U.PORTO
FEUP FACULDADE DE ENGENHARIA
UNIVERSIDADE DO PORTO

Industry-focused development of an analytical tool for rapid performance assessment of composite crash structure designs

Pedro João Cunha

Thesis submitted to Universidade do Porto – Faculdade de Engenharia
in partial fulfilment of the requirements for the degree of
Master in Mechanical Engineering

Supervisors
Professor Albertino Arteiro
Dr. Denis Dalli

Mestrado em Engenharia Mecânica (M.EM)
Departamento de Engenharia Mecânica (DEMec)
Faculdade de Engenharia (FEUP)
Universidade do Porto (UP)

“Success belongs to those who believe in the power of their ideas.”

– Michael Irwin

À minha mãe,

à minha família

Acknowledgements

First and foremost, I'd like to thank Professor Albertino Arteiro for allowing me to work alongside him on such an exciting project. It was a great few months, and I can honestly say that I am glad to have chosen to undertake this project as my master's thesis.

I'd also like to extend my full gratitude to this project's co-coordinator, Doctor Denis Dalli, who, many times during the development of this thesis, took time off his day to help me and offer his guidance and expertise. From the first day, he ensured I was well-accommodated and provided one of the best working environments I've experienced. Finally, I'd like to thank him for all the friendly chats we had throughout these months working together and for helping me to expose my work at conferences and to the McLaren Racing team.

I'd also like to thank the members of the Porto Composites Group, with whom I interacted every day. They made me feel welcome and often offered their help and expertise to solve problems I faced developing the analytical tool.

I have had the privilege of studying under outstanding and intelligent professors throughout my academic career, who have incited my interest in a wide range of engineering subjects and assisted me in finishing my academic journey. To the friends and colleagues I've made during my journey through FEUP, who accompanied me through countless hours of classes, projects and studying for exams.

To all my closest friends who have accompanied me on my journey through life since we were kids. We shared many great and not-so-great moments together, and I am truly blessed to have met them. I hope we can all keep getting along, joking, and having fun together for many more years to come.

Most importantly, I'd like to thank my mother, who worked very hard so I could have the freedom to pursue my degree and follow my dreams. Her support through these years cannot be put into words, and I will forever be grateful to her. Thank you for making me the person I am today.

To my aunts, uncles, grandparents, and family, thank you for all the wisdom, kindness, advice, and support they shared with me.

Abstract

The use of crash structures has significantly increased vehicle safety in recent decades. They are made to gradually absorb energy after an accident by crushing, reducing the impact on the occupants and raising their chances of survival. Composite materials, specifically carbon fibre-reinforced polymers (CFRP), are used to create structures with exceptional strength-to-weight and energy-absorption properties. These materials are used in Formula 1 racing, where vehicles were subjected to severe crash conditions and weight reduction and performance were critical factors. These days, similar structures are also used in high-end consumer cars to improve safety and reduce fuel consumption.

The industry employs numerical methods to predict the energy absorption capabilities of a crash structure since the cost and manufacturing time of composite structures are very high. Macro-scale Finite Element (FE) models are the current industry standard. They can accurately predict the energy absorbed, however, these simulations usually can take up to 12 hours to produce results.

The high computational cost is not an issue in the latter design stages when a final geometry and laminate layup have been defined. However, for the initial stages of design, running an FE model for every design iteration would be highly time-consuming and, therefore, the current early design process relies heavily on in-house knowledge and trial-and-error to achieve a final geometry and material layup.

This work presents an extension of an analytical tool initially developed by Mateus [1] to bridge the gap between the initial and later design stages, allowing users to estimate the energy absorption characteristics of a crash structure quickly. The tool would take as input the CAD model of the structure and the experimental coupon-level data, which is already required for calibrating the numerical models.

All the features introduced in the analytical tool, such as the fitting of the crush stress parameter curves, discretisation of the radius of curvature and inclination, and calculation of the crushing force, were tested to verify that they were correctly implemented and could predict the real-world behaviour of a composite crash structure. The CAD model of the Side Impact Structure (SIS) of a Formula 1 car was adapted to be inputted into the tool, and the parameters influencing the accuracy of the tool were evaluated. Quasi-static and dynamic simulations of the SIS were submitted for analysis, and the results obtained were compared against existing experimental and numerical results to assess the predictive capabilities of the tool.

A test was conducted to evaluate the computational performance of the tool for various tested models. The runtimes ranged from 3 to 61.8 seconds, dramatically faster than existing FE models. The tool was able to predict with high fidelity the numerically obtained results for the quasi-static and dynamic conditions. When compared with the experimental results, the tool replicated the

dynamic results with high fidelity but under-predicted the quasi-static results by 13%.

Resumo

A introdução de estruturas de proteção contra impactos aumentou significativamente a segurança dos veículos nas últimas décadas. Estas estruturas são concebidas para absorver gradualmente, por esmagamento, a energia após um acidente, reduzindo o impacto nos ocupantes e aumentando as suas hipóteses de sobrevivência. Materiais compósitos, especificamente os polímeros reforçados com fibras de carbono (CFRP), são utilizados para criar estruturas com propriedades excecionais de resistência específica e de absorção de energia. Estes materiais foram são utilizados em corridas de Fórmula 1, onde os veículos são sujeitos a condições de colisão severas e onde a redução de peso e o desempenho são factores críticos. Atualmente, estruturas semelhantes são também utilizadas em automóveis de topo de gama para melhorar a segurança e reduzir o consumo de combustível.

A indústria utiliza métodos numéricos para prever as capacidades de absorção de energia de uma estrutura de colisão, uma vez que tanto o custo como o tempo necessários para fabricar estruturas compósitas são muito elevados. Os modelos de elementos finitos (FE) à macro-escala são a atual norma na indústria. Esses conseguem prever com exatidão a energia absorvida; no entanto, estas simulações podem demorar até 12 horas a produzir resultados.

O elevado custo computacional não é um problema nas últimas fases de desenvolvimento, quando a geometria final e a disposição dos laminados já foram definidos. No entanto, nas fases iniciais de desenvolvimento, a execução de um modelo de elementos finitos para cada iteração de *design* consumiria muito tempo e, por esse motivo, o atual processo de *design* inicial baseia-se fortemente no conhecimento interno e na tentativa e erro para obter uma geometria final e uma disposição do laminado.

A ferramenta analítica proposta neste trabalho, inicialmente creada por Mateus [1], foi desenvolvida para colmatar a lacuna entre as fases de projeto iniciais e finais, permitindo aos utilizadores estimar rapidamente as características de absorção de energia de uma estrutura de colisão. A ferramenta utiliza como *input* o modelo CAD da estrutura e os dados experimentais ao nível do provete, que são também necessários para calibrar os modelos numéricos.

O trabalho documenta a totalidade do processo de desenvolvimento da ferramenta analítica, onde as funcionalidades introduzidas, como o *fitting* das curvas dos parametros de *crush stress*, a discretização do raio de curvatura e da inclinação e o cálculo da força de esmagamento, foram testadas para verificar se foram corretamente implementadas e se podiam traduzir o comportamento real de uma estrutura compósita de colisão. O modelo CAD da estrutura de impacto lateral (Side Impact Structure - SIS) de um carro de Fórmula 1 foi adaptado para ser introduzido na ferramenta, e os parâmetros que influenciam a precisão da ferramenta foram avaliados. As simulações quase-estáticas e dinâmicas da SIS foram comparadas com os resultados experimentais e numéricos existentes para avaliar as capacidades de previsão da ferramenta.

Foi efetuado um teste para avaliar o desempenho computacional da ferramenta para vários modelos testados. Os tempos de execução variaram entre 3 e 61,8 segundos, sendo dramaticamente mais rápidos do que os modelos FE existentes. A ferramenta foi capaz de prever com elevada fidelidade os resultados obtidos numericamente para as condições quase-estáticas e dinâmicas. Comparando com resultados experimentais, a ferramenta replicou os resultados dinâmicos com elevada fidelidade, mas demonstrou-se conservador ao prever os resultados quase-estáticos com uma diferença de 13%.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

In recent decades, composite materials, particularly carbon fibre-reinforced polymers (CFRPs), have witnessed a remarkable ascent within the engineering landscape. Initially confined to high-performance applications in aerospace and motorsport, CFRPs have steadily permeated every-day consumer products, including sports equipment and high-end automotive components. This growth trajectory directly results from the exceptional mechanical properties CFRPs possess. Their superior strength-to-weight ratios, impressive stiffness, and remarkable corrosion resistance offer compelling advantages over traditional materials like steel and aluminium.

One particularly significant domain where composite materials are making a profound impact is in the design of crash structures, primarily within the automotive and aerospace industries. These critical components play a pivotal role in occupant safety by absorbing and dissipating the immense energy released during a collision. The demonstrably superior specific energy absorption of CFRPs compared to metallic materials has led to their widespread adoption despite the higher initial costs and the inherent challenges associated with their design and manufacturing.

The design process for composite crash structures remains a complex undertaking due to the intricate behaviour of these materials under impact conditions. Traditional methodologies rely heavily on a trial-and-error approach, necessitating extensive physical testing. This dependence on physical testing translates into significant increases in both design time and associated costs. Alternatively, some design teams might leverage their internal expertise to generate an initial design iteration. This is then followed by an analysis and optimisation phase utilising the Finite Element Method (FEM) at the macroscale level. Further complicating the design process is the inherent anisotropy of composite materials, that is, the variation of their mechanical properties depending on the direction of loading, adding another layer of complexity to the prediction and modelling of the structure's behaviour.

The challenges mentioned above highlight a critical need within the automotive and aerospace industries for a sophisticated analytical tool specifically tailored to the initial design stage of composite crash structures. Such a tool would empower engineers to rapidly prototype and iterate through various design concepts with ease, leading to a more streamlined development cycle and significant cost reductions by minimising reliance on physical testing.

