95) THEN
575       statev(16) = 0
576   END IF
577   IF (statev(6).ge.0.95) THEN
578       statev(16) = 0
579   END IF
580   IF (statev(9).ge.0.95) THEN
581       statev(16) = 0
582   END IF
583   IF (statev(10).ge.0.95) THEN
584       statev(16) = 0
585   END IF
586   IF (statev(13).ge.0.95) THEN
587       statev(16) = 0
588   END IF
589   IF (statev(14).ge.0.95) THEN
590       statev(16) = 0
591   END IF
592   END SUBROUTINE umat
593
594   ! ============================================================
595   ! FUNCTION TO CALCULATE WEAKENING FACTOR (CORREA ET AL., 2008)
596   ! ============================================================
597       FUNCTION ccweak(fftc)
598
599   IMPLICIT NONE
600
601   ! Subroutine parameter
602   INTEGER, PARAMETER :: dp = KIND(0.0d0)
603   REAL(dp), PARAMETER :: one = 1.0d0, half = 0.5d0
604   ! Function declaration
605   REAL(dp) :: ccweak
606   ! Subroutine input
607   REAL(dp), INTENT(IN) :: fftc     ! fiber failure exposure
608
609   IF ((fftc >=0.5).AND.(fftc <=1.5)) THEN
610       ccweak = sqrt(one - ((one - fftc)**2 / &
611       ((one - half) / (sqrt(one - half**2)))**2))
612   ELSE
613       ccweak = one
614   END IF
615   END FUNCTION ccweak
616
617   ! ============================================================
618   ! SUBROUTINE TO CALCULATE STRESS AT FRACTURE PLANE ANGLE FOR IFF
619   ! ============================================================
620   SUBROUTINE ccsfp(streff,theta,sigman,taunt,taun1)
621
622   IMPLICIT NONE
```

```
623
624 ! Subroutine parameter
625 INTEGER, PARAMETER :: dp = KIND(0.0d0)
626 ! Subroutine input
627 REAL(dp), INTENT(IN) :: streff(6),theta
628 ! Subroutine output
629 REAL(dp), INTENT(OUT) :: sigman,taunt,taun1
630
631 sigman = streff(2) * COSD(theta)**2 + streff(3) * SIND(theta)**2 + 2 * &
632             streff(6) * SIND(theta) * COSD(theta)
633
634 taunt = -streff(2) * SIND(theta) * COSD(theta) + streff(3) * &
635         SIND(theta) * COSD(theta) + streff(6) * &
636         (COSD(theta)**2 - SIND(theta)**2)
637
638 taun1 = streff(4) * COSD(theta) + streff(5) * SIND(theta)
639
640 END SUBROUTINE ccsfp
641
642 ! ======================================================
643 ! SUBROUTINE TO CALCULATE INCLINATION TO STRENGTH RATIO
644 ! ======================================================
645 SUBROUTINE ccfpp(taunt,taun1,p23t,p23c,p21t,p21c,yc,s12,r23a,ptr,pcr)
646
647 IMPLICIT NONE
648
649 ! Subroutine parameter
650 INTEGER, PARAMETER :: dp = KIND(0.d0)
651 REAL(dp), PARAMETER :: zero = 0.0d0
652 ! Subroutine temporary variables
653 REAL(dp) :: c2psi,s2psi
654 ! Subroutine input
655 REAL(dp), INTENT(IN) :: taunt,taun1,p23t,p23c,p21t,p21c,yc,s12,r23a
656 ! Subroutine output
657 REAL(dp), INTENT(OUT) :: ptr,pcr
658
659 IF ((taunt == zero).AND.(taun1 == zero)) THEN
660     c2psi = zero
661     s2psi = zero
662 ELSE
663     c2psi = taunt**2/(taunt**2 + taun1**2)
664     s2psi = taun1**2/(taunt**2 + taun1**2)
665 END IF
666
667 pcr = (p23c/r23a)*c2psi + (p21c/s12)*s2psi
668 ptr = (p23t/r23a)*c2psi + (p21t/s12)*s2psi
669
670 END SUBROUTINE ccfpp
671
672 ! ========================================
673 ! FUNCTION TO CALCULATE IFF IN COMPRESSION
674 ! ========================================
675 FUNCTION cciffc(sigman,taunt,taun1,r23a,s12,pcr)
676
677 IMPLICIT NONE
678
679 ! Subroutine parameter
680 INTEGER, PARAMETER :: dp = KIND(0.d0)
```

```
! Function declaration
REAL(dp) :: cciffc
! Subroutine input
REAL(dp), INTENT(IN) :: sigman,taunt,taun1,r23a,s12,pcr

cciffc = SQRT((taunt/r23a)**2 + (taun1/s12)**2 + (pcr*sigman)**2) + &
            pcr*sigman

END FUNCTION cciffc

! ====================================
! FUNCTION TO CALCULATE IFF IN TENSION
! ====================================
FUNCTION ccifft(sigman,taunt,taun1,yt,r23a,s12,ptr)

IMPLICIT NONE

! Subroutine parameter
INTEGER, PARAMETER :: dp = KIND(0.d0)
! Function declaration
REAL(dp) :: ccifft
! Subroutine input
REAL(dp), INTENT(IN) :: sigman,taunt,taun1,r23a,s12,ptr,yt

ccifft = SQRT((((1/yt) - ptr)**2) * (sigman)**2 + (taunt/r23a)**2 + &
            (taun1/s12)**2) + ptr*sigman

END FUNCTION ccifft

! ================================================
! SUBROUTINE TO CALCULATE FAILURE EXPOSURE FOR IFF
! ================================================
SUBROUTINE cfailx(p21c,p21t,p23c,p23t,yc,yt,s12,r23a,streff, &
    sigman,taunt,taun1,pcr,ptr,theta,failex,fxcond)