1.1 Background

The origins of composite materials can be traced back to the ancient civilisations. Egyptians utilised mudbrick reinforced with straw for structural applications, while civilisations in Mesopotamia and South America employed composites of reeds and bitumen[2]. These early examples, though rudimentary, served as the foundation for future advancements, highlighting the potential of combining dissimilar materials to achieve superior properties.

The 19th century witnessed a significant leap forward with the development of new materials like synthetic polymers and glass fibres. The invention of Bakelite, the first synthetic plastic, in 1907 by Leo Baekeland marked a watershed moment [3]. Bakelite's lightweight and insulating properties made it ideal for various applications, including electrical components and aircraft construction. Concurrent with Bakelite's development, the 1930s saw the introduction of glass fibres by Owens-Illinois Glass Company, offering a high-strength reinforcement material [4]. This synergistic development led to the creation of the first modern composites, finding their initial application in military aircraft like the de Havilland Mosquito bomber during World War II [5].

Following the war, the aerospace industry remained a major driver of composite development. The relentless pursuit of lightweight yet robust materials for high-performance aircraft fueled research and development efforts. Epoxy resins emerged as a superior alternative to Bakelite, offering improved strength and adhesion properties [6]. By the 1960s, composite materials like fibreglass-reinforced plastics had begun to find application in commercial aircraft components such as rudders, elevators, and ailerons [7].

In the latter half of the 20th century, the automotive industry began investigating the potential of composite materials in crash structures. However, accurately predicting their behaviour under crush loads proved a major challenge. Unlike the established engineering principles governing the deformation of metals like aluminium and steel, the crushing behaviour of composites remained largely unknown. This presented a significant disadvantage compared to traditional materials. The well-understood energy absorption mechanisms of metals, characterised by progressive buckling and plastic deformation, contrasted starkly with the complex and poorly understood failure mechanisms of composites under crash conditions. This knowledge gap translated into substantial safety concerns, ultimately restricting the initial use of composites to non-structural components like body panels.

The development of sophisticated computer modelling tools, such as Finite Element (FE) analysis, proved to be a critical turning point. FE analysis enabled engineers to simulate the composite structures' complex deformation mechanisms and energy absorption characteristics during a crash. With the increasing accuracy and accessibility of these tools, the use of composites in automotive and aeronautical crash structures gradually gained traction. Today, composite materials play a vital role in the construction of high-performance vehicles, particularly in Formula 1 racing, where lightweight and high-strength components are paramount.

1.2 Motivation

Composite material structures offer better performance and high design flexibility, but their high complexity has created complicated certification processes that manufacturers must observe. In particular, crashworthiness certification for composites presents a unique challenge. The traditional safety factor approach is not readily applicable due to these materials' complex energy absorption behaviour. Overly conservative designs, while seemingly prudent, can lead to excessively stiff structures that subject the occupants to excessive deceleration forces during a crash event. However, engineers often risk having less conservative crash structures to save weight, which might compromise their energy absorption capability and leave the core vehicle structure to bear the brunt of the impact. Therefore, composite crash structures must be meticulously designed to operate within specific, tightly controlled energy absorption limits.

The development of high-performance crash structures presents a significant financial challenge. Material and labour costs in an F1 crash structure can easily exceed 20,000 €, rendering extensive physical testing prohibitively expensive. Historically, this has led to overly conservative designs prioritising safety over potential performance gains. Finite Element (FE) crash modelling software offers a promising solution, enabling the exploration of bolder design concepts at a fraction of the cost and within a shorter timeframe. However, these tools require further development, and the results obtained from the simulations must still be certified through experimental testing.

Additionally, the computational cost associated with FE simulations can be significant. A single crush simulation for a small-scale structure on a single CPU can require over 12 hours of processing time. While FE models are invaluable for later-stage design refinement, their computational intensity hinders their effectiveness in the crucial initial design phase, which focuses on determining the material selection, preliminary laminate configuration and geometric features necessary to meet the target performance criteria of the crash structure.

Currently, the initial design stage is a heavily interactive process that typically relies on internal design expertise and a trial-and-error approach to reach a final design solution. The current design landscape lacks a readily available, computationally inexpensive analytical tool tailored explicitly for early-stage material and layup selection in composite crash structures, enabling engineers to test new ideas quickly. This tool would not supplant the intricate FE models used for detailed structural response and energy absorption prediction in later design phases. Instead, it would be a complementary tool, streamlining the overall design process.

1.3 Thesis Objectives

This thesis aims to expand upon a program, initially created by Mateus [1], that can analytically predict the energy absorbed by a crash structure, given its CAD model and experimental coupon-level data describing the material's energy absorption parameters. Every feature developed for the tool was tested to ensure it was correctly implemented and could accurately represent how a composite crash structure would behave in the real world. It is also an objective of this work that the tool can obtain a final prediction quickly so that engineers can effectively use it during the initial design stage.

Another goal of this thesis is to verify the accuracy of the analytical predictions obtained for a Formula's 1 Side Impact Structure (SIS) by comparing them against existing experimental and numerical results found in the literature.

1.4 Thesis Outline

Chapter 2 conducts a comprehensive literature review on crashworthiness and explores the parameters used to quantify a structure's suitability for crash use. This chapter also addresses composite materials, their failure mechanisms, and their failure modes during crushing. Crush testing and the geometric effects on crush stress are explored. The chapter concludes with the numerical and analytical models currently used for crashworthiness modelling and methods used for uncertainty quantification.

Chapter 3 explores the entire development process for the analytical tool. It explains the programs used and exposes the tool's overall functioning process. This chapter also verifies the correct function of all the features developed. Additionally, crush stress parameter curves for a carbon fibre and a carbon-flax hybrid laminate are fitted from the experimental data obtained by Moreira [8].

Chapter 4 documents the process of adapting a pre-existing SIS CAD model to be submitted for analysis. The effects of partitioning the CAD model on the tool's accuracy are also explored.

In Chapter 5, the results obtained from the SIS simulations are compared to experimental and numerical results from the literature to assess the tool's accuracy. The analytical tool's computational performance is also analysed.

Chapter 6 presents the conclusions and possible future work to increase the tool's predictive accuracy. It explores new features that could be implemented.

Chapter 2

State-of-art and Literature Review

With composite materials becoming more widespread and accessible over the years, they have been replacing traditional materials such as steel and aluminium alloys in the automobile, aeronautical, and aerospace sectors as the preferred structural material [9] due to their low weight, very high strength and stiffness and great corrosion resistance [10]. However, using composite materials comes with new design challenges, such as the material's anisotropy, different manufacturing processes, and new failure mechanisms.

A detailed literature study is carried out on the state of the art for using composite materials in energy-absorbing structures during crushing. This chapter explains the crushing phenomenon, how to analyse it, and the key parameters that describe a structure's energy absorption capability during axial compression, followed by an analysis of the fracture mechanics of composite materials. Afterwards, the current testing methods and coupons for crashworthiness applications will be presented, and the main parameters affecting crush stress will be explained. Finally, an overview of the current crashworthiness predictive models and an exposition of relevant software and programming languages will be provided.

2.1 Crashworthiness

Crashworthiness refers to a structure's ability to induce controlled failure modes during a crash to absorb the impact's energy and guarantee the safety of the occupants [11]. These energy-absorbent structures are typically employed by the automobile and aeronautical industries as safeguards in the event of an accident to decrease the risk of deaths or injury by the occupants; however, several design considerations such as fuel economy, weight, energy absorption, cost and available space need to be taken into account in the design of these structures[11].

With growing concerns about sustainability and performance, the automobile industry has been making a significant effort to pursue lightweight materials and structures for their vehicles without compromising safety [12]. With studies showing that fuel consumption is directly proportional to the vehicle's weight, where a 10% weight decrease translates into 6% to 8% less fuel consumption [13], the incentives for structural design optimisation are clear, especially in applications where performance is the most critical consideration such as motor racing.

For the reasons mentioned above, there has been an evolution in the crashworthiness structure materials, from traditionally being made from steels that absorb the impact energy through plastic deformation [14] to lighter, cheaper, and more ductile aluminium alloys with higher specific energy absorption [15], providing more cost-effective energy-absorbing devices for automotive applications. Additionally, the advancements in the field of composite materials, specifically Carbon Fibre Reinforced Polymers (CRFPs), have enabled the development of composite crashworthiness structures, which due to their different failure modes under crushing that will be discussed in more detail in Section 2.4, can offer higher specific energy absorption than metals, allowing the manufacture of lighter structures, albeit at a higher cost [14].

In Formula 1 (F1), where competing teams go to extreme lengths to achieve the lowest possible lap times, the weight optimisation of the crash structures is a critical design factor. F1's regulatory body, the Federation Internationale De L'Automobile (FIA), imposes safety regulations that must be observed by all competing teams and drivers and track regulations to ensure the driver's safety in the event of a crash. One such measure is the survival cell for the driver. This composite monocoque structure houses the driver and has mounted on it three energy-absorbent structures[16]: the Front Impact Structure, the Side Impact Structure, and the Rear Impact Structure (Figure 2.1).

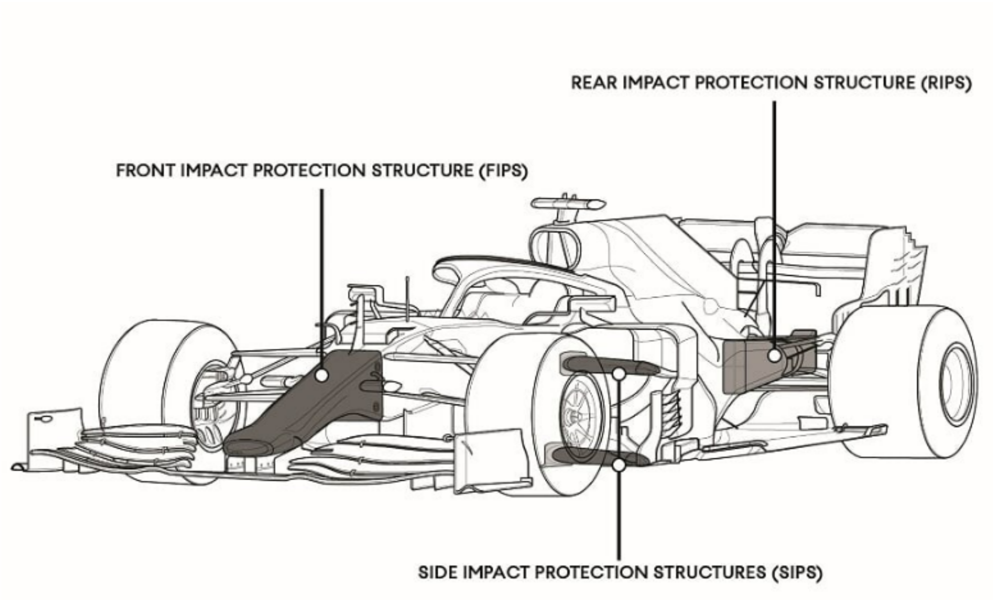


Figure 2.1: Impact structures of a Formula 1 car [17]

To predict the amount of energy absorbed by a structure, axial crushing tests can be performed on specimens, either quasi-statically or dynamically, to obtain force-displacement curves from which several parameters can be calculated to describe the crashworthiness of the structure. Additionally, the force-displacement curves allow the calculation of the deceleration that will be felt by the occupants, which is a critical parameter to consider since if too much energy is absorbed, the deceleration can be too great and cause injuries. On the other hand, if there is insufficient energy absorption, the impact will cause injuries [18].

2.1.1 Total Energy Absorption

The Total Energy Absorption (EA) can be extracted by integrating the experimentally obtained force-displacement curve through the displacement, as specified in Equation (2.1), and is a measure of the total energy that the structure absorbed during the crushing experiment. Figure 2.2 shows the typical load-displacement curve of a thin-walled composite structure crush test.

$$\text{EA} = \int_0^S F(\delta)d\delta \tag{2.1}$$

,where δ is the displacement of the crushing head, $F(\delta)$ is the measured force at the displacement δ and S is the crushing length.

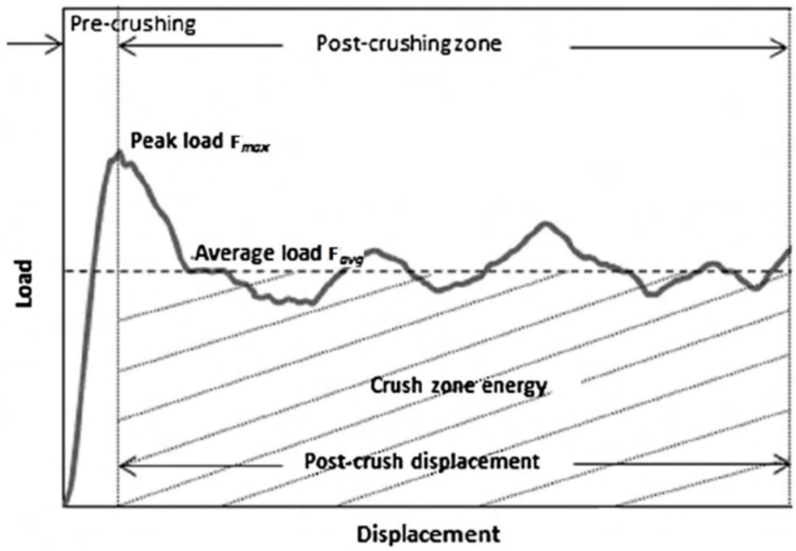


Figure 2.2: Typical load-displacement response of a crush test (adapted from [19])

Mamalis et al. [20] identified several parameters impacting composite materials' energy absorption capacity, including fibre and matrix materials, laminate design, temperature, structural geometry, and loading direction.

2.1.2 Specific Energy Absorption

Specific Energy Absorption (SEA) measures how much energy is absorbed per unit of mass of the material. It is one of the most important parameters to consider since it provides information on how effectively a specific material absorbs energy relative to its weight. In essence, considering two materials with distinct SEAs, the material with a higher SEA will allow the manufacture of a lighter structure than the material with a lower SEA for the same amount of energy absorbed. It is also a useful parameter to compare the efficiency of structural designs by quantifying the effects of geometry or layup. SEA can be calculated with Equation (2.2):

$$\text{SEA} = \frac{1}{m}\int_0^S F(\delta)\text{d}\delta = \frac{\text{EA}}{m} \tag{2.2}$$

,where m is the total crushed mass.

2.1.3 Peak Force

The Peak Force (F_{max}) is given by Equation (2.3) and represents the maximum force during the crushing of the structure and, consequently, to the moment of maximum deceleration experienced by the passengers. The peak force usually occurs at the start of the crush test when the structure still has not entered the stable crushing stage and corresponds to the structure's elastic response. A peak force value which is too high typically either indicates that the structure is overly stiff or that the choice of the geometric trigger to initialise stable crushing was inadequate and the structure might fail be failing unstable [20, 21].

$$F_{\max} = \max(F(\delta)) \tag{2.3}$$

2.1.4 Steady-state Force and Crush Efficiency

The Steady-State Force (F_{avg}) is defined as the average load during the steady stage crushing of the structure and can be calculated via Equation (2.4).

$$F_{\text{avg}} = \frac{\int_{\delta_1}^{\delta_2} F(\delta)\mathrm{d}\delta}{\delta_2 - \delta_1} \tag{2.4}$$

,where δ_1 and δ_2 are the displacement bounds where progressive crushing starts and ends, respectively.

Crush efficiency (η_c) is given by the steady-state and peak forces ratio and is expressed by Equation (2.5). This measure assesses the stability of the crushing process and is influenced by geometry, ply orientation, and velocity [22, 23]. Low efficiency increases the structure's susceptibility to instability caused by high initial peak force. Crash structures with increasing cross-sectional area are widely used to improve crushing efficiency.

$$\eta_c(\%) = \frac{F_{\text{avg}}}{F_{\max}} \cdot 100 \tag{2.5}$$

2.1.5 Crush Stress

The crush stress (σ_c) can be obtained from the force-displacement graph by dividing the force obtained at a certain displacement ($F(\delta)$) by the area of the cross-section being crushed at that displacement ($A(\delta)$), in accordance to Equation (2.6):

$$\sigma_c(\delta) = \frac{F(\delta)}{A(\delta)} \tag{2.6}$$

Having information on how the crush stress varies along the displacement of the crushing head, a graph of crush stress vs displacement can then be plotted. The advantage of utilising crush stress graphs vs displacement over force vs. displacement graphs lies in their independence on specimen cross-section, which makes them a better tool to characterise the structure's crushing behaviour. An in-depth analysis of the factors influencing the crush stress will be made in Section 2.6.

2.2 Composite Materials

Composite materials comprise two or more constituent materials exhibiting noticeably different chemical or physical properties. These components are combined to create a new material with unique qualities, although the individual elements stay separate and distinct within the finished structure [24]. Composite materials are widely employed in various industries due to their superior characteristics over traditional materials, including superior strength, durability, and efficiency [25]. Some examples of commonly used modern-day composite materials include reinforced concrete, composite wood such as plywood, reinforced polymers like fibreglass or carbon fibre, ceramic matrix composites, and metal matrix composites.

A composite material's typical composition is the reinforcements, the load-bearing element, and the matrix, the binding element. Reinforcements, typically in the form of fibres such as carbon fibres, glass fibres, aramid fibre, or natural fibres, are incorporated into the matrix and carry loads along their length, providing strength and stiffness to the composite and delaying crack propagation [26]. The matrix holds the reinforcements together, distributes the load between fibres, and gives the part its geometric configuration. Commonly, composite matrices are organic, polymeric, or ceramic, with the particular distinction of thermosetting resins widely used in composites for high-performance applications. Embedding the matrix with additives is also possible to enhance specific mechanical properties.

2.2.1 Carbon Fibre Reinforced Polymers

Carbon Fibre Reinforced Polymers (CFRPs) are increasingly used in the automobile and aeronautical industries due to their low specific weight, high strength-to-weight ratio, high stiffness, corrosion resistance, and fatigue resistance. Despite their high cost, this plethora of mechanical and chemical properties has made CFRPs ideal for high-performance applications. Their use is also common in F1 crash structures since CFRPs provide considerably higher levers of energy absorption [20]. For reference, tubes made from steel and aluminium have been shown to have 70% and 50% less specific energy, respectively absorption than an equivalent CFRP tube [27].

The carbon fibre reinforcements embedded in the matrix can be discretised into short and continuous fibres. Short or discontinuous fibre components are made from chopped tow and pressed into a resin-filled mould of the desired part, creating an almost isotropic material with much lower mechanical properties than continuous fibre components [28]. The lower properties of discontinuous fibre components and high manufacturing versatility make their use popular for non-structural and aesthetic applications. They are a good way of recycling the carbon fibres from decommissioned parts.

Continuous fibres provide excellent stiffness and strength when the load direction coincides with the fibre direction and much lower properties in the perpendicular and out-of-plane directions. The resulting material is highly anisotropic; however, if the load case is known, the fibre direction can be optimised to produce structures much lighter than what would be possible with traditional materials. F1 teams utilise recently developed intermediate modulus (IM) fibres with very high strength that are capable of superior specific energy absorption, provided that the surrounding matrix is tough enough to withstand fibre-matrix decohesion and ply delamination [29].

Furthermore, CFRPs can also be differentiated by the type of matrix: thermoset matrix and thermoplastic matrix. Currently, the most common type of matrix used for high-performance applications is the thermoset matrix, mainly due to the ease of curing during the part's manufacturing process and the excellent properties obtained from the epoxy-based thermoset matrix. However, thermoplastics can soon become the new norm in composite design. Improvements in the manufacturing of thermoplastic resins have made them more appealing to the industry, and, in the specific case of crush applications, they offer higher SEA than thermosetting resins. Additionally, thermoplastics can be melted once the component lifetime has ended and can be reused in other parts, whilst thermosetting resins are not recyclable [30, 31].