IMPLICIT NONE

! Subroutine parameter
INTEGER, PARAMETER :: dp = KIND(0.0d0)
REAL(dp), PARAMETER :: zero = 0.0d0, one = 1.0d0
! Function declaration
REAL(dp), EXTERNAL :: cciffc, ccifft
! Subroutine input
REAL(dp), INTENT(IN) :: streff(6),theta,p21c,p21t,p23c,p23t,yc,yt,s12,r23a
! Subroutine output
REAL(dp), INTENT(OUT) :: sigman,taunt,taun1,pcr,ptr,failex,fxcond

CALL ccsfp(streff,theta,sigman,taunt,taun1)
CALL ccfpp(taunt,taun1,p23t,p23c,p21t,p21c,yc,s12,r23a,ptr,pcr)

IF (sigman < zero) THEN
    failex = cciffc(sigman,taunt,taun1,r23a,s12,pcr)
    fxcond = zero
ELSE
    failex = ccifft(sigman,taunt,taun1,yt,r23a,s12,ptr)
    fxcond = one
END IF
END SUBROUTINE cfailx
```

```
739
740 ! ============================================================
741 ! SUBROUTINE TO CALCULATE THE FRACTURE PLANE USING SSM ALGORITHM
742 ! ============================================================
743 SUBROUTINE ssm(p21c,p21t,p23c,p23t,yc,yt,s12,r23a,streff,outiff)
744
745 IMPLICIT NONE
746
747 ! Subroutine parameter
748 INTEGER, PARAMETER :: dp = KIND(0.0d0)
749 REAL, PARAMETER :: zero = 0.0d0
750 REAL(dp), PARAMETER :: addang = 0.1 ! CHANGE THIS STEP
751 ! Subroutine temporary variables
752 INTEGER :: iarray,glbpos,locpos
753 REAL(dp) :: theta,points(1810,8)
754 REAL(dp) :: failex,fxcond,sigman,taunt,taun1,pcr,ptr
755 ! Subroutine input
756 REAL(dp), INTENT(IN) :: p21c,p21t,p23c,p23t,yc,yt,s12,r23a, streff(6)
757 ! Subroutine output
758 REAL(dp), INTENT(OUT) :: outiff(8)
759
760 IF ((streff(2) /= zero).AND.(streff(3) /= zero).AND.(streff(4) /= zero).AND. &
761     (streff(5) /= zero).AND.(streff(6) /= zero)) THEN
762
763     ! Initialize variables
764     locpos = 1
765     DO theta = -90,90,addang
766         CALL cfailx(p21c,p21t,p23c,p23t,yc,yt,s12,r23a,streff, &
767             sigman,taunt,taun1,pcr,ptr,theta,failex,fxcond)
768
769         points(locpos,1) = theta
770         points(locpos,2) = failex
771         points(locpos,3) = fxcond
772         points(locpos,4) = sigman
773         points(locpos,5) = taunt
774         points(locpos,6) = taun1
775         points(locpos,7) = pcr
776         points(locpos,8) = ptr
777
778         locpos = locpos + 1
779     END DO
780
781     ! Find global max point
782     glbpos  = MAXLOC(points(:,2), dim = 1)
783     DO iarray = 1,8
784         outiff(iarray) = points(glbpos,iarray)
785     END DO
786 ELSE
787     outiff(1:8) = zero
788 END IF
789 END SUBROUTINE ssm
```

Appendix C

Improved Fast FPOA Search Algorithm

In order to utilises improved fast FPOA search algorithm, the UMAT subroutine on the Appendix B has to be modified. This modification involves updating the calling syntax function of the search algorithm function and replace the search algorithm function with the improved fast FPOA search algorithm. function The following code snippets shows the original and modified lines of the UMAT subroutine.

Modify the calling syntax of the search algorithm function

Original UMAT subroutine lines:

```
197 ! FAILURE PLANE ANGLE CALCULATION (SSM METHOD)
198 IF ((rfift == one).AND.(rfifc == one)) THEN
199
200     CALL ssm(p21c,p21t,p23c,p23t,yc,yt,s12,r23a,streff,outiff)
```

Modified UMAT subroutine lines:

```
197 ! FAILURE PLANE ANGLE CALCULATION (ISRBM METHOD)
198 IF ((rfift == one).AND.(rfifc == one)) THEN
199     stangl = 10 ! inclination angle with 10 degree interval
200     CALL isrbm(p21c,p21t,p23c,p23t,yc,yt,s12,r23a,streff,stangl,outiff)
```

Replace the search algorithm function

Original UMAT subroutine lines:

```
740 ! ============================================================
741 ! SUBROUTINE TO CALCULATE THE FRACTURE PLANE USING SSM ALGORITHM
742 ! ============================================================
743 SUBROUTINE ssm(p21c,p21t,p23c,p23t,yc,yt,s12,r23a,streff,outiff)
744
745 IMPLICIT NONE
746
747 ! Subroutine parameter
748 INTEGER, PARAMETER :: dp = KIND(0.0d0)
749 REAL, PARAMETER :: zero = 0.0d0
750 REAL(dp), PARAMETER :: addang = 0.1 ! CHANGE THIS STEP
751 ! Subroutine temporary variables
752 INTEGER :: iarray,glbpos,locpos
```