Most commercially sold continuous fibre composites are in unidirectional (UD) or 2D woven configurations. UD composites were the first continuous fibre systems used in structures due to their high specific stiffness in the fibre direction; however, their poor out-of-plane properties can cause delaminations [32, 33]. In F1, 2D woven materials are utilised due to their balanced in-plane qualities, increased drapability, and superior interlaminar properties [34]. The configuration of the weave, that is, the pattern of how the warp and weft tows interlace to create the weave, is a parameter that affects stiffness and energy absorption. Higher crimping weaves (more number of crossing-overs of the weft and warp tow yarns) are associated with lower drapability, lower strength and stiffness due to a higher portion of the fibres being misaligned with the in-plane directions and higher energy absorption capability due to increased transfer of energy between the tows during crushing [35].

Carbon fibre plies are usually bought in the pre-impregnated configuration (prepreg), where the fibres are already impregnated in a partially cured epoxy resin, which must then be consolidated in an autoclave after draping. This configuration allows the manufacturer to skip the resin infusion of dry performs, a complex and laborious process prone to manufacturing defects, and offers a more straightforward and consistent result.

2.2.2 Natural Fibre Reinforced Polymers

The increasing utilisation of Natural Fibre Reinforced Polymers (NFRPs) comes from the industry's attempts to reduce its carbon footprint by replacing carbon fibre, an oil-based product, with natural fibres that can be sustainably produced and recycled. They are derived from renewable plant-based sources, have low density, and offer competitive mechanical properties compared to traditional materials. However, they are still significantly weaker than identical carbon fibre laminates. All these factors make NFRPs a viable and sustainable alternative for commercial applications where performance is secondary. However, more advancements are still needed before they can replace carbon fibres in high-performance applications. The introduction of NFRPs into crash structures as an energy-absorbent material comes from regulatory changes from governing bodies, such as the FIA in Formula 1, aiming to reduce the environmental impact and dependence on fossil fuels [36].

Flax Fibre Reinforced Polymers (FFRPs) are currently the best natural alternative to carbon fibres for crash applications. This is mostly because they combine the best aspects of low cost, low density, and greater mechanical and specific energy absorption qualities over other natural fibres [37]. Research on quasi-static testing to characterise the FFRPs revealed similar mechanical properties to glass fibre reinforced composites [38, 39].

However, several problems have been documented regarding FFRPs. Batch variability is a major disadvantage for FFRPs; the fibres' quality depends on several external factors, such as weather conditions, harvest time, and soil quality, leading to variations in the final product [40]. Flax fibres' tendency to absorb moisture leads to poor dimensional stability and causes fungal development [41], requiring the surface of FFRP laminates to be chemically treated to delay these effects [40]. Additionally, their hydrophilic nature causes bonding problems with the hydrophobic thermoset matrices, which poses a significant manufacturing challenge and is one of the plant fibre's main drawbacks [42, 43]. Finally, it is worth noting that most thermoset matrices are still oil-based products, which decreases the environmental advantages of using flax fibres.

To assess the suitability of FFRP for crush applications, Lobão [44] and Moreira [8] tested tubular and flat FFRP coupons under dynamic and quasi-static conditions. Due to the low compressive strength of the flax fibres, the authors experienced premature coupon failure on several flat coupons and the smaller radii tubular coupons. Additionally, for the coupons that did not fail prematurely, the overall energy absorbed and specific energy absorbed was low. The authors concluded that laminates exclusively made from FFRP were unsuitable for crush applications.

A hybrid laminate composed of carbon and flax fibre plies could present an acceptable solution. It would compensate for the flax's inferior mechanical properties while being a more sustainable and cheaper option than a pure carbon fibre laminate. Strohrmann et al. [45] evaluated the crashworthiness of flax/carbon fibre reinforced polymer hybrid tubes. The authors investigated the crushing performance of four distinct laminates: $[C]_8$, $[C_2/F_2]_S$, $[F_2/C_2]_S$, and $[F]_8$ under quasi-static and dynamic testing conditions, where C denotes carbon fibres, and F denotes flax fibres. The results revealed that FFRP tubes exhibited lower energy absorption than CFRP tubes. However, the hybrid tubes demonstrated interesting crush stress and specific energy absorption (SEA) values, exceeding the arithmetic mean of the corresponding values for their individual CFRP and FFRP counterparts. Moreira [8] tested tubular and flat hybrid coupons under dynamic and quasi-static conditions and found that the premature failure of FFRP coupons had been eliminated on the hybrid coupons and that the carbon-flax hybrid laminate could be used for crash structures.

2.3 Composite Failure Mechanisms

Composite failure mechanisms of thin-walled structures under compressive loading differ significantly from metallic ones. The latter primarily undergoes plastic deformation, progressively buckling the structure and absorbing energy, whilst composite structures crush progressively without plastic deformation of the structure and absorb energy through complex fracture and damage propagation mechanisms occurring around the interface of the crushing head and the structure. The overall amount of energy absorbed and the failure modes observed are a function of the material properties, geometric features of the structures and testing conditions such as strain rate [46]. Additionally, to absorb the maximum amount of energy, a composite structure should crush stably and progressively, as failures related to geometric instability (for example, buckling) will lead to a premature catastrophic failure and significantly reduced total energy absorption.

During stable crushing of a laminate, a study by Savona and Hogg [47] observed delaminations, matrix cracking, fibre fracture, intralaminar crack growth and shear fracture mechanisms that characterise the compressive failure behaviour, as illustrated by Figure 2.3. The total amount of energy absorbed can be considered the sum of all the energy absorbed by the individual fracture mechanisms acting on the structure, following Equation (2.7).

$$U_{\text{total}} = U_{1\text{c}} + U_{2\text{c}} + U_{\sigma} + U_{\text{ff}} + U_{\text{fr}} + U_{\text{others}} \tag{2.7}$$

Where:

- $U_{1\text{c}}$ is the Mode-I fracture energy. Mode-I (opening mode) fracture energy describes the energy absorbed by delaminations in the laminate caused by tensile stresses acting normal to the delamination plane; the governing material parameter is Mode-I fracture toughness G_{I_c}.
- $U_{2\text{c}}$ is the Mode-II fracture energy. Mode-II (sliding mode) fracture energy describes delaminations in the laminate caused by shear stresses acting parallel to the plane of the crack; the governing material parameter is Mode-II fracture toughness G_{II_c}.
- U_{σ} is the energy absorbed by bending of the fronds. At the initial stage of crushing the laminate, a central crack may appear, splitting it into two and creating fronds that splay inwards and outwards and are progressively bent by the crushing head as it moves downwards, forming transversal cracks in the laminate and absorbing energy.
- U_{ff} is the fibre fracture energy. It represents the energy absorbed by the fracture of the fibres due to high stresses, either from tension or compression.

- U_{fr} is the friction energy. Friction occurs at the interface between the fronds and the crushing head, between fronds and between the debris wedge of crushed material on the central crack and the fronds.
- U_{others} is the energy absorbed by other unspecified mechanisms, such as Mode-III (tearing mode) fracture.

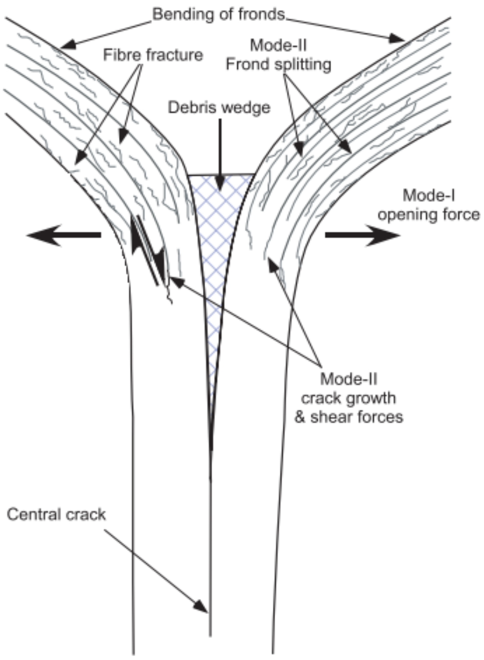


Figure 2.3: Schamatic representation of the fracture effects at the crush front of a flat coupon (adapted from [47])

Some studies have quantified the energy absorbed by each fracture mechanism compared to the total energy absorbed. Hussein et al. [48] who studied the crushing behaviour of square tubes, found that shear-dominated fractures dominated the energy absorption, being responsible for 65% of the total energy absorbed, followed by friction, which accounted for 30% and the remaining 5% were attributed to crack propagation. Boria et al. [49], who studied thin-walled cylinders and cones concluded 22% energy absorption from bending of the fronds, 20% from fibre fracture, 27% from crack propagation and 31% from friction, differing from the values obtained by Mamalis et al. [50] where the frictional energy, the bending energy, the crack energy and the axial splitting correspond 45%, 40%, 13% and 2% of the total energy absorbed, respectively. It is clear from the literature that there is no consensus on how much each factor contributes to total energy absorption and more research needs to be done on this topic; it can be concluded, however, that both material choice and the geometry of the structure both are determinant factor on how much total energy is absorbed and by what mechanisms.

The remainder of this section will focus on some of the failure mechanisms to provide a complete understanding of crush failure in composite structures.

2.3.1 Fibre Fracture

Fibre fracture refers to the breakage of individual fibres due to the stresses acting on the tows exceeding either the fibre's ultimate tensile strength or ultimate compressive strength or, alternatively, the ultimate tensile or compressive strains. Large compressive stresses can also cause the formation of kink bands, localised regions on the material where the fibres have buckled and rotated, resulting in a band of misaligned fibres [51] (see Figure 2.4). These kink bands are the mechanism by which laminate failure under compressive loads is initiated.

Usually, individual fibres snap at distinct values of the applied strain. However, the remaining fibres will experience localised stress concentrations due to the loss of a load-bearing fibre, which can trigger adjacent fibres to fail. This effect accumulates until complete composite failure is achieved [52].

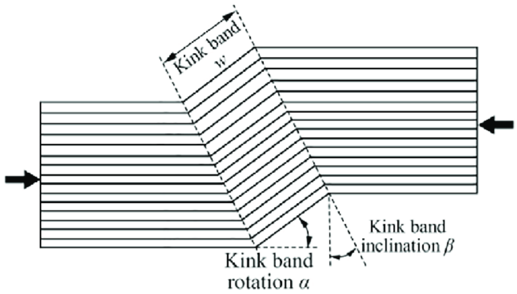


Figure 2.4: Schematic representation of a kink-band [53]

2.3.2 Matrix Cracking

Matrix cracking refers to the proliferation of cracks along the matrix of the composite material. It typically results from normal transverse or in-plane shear stresses being applied in the laminate, which are entirely or partially supported by the matrix [54]. A composite with a cracked matrix may still withstand high loads, making it a non-catastrophic form of damage. However, the continuous propagation of cracks will keep degrading the laminate's mechanical properties until failure occurs [52].

It is also worth noting that cracked matrixes can induce other failure mechanisms. For example, should a crack on a transversely loaded ply propagate through the ply thickness, it will cause premature delamination from the adjacent plies [55] (see Figure 2.5). Alternatively, matrix cracking may result in localised clustering of fibre fractures.

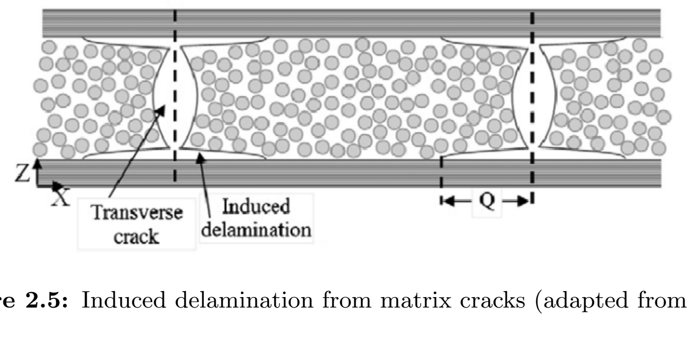


Figure 2.5: Induced delamination from matrix cracks (adapted from [56])

2.3.3 Delamination

Delamination is defined as the peeling apart of bonded plies and is mainly caused by a peel or shear stress acting on a free edge of a laminate [52]. Even so, matrix cracking is almost always a precedent to delamination, as was exemplified in Section 2.3.2, alongside bending and shear cracks

[57]. Delamination of the laminate significantly affects its compressive strength and may cause unstable crushing [57], making this form of damage particularly insidious, especially considering that the delamination damage might be internal to the laminate and challenging to detect on its surface.

Woven composites have greater resistance to delamination than UD composites. The weave creates a texture in the plies interface that significantly prevents delamination crack growth once it reaches the weave structure during its propagation path [58].

2.4 Crushing Failures Modes

Mamalis et al. [20] categorised the macroscopic collapse modes of axially crushed tubes as unstable or progressive/stable. For crashworthiness applications, only progressive crushing may occur as a failure mode due to much lower energy absorbed by an unstable fracture, as illustrated in Figure 2.6.

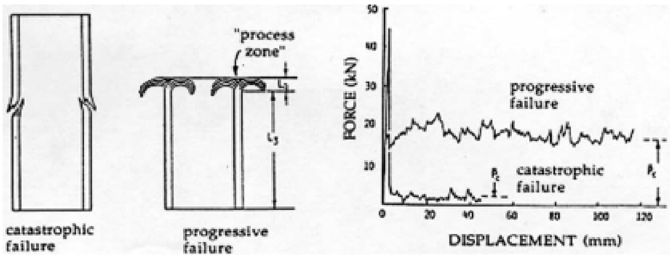


Figure 2.6: Pogressive and catastrophic failure of axially crushed composite tubes [59]

Experiments conducted by Hull [60] on $[0/90]_S$ glass fibre polyester resin tubes and glass fibre cloth-epoxy tubes identified two distinct modes of progressive crushing occurring on the specimens: splaying mode and fragmentation mode. These failure modes are related to the primary failure mechanisms of composite material described in Section 2.3.

2.4.1 Splaying Mode

The splaying failure mode is characterised by a central crack in the laminate, creating delamination of the plies in the axial direction, forming fronds [45]. Furthermore, some fronds will bend outwards while the remainder will bend inwards. If the same number of plies bend outwards and inwards, the failure mode can be categorised as symmetric slaying of the laminate. Otherwise, the failure mode is asymmetric splaying.

For his studies on $[0/90]_S$ tubes, Hull [60] utilised the notation that 0° corresponded to the hoop direction and 90° to the axial direction. During the axial compression of the tube, the axially aligned fibres located in the two centres plies of the laminate support most of the compressive load. The hoop fibres, located on the surfaces of the laminate, support the axial fibres, maintain the integrity of the laminate, and minimise premature buckling.

To induce the failure mode, damage initiation triggers, typically chamfers, are machined to the top of the specimen. As the trigger is crushed, the plies begin to bend and a hard debris wedge composed of thinly crushed fibres and resin forms along with a central crack. As crushing progresses, the debris wedge is forced downwards, propagating the crack and delaminating the laminate beneath. At this point, the bent axial fibres no longer provide structural stability, and the hoop fibres provide strength by holding the laminate together and resisting the progression of the splaying until the outer hoop fibres break due to the tensional stresses and internal hoop fibres due to compressive stresses [60]. Figure 2.7 illustrates the described process.

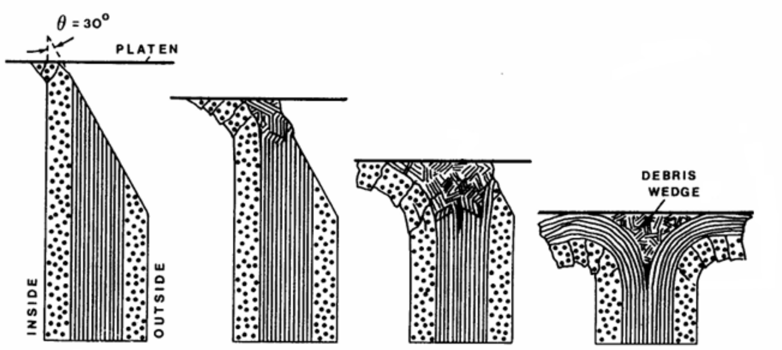


Figure 2.7: Initiation of the splaying mode on a composite tube (adapted from [60])

The energy absorption during the splaying of a structure is mainly attributed to the energy dissipated by friction between fronds and the crushing plate, intralaminar shear cracking of the matrix, tensile and compressive failure of the hoop fibres, central crack propagation, Mode-I delamination and bending of the fronds [60]. Figure 2.8 showcases the forces acting on the crushing zone of the structure.

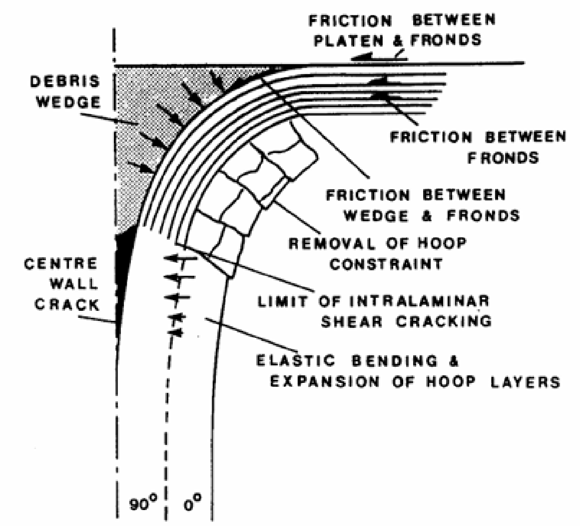


Figure 2.8: Forces acting on the crushing zone [60]

2.4.2 Fragmentation Mode

Fragmentation is a crushing mode characterised by the development of in-plane failure mechanisms in the laminate. After the damage trigger is crushed, the compressive forces generate high shear stresses, which cause the propagation of interlaminar shear cracks that break the laminate into large fragments. The hoop constraints retain structural integrity and hold all the fragments together. Still, as the crushing head progresses, the hoop fibres induce compressive and tensile stresses, further breaking the fragments into smaller pieces until the hoop fibres break from compressive buckling or tensile failure and that section of the tube collapses. Figure 2.9 illustrates the described process.

On the review by Hull [60], experimental tests with composite tubes of glass fibre polyester with

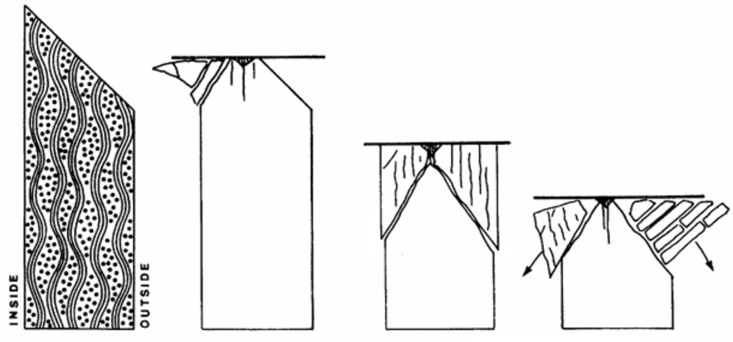


Figure 2.9: Initiation of the fragmentation mode on a composite tube [60]

a woven $[0/90]_S$ configuration failed through fragmentation mode. The same layup was utilised for the tests for the splaying and fragmentation modes; however, the laminate was composed of UD plies for the splaying mode, while for the fragmentation mode, a 2D woven structure was used. These results indicate that the number of hoop fibres in the laminate dictates which failure modes will be observed during the progressive crushing of a tube, where tubes with higher quantities of fibres in the hoop direction are more likely to display the fragmentation mode.