```
753 REAL(dp) :: theta,points(1810,8)
754 REAL(dp) :: failex,fxcond,sigman,taunt,taun1,pcr,ptr
755 ! Subroutine input
756 REAL(dp), INTENT(IN) :: p21c,p21t,p23c,p23t,yc,yt,s12,r23a, streff(6)
757 ! Subroutine output
758 REAL(dp), INTENT(OUT) :: outiff(8)
759
760 IF ((streff(2) /= zero).AND.(streff(3) /= zero).AND.(streff(4) /= zero).AND. &
761     (streff(5) /= zero).AND.(streff(6) /= zero)) THEN
762
763     ! Initialize variables
764     locpos = 1
765     DO theta = -90,90,addang
766         CALL cfailx(p21c,p21t,p23c,p23t,yc,yt,s12,r23a,streff, &
767             sigman,taunt,taun1,pcr,ptr,theta,failex,fxcond)
768
769         points(locpos,1) = theta
770         points(locpos,2) = failex
771         points(locpos,3) = fxcond
772         points(locpos,4) = sigman
773         points(locpos,5) = taunt
774         points(locpos,6) = taun1
775         points(locpos,7) = pcr
776         points(locpos,8) = ptr
777
778         locpos = locpos + 1
779     END DO
780
781     ! Find global max point
782     glbpos  = MAXLOC(points(:,2), dim = 1)
783     DO iarray = 1,8
784         outiff(iarray) = points(glbpos,iarray)
785     END DO
786 ELSE
787     outiff(1:8) = zero
788 END IF
789 END SUBROUTINE ssm
```

Modified UMAT subroutine lines:

```
740 ! ============================================================
741 ! SUBROUTINE TO CALCULATE FRACTURE ANGLE USING BRENT'S METHOD
742 ! ============================================================
743 SUBROUTINE cbrent(p21c,p21t,p23c,p23t,yc,yt,s12,r23a,streff, &
744     sangle,eangle,outiff)
745
746 IMPLICIT NONE
747
748 ! Subroutine parameter
749 INTEGER, PARAMETER :: dp = KIND(0.0d0)
750 REAL, PARAMETER :: zero = 0.0d0
751 ! Subroutine temporary variable
752 REAL(dp) :: failex,fxcond,sigman,taunt,taun1,pcr,ptr
753 REAL(dp) :: c,d,e,epsi,fu(8),fv(8),fw(8),fx(8),&
754             m,p,q,r,s,sa,sb,t,t2,tol,u,v,w,x
755 ! Subroutine input
756 REAL(dp), INTENT(IN) :: p21c,p21t,p23c,p23t,yc,yt,s12,r23a, streff(6)
757 REAL(dp), INTENT(IN) :: sangle, eangle
758 ! Subroutine output
```

```
759 REAL(dp), INTENT(OUT) :: outiff(8)
760
761 ! C is the square of the inverse of the golden ratio.
762 c = 0.5d0 * (3.0d0 - SQRT(5.0d0))
763
764 epsi    = 1.0e-4
765 t       = 1.0e-4
766
767 sa = sangle
768 sb = eangle
769 x = sa + c * ( eangle - sangle )
770 w = x
771 v = w
772 e = 0.0d0
773
774 CALL cfailx(p21c,p21t,p23c,p23t,yc,yt,s12,r23a,streff, &
775     sigman,taunt,taun1,pcr,ptr,x,failex,fxcond)
776
777 fx(1) = x
778 fx(2) = failex
779 fx(3) = fxcond
780 fx(4) = sigman
781 fx(5) = taunt
782 fx(6) = taun1
783 fx(7) = pcr
784 fx(8) = ptr
785
786 fw = fx
787 fv = fw
788
789 DO
790     m = 0.5d0 * (sa + sb)
791     tol = 1.0d0
792     t2 = 2.0d0 * tol
793
794     ! Check the stopping criterion.
795     IF (ABS(x - m) <= t2 - 0.5d0 * (sb - sa)) THEN
796         EXIT
797     END if
798
799     ! Fit a parabola.
800     r = 0.0d0
801     q = r
802     p = q
803
804     IF (tol < ABS(e)) THEN
805         r = (x - w) * (fx(2) - fv(2))
806         q = (x - v) * (fx(2) - fw(2))
807         p = (x - v) * q - (x - w) * r
808         q = 2.0d0 * (q - r)
809
810         IF ( 0.0d0 < q ) THEN
811         p = - p
812         END IF
813
814         q = ABS(q)
815
816         r = e
```

```
817         e = d
818
819     END IF
820
821     IF (ABS(p) < ABS(0.5d0 * q * r ).AND. &
822         q * (sa - x) < p .AND. &
823         p < q * (sb - x)) THEN
824
825     ! Take the parabolic interpolation step.
826         d = p / q
827         u = x + d
828
829         ! F must not be evaluated too close to A or B.
830         IF ((u - sa) < t2 .OR. (sb - u) < t2) THEN
831             IF (x < m) THEN
832             d = tol
833             ELSE
834             d = - tol
835             END IF
836         END IF
837
838     ! A golden-section step.
839     ELSE
840         IF (x < m) THEN
841             e = sb - x
842         ELSE
843             e = sa - x
844         END IF
845         d = c * e
846     END IF
847
848     ! F must not be evaluated too close to X.
849     IF (tol <= ABS(d)) THEN
850         u = x + d
851     ELSE IF ( 0.0D+00 < d ) THEN
852         u = x + tol
853     ELSE
854         u = x - tol
855     END IF
856
857     CALL cfailx(p21c,p21t,p23c,p23t,yc,yt,s12,r23a,streff, &
858             sigman,taunt,taun1,pcr,ptr,u,failex,fxcond)
859
860     fu(1) = u
861     fu(2) = failex
862     fu(3) = fxcond
863     fu(4) = sigman
864     fu(5) = taunt
865     fu(6) = taun1
866     fu(7) = pcr
867     fu(8) = ptr
868
869     ! Update A, B, V, W, and X.
870     IF (fu(2) >= fx(2)) THEN
871         IF (u < x ) THEN
872             sb = x
873         ELSE
874             sa = x
```