Berry [61] investigated the impact of various hoop and axial fibre ratios (H:A ratio). A range of prepreg glass-polyester fabric materials with varying warp-to-weft ratios were used to create 10-ply tubes with a thickness of 2 mm. Fragmentation was the primary method of crushing for the tubes with a higher fibre count in the hoop direction. Similar to what was observed by Hull [60], the hoop fibres experienced compressive and tensile loads as a result of the shear crack development, which led to the fracture of the fragments into pieces of different lengths. The most common failure mode was splaying for the remaining tubes with H:A ratios equal to or less than 1.

2.5 Crush Testing

Axial crushing tests must be performed under static and dynamic loading conditions. During the tests, data is collected to calculate the crashworthiness parameters described in Section 2.1 and an analysis is performed on the structure's failure modes and fracture mechanisms. The crashworthiness parameter of highest relevance is the crush stress, which depends not only on the material's choice but also on the geometric characteristics of the structure in question, such as curvature, inclination relative to the crushing direction, strain rate and laminate thickness. A more comprehensive review of the geometrical parameters affecting crush stress will be performed in Section 2.6.

In an attempt to create a database of materials and structural designs for crashworthiness applications, a group formed by entities from the aerospace, automotive, academia, and government laboratories banded together and developed a document of standardised methods for the testing of laminated flat coupons and thin-walled tubular specimens, denominated Composites Material Handbook (CMH) [62, 63]. Testing for crashworthiness and energy absorption is typically done quasi-statically at the coupon level due to fully developed structures' high manufacturing and material costs and the higher price and complexity of performing dynamic tests [64]. However, for applications in practical accidents where the structure is subjected to high-speed decelerations, as is the case of the impact structures in an F1 car, the certifying organisations require dynamic testing of the final structure before the crashworthiness certification for the part can be acquired [65].

2.5.1 Crash Strucutres

Energy-absorbing structures are paramount in moving vehicles to ensure the safety of their occupants during a collision. Ensuring that the crash structure is not over or under-dimensioned is critical. Overdimensioning can cause rapid decelerations, putting passengers at risk. Underdimensioning can render the structure incapable of absorbing enough energy to protect passengers' physical integrity [22]. The FIA requires Formula One impact structures to undergo dynamic tests with an initial impact velocity of at least 15 m/s.[66].

Dynamic testing of impact-absorbing structures involves complicated experimental setups, such as drop towers or sled impactors and significant financial investment from firms [23]. Coupon-level testing is more effective for characterising materials and determining the relevant mechanical properties for design. Ideally, coupons should have geometries similar to the final structure and identical manufacturing procedures for coupon testing to be representative of the component testing. Coupon tests can additionally be used to develop databases for numerical or analytical crashworthiness modelling [67].

2.5.2 Open Section Coupons

Many different geometries of open-section coupons have been tested. Several studies focused on testing of flat coupons [18, 68–70], while Feraboli [71] studied the energy absorption of C-channels, corrugated and semi-circular geometries and Waimer et al. [72] investigated C-channels for aeronautical applications.

The testing of flat coupons has some advantages and disadvantages compared to closed-section coupons. They are cheaper, more accessible, and quicker to manufacture. However, their non-self-standing nature requires using specific fixtures and careful consideration of the unsupported height. Dalli et al. [23] and Feraboli [70] reported challenges with the unsupported height causing geometric instabilities such as buckling, compromising the energy absorption capabilities of the flat coupon and invalidating the test. Additionally, according to Feraboli [70], based on the analysis of the supports used for flat coupon testing, the way the coupons are fastened to the test support has a significant impact on the crush response and failure modes

The choice of unsupported height in the test of flat specimens significantly affects energy absorption. In addition to the geometric instabilities that arise from a too-long unsorted height, having a too-short height will cause the specimen to be overly constrained and lead to unusually high energy absorption levels. Also, the unsupported height gradually becomes shorter as the crushing plate moves. It can lead to an increase in force over the displacement, preventing the calculation of an accurate estimation for the crush stress [18]. However, this effect can be counteracted by utilising special fixtures, such as the one developed by Vigna et al. [73], that allow the flat coupon to slide and maintain a constant unsupported height throughout the test.

The type of damage initiation trigger machined unto the specimen can also affect the coupon's crushing response. Test coupons with V-shaped triggers were found to promote splaying and lamina bending, as was observed by Ueda et al. [68]. Test specimens with a triangular through-thickness (TTT) trigger facilitated fragmentation failure [69]. Israr et al. [74] studied a bevel trigger and found it to aid out-of-plane shear and splaying. When Feraboli [70] evaluated steeple and sawtooth triggers, he discovered that steeple triggers had a lower energy absorption capacity despite displaying identical failure modes.

2.5.3 Closed Section Coupons

Several studies have been conducted on the axial crushing of closed-section specimens: round tubes [23, 44–46, 60, 75], square tubes [48, 76–78] and cones [50, 79–82]. Despite the higher material costs and more complicated manufacturing procedures [44], the increased structural stability, their self-standing nature and higher energy absorption capabilities [18] makes them an excellent alternative to flat coupons for material characterisation.

The geometrical similarity of circular tubes and cones to energy-absorbent structures is also appealing because the failure modes and amount of energy the specimen absorbs should be similar to those of the crash structure, which can provide helpful design guidelines to engineers. Additionally, the self-standing nature of the specimens circumvents the need to develop specialised fixtures for the crush test. However, as reported by Dalli [18], not fixing the specimen to the testing machine can lead to mechanical vibrations and make interpreting the results challenging, especially under dynamic testing conditions.

Several tests have been conducted on circular tubes with different diameters to assess how specimen geometry affects composite materials' capacity to absorb energy. The coupons' energy absorption was influenced by both the laminate thickness and the radius of curvature [83, 84]. Likewise, for conical specimens, different sizes were tested to assess the influence of inclination on the failure modes and energy absorption capabilities [20].

2.6 Geometric Effects on Crush Stress

As was mentioned in Section 2.5, the crush stress is not exclusively a material property. Experiments in the literature on closed and open-section coupons have found geometrical characteristics that affect crush stress and energy absorption. Understanding what factors influence the crush stress is essential to develop the analytical tool.

This section will explore the most prevalent geometrical influences on crush stress: the laminate's curvature, inclination relative to the crushing axis, strain rate, and thickness.

2.6.1 Curvature

Moreira [8] and Lobão [44] studied the influence of curvature of the laminate in the crush stress by testing multiple specimens with different sizes of CFRP, flax fibre reinforced polymers (FFRP) and hybrid composite circular tubes.

In the study conducted by Lobão [44], five distinct diameters for CFRP circular tubes were tested quasi-statically, with three specimens tested for each size. The results can be seen in Figure 2.10 and showcase a decrease of crush stress with the increase in radius of curvature, tending asymptotically to the crush stress value of a flat coupon (dashed line in Figure 2.10), which is equivalent to a tube with an infinite radius of curvature. However, it should be noted that tubes with a bigger radius should be tested to confirm that the crush stress approaches the value for flat coupons when the radius tends to infinity.

Moreira [8] also tested five distinct sizes of CFRP, FFRP and hybrid tubes, whose results are displayed in Figure 2.11 and Figure 2.12 and are in agreement with the results obtained by Lobão [44]. Although the author also tested FFRP tubes, the smaller tube sizes buckled, invalidating those experimental results for crush stress analysis since progressive crushing was not achieved.

It is not yet known why crush stress decreases with the radius of curvature of tubular specimens,

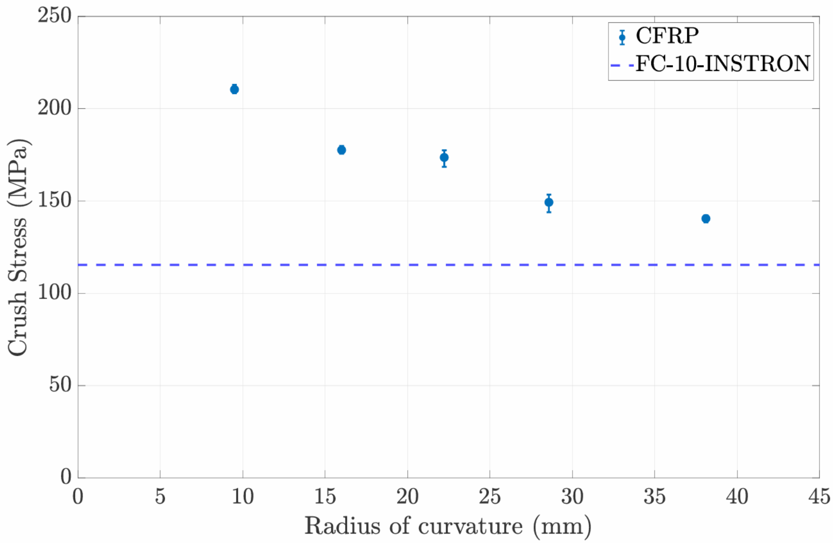


Figure 2.10: Effect of curvature of the CFRP tubular coupons on the crush stress [44]

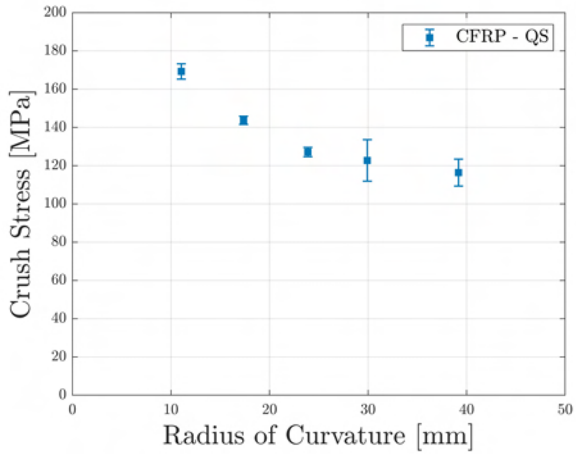


Figure 2.11: Curvature effects on CFRP circular tubes [8]

but Moreira [8] and Lobão [44] made some observations from the post-crushed CFRP specimens (see Figure 2.13) in an attempt to justify this phenomenon. The authors observed that in the smaller tube sizes (R1 and R2), the width of the fronds was smaller than in bigger tube sizes (R3, R4 and R5), indicating that more fibre breakage is needed to splay those specimens and, therefore, higher specific energy absorption. Additionally, on the smaller specimens, the internal fronds contact each other, generating friction, and more shearing seems to occur.

In their investigation of the crashworthiness of a composite Side Impact Structure (SIS) for a Formula 1 car, Dalli et al. [23] also confirmed the impact of curvature on crush stress and energy absorption. The stress-displacement graph of the structure was calculated using the load-displacement curve obtained during quasi-static progressive crushing. The graph showed a progressive decrease in crushing stress, with the top of the SIS having a smaller radius of curvature and a crush stress of 140–180 MPa. The area nearer the base had an increased average radius due to the inclusion of two flat regions, and the crush stress in that region ranged from 100 to 120 MPa.

This effect, however, varies greatly depending on the material of the crushed structure and is not always present. Yan and Chouw [19] observed during the crush testing of flax fibre-reinforced

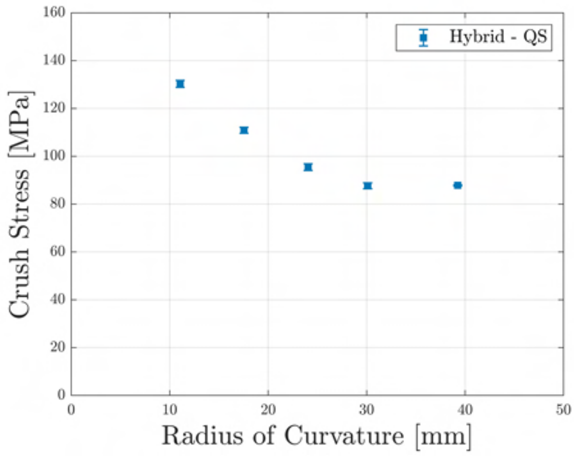


Figure 2.12: Curvature effects on hybrid circular tubes [8]

epoxy tubes (FFRP) that an increase in section diameter on single-ply laminate coupons did not significantly affect the peak load, and that even with two or three-ply laminate specimens, there was no discernible relationship between peak load and diameter. Lobão [44] noted a considerably less pronounced pattern of decreasing SEA and crush stress with increasing diameter on the crushed FFRP coupons.

2.6.2 Inclination

The crushing of conical shell specimens can be used to understand how crush stress is affected by the inclination of the laminate relative to the crushing axis. Research by Mamalis et al. [50] observed a gradual increase in crush stress for values of inclination ranging from $\theta = 5°$ to $\theta = 15°$. For inclination values equal to 20°, the author found a critical decrease in specific energy absorption and failure due to geometric instability for higher inclination angles.

The decrease in crush stress can be attributed to changes in the crushing failure modes as the inclination angle increases. For CFRP circular tubes (0° inclination), the predominant failure mode displayed was symmetrical splaying [8, 18, 44], whilst conical specimens with angles ranging from 5° to 15° displayed asymmetrical splaying and, for angles around 20°, inward bending of the laminate occurred [50] (see Figure 2.14) which drastically decreased the energy absorption capability of the structure.

In his work, Moreira [8] quasi-statically tested CFRP cones with distinct values of inclination. The stress-displacement curves were obtained by dividing the force at a particular displacement by the conical cross-section. The resultant stress-displacement plots can be seen in Figure 2.15, and the utilised notation of C-C-QI-2.5-20.5 means a cone made of a QI CFRP laminate, tested quasi-statically, with 2.5 degrees of inclination and 20.5 mm top internal diameter. The stress curves show a gradual decrease in crush stress with the displacement due to the increase in the radius of curvature as the cone is progressively crushed. Moreover, there is an apparent overall reduction in crush stress with inclination and a drastic decrease once the inclination reaches 20°. It should be noted, though, that only one conical specimen per inclination was tested.

2.6.3 Laminate Thickness

Laminate thickness, which is directly proportional to the number of plies in the layup, influences the energy absorption of a structure. Thicker specimens absorb more energy during crushing when

(a) T-C-QI-R1 crushed tube.

(b) T-C-QI-R2 crushed tube.

(c) T-C-QI-R3 crushed tube.

(d) T-C-QI-R4 crushed tube.

(e) T-C-QI-R5 crushed tube.

Figure 2.13: Diffent radii tubular coupons crushed quasi-statically [8] (R1=19.6 mm, R2=32.2 mm, R3=45.3 mm, R4=57.2 mm, R5=75.8 mm)

evaluated quasi-statically and dynamically, according to Liu et al. [84]. As demonstrated in Figure 2.16, under quasi-static loading, an increase in thickness directly increases the peak and average crush loads, which equates to increased energy absorption capacities.

Furthermore, Liu et al. [84] concluded that the force resisted by the specimens was not proportional to the increase in the cross-sectional area by the addition of plies to the laminate, suggesting that the laminate thickness is directly affecting the crush stress of the structure. A similar result was seen when employing square tube specimens; Mamalis et al. [46] reported that, for the same length and aspect ratio, peak load increases as tube wall thickness increases. They could not, however, come at the same findings on SEA and the overall absorbed crash energy.

Yan and Chouw [19] found that, for coupons with the same inner diameter and height/diameter ratio, the peak force increased with the number of plies during the crush tests of flax fibre-reinforced epoxy tubes (FFRP). This increase was proportionate for lower diameters but only marginally increased peak force for bigger diameters due to the additional plies.

2.6.4 Strain Rate

Several authors have conducted dynamic tests on open and closed section coupons to assess the effect of strain rate on the specimen's SEA. There are contradicting findings in the literature regarding how dynamic crushing affects composite materials' ability to absorb energy. In comparison to quasi-static testing, the measured specific energy absorption of various coupons rose [46, 50, 83, 85], reduced [8, 44, 64, 79, 86–92], or stayed constant [83, 93, 94] during dynamic

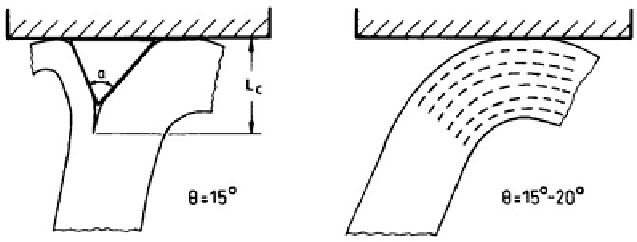


Figure 2.14: Changes in failure mode with inclination angle (adapted from [20]).

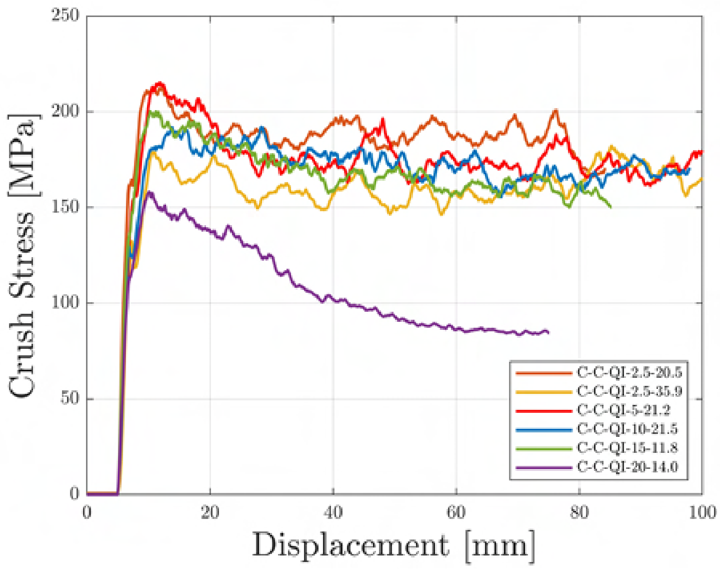


Figure 2.15: Stress-displacement curves from quasi-static tested conical CFRP coupons [8].

loading. The authors' contradictory findings indicate that the dynamic response heavily depends on the composite material chosen, mainly on the fibre's strain rate response.

Dynamic tests conducted by Dalli [18] on tubular CFRP tubes showed an undulatory response in the force-displacement graphs due to the specimens being fixed to the testing plate, which created mechanical vibrations that made the assessment of the crush stress difficult. Further work by Moreira [8], who developed a novel fixture for dynamic testing of tubular coupons, resolved the undulations and showed a performance drop of roughly 11% across all the tube sizes for initial crushing velocities of 8.5 m/s (see Figure 2.17). The author noted that in the stress-displacement graphs of the specimens, there was a slight increase in crush stress along the displacement, which can be attributed to the decrease in the velocity of the crushing head during the duration of the test.

Moreira [8] also dynamically tested FFRP tubular specimens of various sizes, but no conclusions on the strain rate effects could be made since most of the quasi-statically tested FFRP specimens failed through buckling.

2.7 Crashworthiness Modelling and Simulation

Testing high-performance materials can be pretty costly. For instance, an F1 front crash structure can have workforce and material costs of up to £20,000. It is, therefore, impractical to test such a structure so frequently [18]. Several computational methodologies have been used to repli-

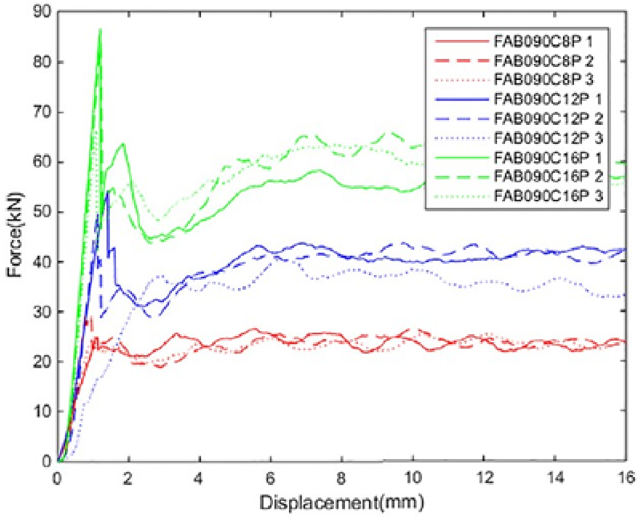


Figure 2.16: Force-displacement curves of crush tests on tube specimens with 8, 12 and 16 plies, under quasi-static loading conditions (adapted from [78]).

cate the behaviour of composite materials under crush loading. Namely, numerical models based on the Finite Element Model (FEM) and analytical models will be explored in this section.