```
875             END IF
876
877             v = w
878             fv = fw
879             w = x
880             fw = fx
881             x = u
882             fx = fu
883         ELSE
884             IF (u < x) THEN
885                 sa = u
886             ELSE
887                 sb = u
888             END IF
889
890             IF (fu(2) >= fw(2) .OR. w == x ) THEN
891                 v = w
892                 fv = fw
893                 w = u
894                 fw = fu
895             ELSE IF (fu(2) >= fv(2) .OR. v == x .OR. v == w ) THEN
896                 v = u
897                 fv = fu
898             END IF
899         END IF
900     END DO
901     outiff = fx
902     END SUBROUTINE cbrent
903
904     ! ================================================================
905     ! SUBROUTINE TO CALCULATE THE FRACTURE PLANE USING ISRBM ALGORITHM
906     ! ================================================================
907     SUBROUTINE isrbm(p21c,p21t,p23c,p23t,yc,yt,s12,r23a,streff,stangl,outiff)
908
909     IMPLICIT NONE
910
911     ! Subroutine parameter
912     INTEGER, PARAMETER :: dp = KIND(0.0d0)
913     REAL, PARAMETER :: zero = 0.0d0
914     ! Subroutine temporary variables
915     REAL(dp) :: theta, points(50,8)
916     REAL(dp) :: failex,fxcond,sigman,taunt,taun1,pcr,ptr
917     REAL(dp) :: eangle,sangle,epsi
918     INTEGER :: locpos,suppos,glbpos
919     ! Subroutine input
920     REAL(dp), INTENT(IN) :: p21c,p21t,p23c,p23t,yc,yt,s12,r23a,streff(6)
921     REAL(dp), INTENT(IN) :: stangl
922     ! Subroutine output
923     REAL(dp), INTENT(OUT) :: outiff(8)
924
925     ! Initialize variables
926     points(:,:) = 0.0d0
927     epsi        = 5.0e-3
928     theta       = -90
929
930     ! Calculate support points
931     locpos = 1 ! Initialize the iteration of the support points position
932     DO WHILE (theta <= 90 + stangl)
```

```
933         points(locpos,1) = theta
934         IF (theta >= 90) THEN
935             IF (theta <= (90 + epsi)) THEN
936                 points(locpos,2:8) = points(1,2:8)
937             ELSE
938                 points(locpos,2:8) = points(2,2:8)
939             END IF
940         ELSE
941             CALL cfailx(p21c,p21t,p23c,p23t,yc,yt,s12,r23a,streff, &
942                 sigman,taunt,taun1,pcr,ptr,theta,failex,fxcond)
943 
944             points(locpos,2) = failex
945             points(locpos,3) = fxcond
946             points(locpos,4) = sigman
947             points(locpos,5) = taunt
948             points(locpos,6) = taun1
949             points(locpos,7) = pcr
950             points(locpos,8) = ptr
951         END IF
952         locpos = locpos + 1
953         theta = theta + stangl
954     END DO
955 
956     DO suppos = 2, (locpos - 2)
957         IF((points((suppos-1),2) <= points((suppos),2)).AND. &
958             (points((suppos),2) >= points((suppos+1),2))) THEN
959             ! Use Brent's Method
960             CALL cbrent(p21c,p21t,p23c,p23t,yc,yt,s12,r23a,streff, &
961                 points((suppos-1),1),points((suppos+1),1),outiff)
962 
963             points(locpos,:) = outiff
964 
965             locpos = locpos + 1
966         END IF
967     END DO
968 
969     ! Find global max point
970     glbpos  = MAXLOC(points(:,2), dim = 1)
971     outiff  = points(glbpos,:)
972 
973     IF(outiff(1) >= 90) THEN
974         IF(outiff(1) <= (90 + epsi)) THEN
975             outiff = points(1,:)
976         ELSE
977             outiff(1) = outiff(1) - 180
978             outiff(6) = -1 * outiff(6)
979         ENDIF
980     ENDIF
981     END SUBROUTINE isrbm
```

Appendix D

3D Puck Failure Criterion within XFEM Framework Subroutine

The following non-standalone UDMGINI subroutine incorporates the 3D Puck failure criterion within the XFEM framework. This subroutine has to be combined with the UMAT subroutine in Appendix C to avoid error.

```
! SUBROUTINE HEADER
SUBROUTINE udmgini(findex,nfindex,fnormal,ndi,nshr,ntens,props, &
nprops,statev,nstatv,stress,strain,strainee,lxfem,time, &
dtime,temp,dtemp,predef,dpred,nfield,coords,noel,npt,layer, &
kspt,kstep,kinc,kdircyc,kcyclelcf,timecyc,sse,spd,scd,svd, &
smd,jmac,jmatyp,matlayo,laccfla,celent,drot,ori)

INCLUDE 'aba_param.inc'

DIMENSION findex(nfindex),fnormal(ndi,nfindex),coords(*), &
stress(ntens),strain(ntens),strainee(ntens),props(nprops), &
statev(nstatv),predef(nfield),dpred(nfield),time(2),jmac(*), &
jmatyp(*),drot(3,3),ori(3,3),lxfem(3)

! SUBROUTINE PARAMETER
INTEGER, PARAMETER :: dp = KIND(0.0d0)
REAL(dp), PARAMETER :: zero = 0.0d0, one = 1.0d0, two = 2.0d0, four = 4.0d0
! SUBROUTINE TEMPORARY VARIABLES
REAL(dp) :: outiff(8)

u12 = props(1)
xt  = props(2)
xc  = props(3)
yt  = props(4)
yc  = props(5)
s12 = props(6)

! FIBRE VOLUME FRACTION
! e11/e11f substituted with volume fraction, Deuschle et al. (2003)
volfib = 0.591 ! Camanho et al. (2007)
u12f = 0.27 ! CFRP average Poisson's Ratio

! MAGNIFICATION FACTOR FOR CFRP
fm = 1.1

```

```
36 ! INCLINATION PARAMETER FOR CFRP
37 p21c = 0.30
38 p21t = 0.25
39 p23c = 0.30
40 p23t = 0.35
41
42 ! ========================
43 ! FAILURE MODE CALCULATION
44 ! ========================
45 ! ~~~~~~~~~~~~~
46 ! FIBRE FAILURE
47 ! ~~~~~~~~~~~~~
48 ! FAILURE INDEX CALCULATION
49 IF (stress(1) >= zero) THEN
50     findex(1) = (1/xt) * (stress(1) - (u12 - (volfib) * u12f * fm) *
51                 (stress(2) + stress(3)))
52     findex(2) = zero
53     rho = ccweak(findex(1))
54 ELSE
55     findex(1) = zero
56     findex(2) = (1/-xc) * (stress(1) - (u12 - (volfib) * u12f * fm) *
57                 (stress(2) + stress(3)))
58     rho = ccweak(findex(2))
59 END IF
60
61 ! ~~~~~~~~~~~~~~~~~~~
62 ! INTER-FIBRE FAILURE
63 ! ~~~~~~~~~~~~~~~~~~~
64 ! STRENGTH PARAMETER FOR TRANSVERSE DIRECTION OF THE FRACTURE PLANE
65 r23a = yc/(two*(one + p23c))
66
67 stangl = 10 ! iteration angle
68
69 CALL isrbm(p21c,p21t,p23c,p23t,yc,yt,s12,r23a,stress,stangl,outiff)
70
71 theta   = outiff(1)
72 failex  = outiff(2)
73 fxcond  = outiff(3)
74 sigman  = outiff(4)
75 taunt   = outiff(5)
76 taun1   = outiff(6)
77 pcr     = outiff(7)
78 ptr     = outiff(8)
79
80 ! FAILURE INDEX CALCULATION
81 failex = failex/rho
82 IF (fxcond < 0.5) THEN
83     findex(3) = failex
84     findex(4) = zero
85 ELSE
86     findex(3) = zero
87     findex(4) = failex
88 END IF
89
90 ! ======================
91 ! FAILURE MODE DIRECTION
92 ! ======================
93 IF ((findex(1) >= findex(2)).AND. &
```

```
94      (findex(1) >= findex(3)).AND. &
95      (findex(1) >= findex(4))) THEN
96      ! ~~~~~~~~~~~~~~~
97      ! FAILURE MODE 1
98      ! ~~~~~~~~~~~~~~~
99      DO i = 1, ndi
100         fnormal(i,1) = ori(i,1)
101     END DO
102 ELSEIF ((findex(2) >= findex(1)).AND. &
103         (findex(2) >= findex(3)).AND. &
104         (findex(2) >= findex(4))) THEN
105     ! ~~~~~~~~~~~~~~~
106     ! FAILURE MODE 2
107     ! ~~~~~~~~~~~~~~~
108     DO i = 1, ndi
109         fnormal(i,2) = ori(i,1)
110     END DO
111 ELSEIF ((findex(3) >= findex(1)).AND. &
112         (findex(3) >= findex(2)).AND. &
113         (findex(3) >= findex(4))) THEN
114     ! ~~~~~~~~~~~~~~~
115     ! FAILURE MODE 3
116     ! ~~~~~~~~~~~~~~~
117     DO i = 1, ndi
118         DO j = 1, ndi
119             fnormal(i,3) = fnormal(i,3) + ori(i,j)
120         END DO
121         IF (i == 1) THEN
122             fnormal(i,3) = fnormal(i,3) * zero
123         ELSE IF (i == 2) THEN
124             fnormal(i,3) = fnormal(i,3) * COSD(theta)
125         ELSE
126             fnormal(i,3) = fnormal(i,3) * SIND(theta)
127         END IF
128     END DO
129 ELSE
130     ! ~~~~~~~~~~~~~~~
131     ! FAILURE MODE 4
132     ! ~~~~~~~~~~~~~~~
133     DO i = 1, ndi
134         DO j = 1, ndi
135             fnormal(i,4) = fnormal(i,4) + ori(i,j)
136         END DO
137         IF (i == 1) THEN
138             fnormal(i,4) = fnormal(i,4) * zero
139         ELSE IF (i == 2) THEN
140             fnormal(i,4) = fnormal(i,4) * COSD(theta)
141         ELSE
142             fnormal(i,4) = fnormal(i,4) * SIND(theta)
143         END IF
144     END DO
145 END IF
146
147 RETURN
148 END SUBROUTINE udmgini
```

Bibliography

[1] Joel Galos. Thin-ply composite laminates: a review. *Composite Structures*, 236:111920, 2020.

[2] Justin Hale. Boeing 787 from the ground up. *Aero*, 4(24):7, 2006.

[3] M Kinsley-Jones. Airbus's A350 vision takes shape—Flight takes an in-depth look at the new twinjet. *Flight International*, 12, 2006.

[4] Arthur Roderick Collar. The first fifty years of aeroelasticity. *Aerospace (Royal Aeronautical Society Journal)*, 5(2):12–20, 1978.

[5] N Wirawan, N A Abdullah, Mahesa Akbar, and J L Curiel-Sosa. Analysis on cracked commuter aircraft wing under dynamic cruise load by means of XFEM. *Journal of Physics: Conference Series*, 1106(1):012014, October 2018. DOI: 10.1088/1742-6596/1106/1/012014. URL: https://dx.doi.org/10.1088/1742-6596/1106/1/012014.

[6] Ted Belytschko and Tom Black. Elastic crack growth in finite elements with minimal remeshing. *International journal for numerical methods in engineering*, 45(5):601–620, 1999.

[7] Nicolas Moës, John Dolbow, and Ted Belytschko. A finite element method for crack growth without remeshing. *International journal for numerical methods in engineering*, 46(1):131–150, 1999.

[8] Nur Azam Abdullah, Mahesa Akbar, Nanda Wirawan, and Jose Luis Curiel-Sosa. Structural integrity assessment on cracked composites interaction with aeroelastic constraint by means of XFEM. *Composite Structures*, 229:111414, 2019.

[9] Hanif S Hoseini and Dewey H Hodges. Aeroelastic stability analysis of damaged high-aspect-ratio composite wings. *Journal of Aircraft*, 56(5):1794–1808, 2019.

[10] Alfred Puck and Helmut Schürmann. Failure analysis of FRP laminates by means of physically based phenomenological models. *Composites science and technology*, 62(12-13):1633–1662, 2002.

[11] A Puck, J Kopp, and M Knops. Guidelines for the determination of the parameters in Puck's action plane strength criterion. *Composites Science and Technology*, 62(3):371–378, 2002.

[12] Raymond L Bisplinghoff, Holt Ashley, and Robert L Halfman. *Aeroelasticity*. Courier Corporation, 2013.

[13] Jan Robert Wright and Jonathan Edward Cooper. *Introduction to aircraft aeroelasticity and loads*. John Wiley & Sons, 2008.

[14] Fighter Jets World. The story of a lucky F-117A pilot who survived Nighthawk disintegration at the Chesapeake Air Show. 2019. URL: https://fighterjetsworld.com/air/the-story-of-a-lucky-f-117a-pilot-who-survived-nighthawk-disintegration-at-the-chesapeake-air-show/13994/.

[15] Matthew L. Wald. Old problem is pinpointed in the crash of fighter. 1997. URL: https://www.nytimes.com/1997/09/16/us/old-problem-is-pinpointed-in-the-crash-of-fighter.html.

[16] National Aeronautics and Space Administration. NASA releases Helios prototype aircraft mishap report. 2004. URL: https://www.nasa.gov/news-release/nasa-dryden-flight-research-center-news-room-news-releases-nasa-releases-helios-prototype-aircraft-mishap-report/.

[17] National Transportation Safety Board. Pilot/Race 177, The Galloping Ghost, North American P-51D, N79111 (NTSB/AAB-12/01 PB2012-102899). Technical report, 2012. URL: https://www.ntsb.gov/investigations/AccidentReports/Reports/AAB1201.pdf.

[18] John M Brown, Marla E Perez-Davis, Stephen D Ishmael, and Geary C Tiffany. Investigation of the Helios Prototype Aircraft Mishap. Technical report, National Aeronautics and Space Administration, 2004. URL: https://www.nasa.gov/pdf/64317main_helios.pdf.

[19] E Viola, L Federici, and L Nobile. Detection of crack location using cracked beam element method for structural analysis. *Theoretical and Applied Fracture Mechanics*, 36(1):23–35, 2001.

[20] Erasmo Viola and Alessandro Marzani. Crack effect on dynamic stability of beams under conservative and nonconservative forces. *Engineering Fracture Mechanics*, 71(4-6):699–718, 2004.

[21] Erasmo Viola, Alessandro Marzani, and Nicholas Fantuzzi. Interaction effect of cracks on flutter and divergence instabilities of cracked beams under subtangential forces. *Engineering Fracture Mechanics*, 151:109–129, 2016.

[22] Wen-Hwa Chen and Heng-Chih Lin. Flutter analysis of thin cracked panels using the finite element method. *AIAA journal*, 23(5):795–801, 1985.

[23] Kuo-Jiun Lin, Pong-Jeu Lu, and Jiann-Quo Tarn. Flutter analysis of anisotropic panels with patched cracks. *Journal of aircraft*, 28(12):899–907, 1991.

[24] Thomas W Strganac and Young Ik Kim. Aeroelastic behavior of composite plates subject to damage growth. *Journal of aircraft*, 33(1):68–73, 1996.

[25] RMV Pidaparti. Free vibration and flutter of damaged composite panels. *Composite structures*, 38(1-4):477–481, 1997.

[26] RMV Pidaparti and CC Chang. Finite element supersonic flutter analysis of skewed and cracked composite panels. *Computers & structures*, 69(2):265–270, 1998.

[27] Kaihong Wang, Daniel J Inman, and Charles R Farrar. Crack-induced changes in divergence and flutter of cantilevered composite panels. *Structural Health Monitoring*, 4(4):377–392, 2005.

[28] Georgia Georgiou, Abdul Manan, and Jonathan E Cooper. Modeling composite wing aeroelastic behavior with uncertain damage severity and material properties. *Mechanical Systems and Signal Processing*, 32:32–43, 2012.

[29] Nur Azam Abdullah, Jose Luis Curiel-Sosa, and Mahesa Akbar. Aeroelastic assessment of cracked composite plate by means of fully coupled finite element and Doublet Lattice Method. *Composite structures*, 202:151–161, 2018.

[30] Jayant Prakash Varun, Pritam Mondal, and Prashanta K Mahato. Enhancement of aeroelastic performance of a smart delaminated composite plate under hygrothermal environment. *Composite Structures*, 292:115662, 2022.

[31] Jens M Melenk and Ivo Babuška. The partition of unity finite element method: basic theory and applications. *Computer methods in applied mechanics and engineering*, 139(1-4):289–314, 1996.

[32] Carlos Armando Duarte and J. Oden. Hp clouds - A Meshless Method to Solve Boundary-Value Problems. Technical report, TICAM, 1995.

[33] M Fleming, YA Chu, Brian Moran, and T Belytschko. Enriched element-free galerkin methods for crack tip fields. *International journal for numerical methods in engineering*, 40(8):1483–1504, 1997.

[34] N Sukumar, N Moës, B Moran, and T Belytschko. Extended finite element method for three-dimensional crack modelling. *International journal for numerical methods in engineering*, 48(11):1549–1570, 2000.

[35] N Sukumar, ZY Huang, J-H Prévost, and Z Suo. Partition of unity enrichment for bimaterial interface cracks. *International journal for numerical methods in engineering*, 59(8):1075–1102, 2004.

[36] E Giner, N Sukumar, JE Tarancón, and FJ Fuenmayor. An abaqus implementation of the extended finite element method. *Engineering fracture mechanics*, 76(3):347–368, 2009.

[37] Emer M Feerick, Xiangyi Cheryl Liu, and Patrick McGarry. Anisotropic mode-dependent damage of cortical bone using the extended finite element method (XFEM). *Journal of the mechanical behavior of biomedical materials*, 20:77–89, 2013.

[38] Björn van Dongen, Alexander van Oostrum, and Dimitrios Zarouchas. A blended continuum damage and fracture mechanics method for progressive damage analysis of composite structures using xfem. *Composite Structures*, 184:512–522, 2018.

[39] ST Pinho, Roohoolamin Darvizeh, P Robinson, C Schuecker, and PP Camanho. Material and structural response of polymer-matrix fibre-reinforced composites. *Journal of Composite Materials*, 46(19-20):2313–2341, 2012.

[40] Stephen W Tsai and Edward M Wu. A general theory of strength for anisotropic materials. *Journal of composite materials*, 5(1):58–80, 1971.

[41] Stephen W Tsai. *Strength characteristics of composite materials*. National Aeronautics and Space Administration, 1965.

[42] Zvi Hashin. Failure criteria for unidirectional fiber composites, 1980.

[43] Robert E Rowlands. Strength(failure) theories and their experimental correlation. *Elsevier Science Publishers B. V., Handbook of Composites.*, 3:71–125, 1985.

[44] LJ Hart-Smith. Predictions of the original and truncated maximum-strain failure models for certain fibrous composite laminates. *Composites Science and Technology*, 58(7):1151–1178, 1998.

[45] A Rotem. Prediction of laminate failure with the Rotem failure criterion. *Composites Science and Technology*, 58(7):1083–1094, 1998.

[46] RM Christensen. Stress based yield/failure criteria for fiber composites. *International journal of solids and structures*, 34(5):529–543, 1997.

[47] VD Azzi and Stephen W Tsai. Anisotropic strength of composites: investigation aimed at developing a theory applicable to laminated as well as unidirectional composites, employing simple material properties derived from unidirectional specimens alone. *Experimental mechanics*, 5:283–288, 1965.

[48] Oscar Hoffman. The brittle strength of orthotropic materials. *Journal of Composite materials*, 1(2):200–206, 1967.

[49] Christos Chamis. *Failure criteria for filamentary composites*. National Aeronautics and Space Administration, 1969.

[50] Zvi Hashin and Assa Rotem. A fatigue failure criterion for fiber reinforced materials. *Journal of composite materials*, 7(4):448–464, 1973.

[51] Zvi Hashin. Fatigue failure criteria for unidirectional fiber composites. *Journal of applied mechanics*, 47(4):329–334, 1980.

[52] MJ Hinton, A Sam Kaddour, and Peter D Soden. *Failure criteria in fibre reinforced polymer composites: the world-wide failure exercise*. Elsevier, 2004.

[53] Lasar Kachanov. *Introduction to continuum damage mechanics*, volume 10. Martinus Nijhoff Publishers, 1986.

[54] ALJTR Matzenmiller, Jacob Lubliner, and Robert L Taylor. A constitutive model for anisotropic damage in fiber-composites. *Mechanics of materials*, 20(2):125–152, 1995.

[55] JC Simo and JW Ju. Strain-and stress-based continuum damage models—I. formulation. *International journal of solids and structures*, 23(7):821–840, 1987.

[56] JC Simo and JW Ju. Strain-and stress-based continuum damage models—II. computational aspects. *International journal of solids and structures*, 23(7):841–869, 1987.

[57] W Wagner and CJEFM Balzani. Prediction of the postbuckling response of composite airframe panels including ply failure. *Engineering Fracture Mechanics*, 77(18):3648–3657, 2010.

[58] M Vogler, R Rolfes, and PP Camanho. Modeling the inelastic deformation and fracture of polymer composites–Part I: Plasticity model. *Mechanics of Materials*, 59:50–64, 2013.

[59] H Matthias Deuschle and Bernd-H Kröplin. Finite element implementation of Puck's failure theory for fibre-reinforced composites under three-dimensional stress. *Journal of composite materials*, 46(19-20):2485–2513, 2012.

[60] H Matthias Deuschle and Alfred Puck. Application of the puck failure theory for fibre-reinforced composites under three-dimensional stress: Comparison with experimental results. *Journal of Composite Materials*, 47(6-7):827–846, 2013.

[61] Burton Paul. A modification of the Coulomb-Mohr theory of fracture. *Journal of Applied Mechanics*, 28:259–268, 2, 1961.

[62] E Correa, V Mantic, and F París. Numerical characterisation of the fibre–matrix interface crack growth in composites under transverse compression. *Engineering Fracture Mechanics*, 75(14):4085–4103, 2008.

[63] E Correa, V Mantic, and F París. A micromechanical view of inter-fibre failure of composite materials under compression transverse to the fibres. *Composites Science and Technology*, 68(9):2010–2021, 2008.

[64] Jose Reinoso, Giuseppe Catalanotti, Antonio Blázquez, Pedro Areias, Pedro P Camanho, and Frederico París. A consistent anisotropic damage model for laminated fiber-reinforced composites using the 3D-version of the Puck failure criterion. *International Journal of Solids and Structures*, 126:37–53, 2017.

[65] BG Falzon and Paola Apruzzese. Numerical analysis of intralaminar failure mechanisms in composite structures. Part I: FE implementation. *Composite Structures*, 93(2):1039–1046, 2011.

[66] BG Falzon and Paola Apruzzese. Numerical analysis of intralaminar failure mechanisms in composite structures. Part II: Applications. *Composite Structures*, 93(2):1047–1053, 2011.

[67] Ireneusz Lapczyk and Juan A Hurtado. Progressive damage modeling in fiber-reinforced materials. *Composites Part A: Applied science and manufacturing*, 38(11):2333–2341, 2007.

[68] Elias N Eliopoulos, Vassilis Kostopoulos, and Theodore P Philippidis. A three-dimensional progressive damage FE model for gfrp composites under monotonic loading. *Composites Science and Technology*, 123:79–91, 2016.

[69] P Maimí, Pedro Ponces Camanho, JA Mayugo, and CG Dávila. A continuum damage model for composite laminates: Part I–Constitutive model. *Mechanics of materials*, 39(10):897–908, 2007.

[70] P Maimí, Pedro Ponces Camanho, JA Mayugo, and CG Dávila. A continuum damage model for composite laminates: Part II–Computational implementation and validation. *Mechanics of materials*, 39(10):909–919, 2007.

[71] G Duvaut and JL Lions. Lions, inequalities in mechanics and physics, 1976.

[72] JF Chen, EV Morozov, and K Shankar. A combined elastoplastic damage model for progressive failure analysis of composite materials and structures. *Composite Structures*, 94(12):3478–3489, 2012.

[73] YJ Lee, CH Lee, and WS Fu. Study on the compressive strength of laminated composite with through-the-width delamination. *Composite Structures*, 41(3-4):229–241, 1998.

[74] ST Pinho. *Modelling failure of laminated composites using physically-based failure models*. PhD thesis, Department of Aeronautics, Imperial College London, UK, 2005.

[75] ST Pinho, L Iannucci, and P Robinson. Physically-based failure models and criteria for laminated fibre-reinforced composites with emphasis on fibre kinking: part i: development. *Composites Part A: Applied Science and Manufacturing*, 37(1):63–73, 2006.

[76] Oliver Völkerink, Enno Petersen, Josef Koord, and Christian Hühne. A pragmatic approach for a 3D material model considering elasto-plastic behaviour, damage initiation by Puck or Cuntze and progressive failure of fibre-reinforced plastics. *Computers & Structures*, 236:106280, 2020.

[77] Ren-Horng Maa and Jung-Ho Cheng. A CDM-based failure model for predicting strength of notched composite laminates. *Composites Part B: Engineering*, 33(6):479–489, 2002.

[78] Ana M Girão Coelho, J Toby Mottram, and Kent A Harries. Finite element guidelines for simulation of fibre-tension dominated failures in composite materials validated by case studies. *Composite Structures*, 126:299–313, 2015.

[79] Alfred Puck. *Festigkeitsanalyse von Faser-Matrix-Laminaten: Modelle für die Praxis*. Hanser, 1996.

[80] Verein Deutscher Ingeniere. *Development of fibre-reinforced plastics components – analysis*. Beuth Verlag GmbH, 2006.

[81] J Wiegand, N Petrinic, and B Elliott. An algorithm for determination of the fracture angle for the three-dimensional Puck matrix failure criterion for UD composites. *Composites Science and Technology*, 68(12):2511–2517, 2008.

[82] Jack Kiefer. Sequential minimax search for a maximum. *Proceedings of the American mathematical society*, 4(3):502–506, 1953.

[83] P Jarratt. An iterative method for locating turning points. *The Computer Journal*, 10(1):82–84, 1967.

[84] FJ Schirmaier, J Weiland, Luise Kärger, and Frank Henning. A new efficient and reliable algorithm to determine the fracture angle for Puck's 3D matrix failure criterion for UD composites. *Composites science and technology*, 100:19–25, 2014.

[85] M Rezasefat, Daniel Badel Torres, Alvaro Gonzalez-Jimenez, M Giglio, and A Manes. A fast fracture plane orientation search algorithm for Puck's 3D IFF criterion for UD composites. *Materials Today Communications*, 28:102700, 2021.

[86] PP Camanho, Pere Maimí, and CG Dávila. Prediction of size effects in notched laminates using continuum damage mechanics. *Composites science and technology*, 67(13):2715–2727, 2007.

[87] PP Camanho, MA Bessa, G Catalanotti, M Vogler, and R Rolfes. Modeling the inelastic deformation and fracture of polymer composites–Part II: Smeared crack model. *Mechanics of Materials*, 59:36–49, 2013.

[88] William H. Press, Saul A Teukolsky, William T Vetterling, and Brian P Flannery. *Numerical Recipes in Fortran 77: The Art of Scientific Computing*. Cambridge University Press, 1997.

[89] PD Soden, MJ Hinton, and AS Kaddour. Biaxial test results for strength and deformation of a range of E-glass and carbon fibre reinforced composite laminates: failure exercise benchmark data. *Composites Science and Technology*, 62(12):1489–1514, 2002.

[90] Tanaka Mori and Kohichi Tanaka. Average stress in matrix and average elastic energy of materials with misfitting inclusions. *Acta metallurgica*, 21(5):571–574, 1973.

[91] Yakov Benveniste. A new approach to the application of mori-tanaka's theory in composite materials. *Mechanics of materials*, 6(2):147–157, 1987.

[92] ESDU. Average gust frequencies subsonic transport aircraft. Technical report, 2006.