2.7.1 Numerical Models

Composite structures can be simulated using a variety of techniques: micro-scale, in which the matrix and reinforcement are separately considered; mesoscale, in which the ply properties are calculated using the properties of the matrix and reinforcement, and the laminate is viewed as a stack of plies; and macro-scale, in which the laminate properties are determined using the ply properties, orientation, stacking sequence, etc. In crashworthiness simulations, mesoscale and macroscale models are most frequently utilised. Some authors used solid elements and cohesive surfaces to depict interlaminar failure in mesoscale simulations [95–97]. Others used shell elements to represent the entire laminate in macroscale simulations[98, 99].

Solid elements should be used if out-of-plane stresses are significant, but 2D shell elements are usually preferred due to lower computational times [100]. The laminate layers can be represented in shell elements by placing integration points in the thickness direction.

Solid components can be made of layers of distinct materials, such as when modelling solid composite structures where each ply is treated separately. Alternatively, solid elements can be made of a single material. Typically, solid element simulations consist of isoparametric quadrilateral and hexahedral elements, which are less computationally expensive and permit seamless stress transitions between the elements. These elements can be applied in various ways, including for sophisticated non-linear and linear simulations incorporating contact, plasticity, and large deformations [101, 102].

The outputs at the elements can be calculated using full or reduced integration for both solid and shell elements. Reduced integration utilises fewer Gaussian integration points when calculating the stiffness matrix than those required for obtaining the exact results, which cuts CPU time and prevents shear locking. First-order elements with reduced integration are advised when bending or significant deformations are present, which is the case for crashworthiness models. However, using reduced integration can cause the hourglass effect and impact the simulation's accuracy

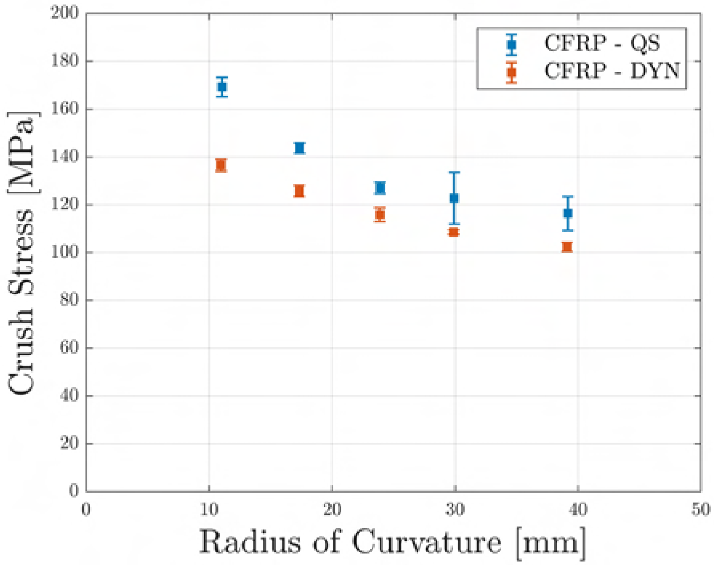


Figure 2.17: Comparison of the effect of curvature on the crush stress of the CFRP tubular coupons tested quasi-statically and dynamically [8].

[103].

Regarding solid elements, interlaminar failure can be included to simulate fracture mechanisms that account for a significant amount of energy absorption and, as a result, represent behaviour more similar to that found in actual specimens. In the case of shell elements, the entire composite laminae is simulated as a single structure, and this level of detail is lost.

Boria et al. [100] investigated the impact and accuracy of solid and shell modelling of an impact attenuator for formula student applications. The structure resembling a square frustum was first simulated using the LS-Dyna software. It was then produced using CFRP plain weave prepregs and tested in a drop tower. Mode-I and Mode-II fracture toughness were assessed using double cantilever beams and four-point end-notch flexure tests. In the solid model, cohesive elements were inserted between bundles of a specific number of plies to capture interlaminar failure, but in the shell model, nodes would be eliminated if all of an element's plies failed. The authors found that the solid and shell simulations could correctly forecast the structural collapse in terms of quantitative values compared to the experimental results; however, the solid simulation took 60 times longer to run.

Axial crushing of quasi-isotropic composite crush tubes was investigated in Reiner et al.'s [104] work utilising three distinct continuum damage mechanics models in Abaqus/Explicit and LS-DYNA. Excellent force-deflection and SEA predictions were made in that study. However, delamination could not be captured because a macroscale model of just one shell element was used through the thickness to represent the composite layup. By accounting for the interaction between the specimen and the test fixture, Wade et al. [105] obtained a high correlation between the force-displacement curves of crushed coupons when simulating the crushing of composite tubes using the single-layer approach. However, because it is unable to account for intra- and interlaminar fracture processes, the single-layer model is not the best for modelling the failure and crushing behaviour of coupons and crash structures, as noted by Siromani et al. [106]. Compared to multi-layer techniques, this approach's main attractive feature is its highly cheap computational cost.

Engenuity Ltd. created an add-on called CZone to enable robust simulations of composite materials for crashworthiness applications with the same level of predictability as in metallic structures. This was done to implement numerical models in commercial software, such as Abaqus. CZone requires the input of crush stress, a geometrical property, as seen in Section 2.6. A comprehensive set of experimental coupon crash tests was conducted by Dalli [18]. Using the CZone add-on, a crash structure, specifically an SIS from an F1 car, could be modelled by combining the crush stress observations from the experimental test program with the ply-level material characteristics. Using the methods depicted in Figure 2.18, the author manually assigned the crush stress parameter based on the outcomes of the crush tests conducted on the tubular and flat coupons. In this instance, the authors assigned the curved crush stress parameter to sections they believed to be curved and vice versa for parts they believed to be flat. The approach is susceptible to subjective interpretation due to the intricate geometry of the crash structure, which consists of several radii of curvature. It was determined that more research is required to comprehend how different curvatures affect the crush stress parameter. By assigning these parameters more precisely, the conformance between the outcomes of simulations and the actual behaviour of composite structures can be improved.

In summary, there are still discrepancies between the numerical models and physical behaviour produced by the present crashworthiness models, even though they can produce quantitative crash structure predictions at a very affordable computational cost. The macro-scale models' extreme simplification makes it impossible to represent important energy dissipation mechanisms, including contact friction, fibre breaking, matrix fracture propagation, and shear nonlinearity. Nevertheless, these simplifications are frequently required for the computationally efficient evaluation of complex composite structures [107]. But even in the macro-scale methods, more work must be done on the methodology. The behaviour of various geometries in crushing applications is poorly understood, making it challenging to generalise results from coupon-level testing to models of entire composite structures. Finally, current technology does not allow the implementation of other crush stress dependent parameters into the simulation, such as strain rate or inclination, and the end prediction is deterministic; that is, there is no uncertainty quantification to allow the assessment of the error incurred in the simulation.

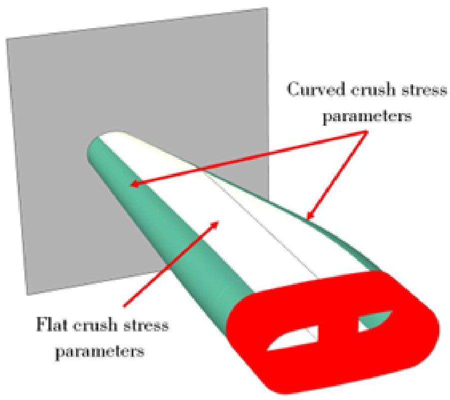


Figure 2.18: Subjective assignment of geometry-dependent energy absorption parameters in the model of a SIS (adapted from [18]).

2.7.2 Analytical Models

Several analytical techniques that involve significantly less computing work have been developed to yield energy absorption predictions of a given structure with significantly reduced computational effort. While numerical models can anticipate the force-displacement curves of more complicated geometries, these methods are limited to predicting specific crashworthiness metrics, such as peak force, mean force, and energy absorption. Additionally, these models assume that progressive failure occurs in the structure and cannot simulate failure due to elastic instabilities such as buckling.

Many analytical models produced calculated and experimental results that differed significantly from each other. For example, the models introduced by Boria et al. [49], which aimed to compute the mean crushing forces for conical and circular CFRP tubes, differed from the experimental values by 21% and 83%, respectively. Although most composite materials are brittle, some analytical models expected the material to be perfectly elastic-plastic [48]. Consequently, Hussein et al. [48] presented an analytical model that computed the mean force and partial energy absorption for square CFRP tubes under axial compression. The model produced good results, with the mean force discrepancy between the estimated and experimental values being just 7%. Other models, like the one for thin-walled fibreglass conical shells created by Mamalis et al. [50], demonstrated that accurate results for mean force and energy absorption can be obtained through analytical methods despite the simplifications made in modelling the failure modes.

Recently, an analytical tool developed by Mateus [1] attempted to correlate the geometrical influences described in Section 2.6 on crush stress to create an initial laminate design of composite crash structures. The tool took as input a pre-partitioned STL file of the structure and a material database derived from the experimental results of Dalli [18] and Moreira [8]. It then calculated the structure's curvature and inclination and correlated it to the material data to predict the crush stress. Having calculated the predicted crush stress, the force was then estimated and compared to a set target force, which was inputted by the user, and if the force was lower, then the number of plies was increased until the target was reached.

The overall developed tool showed promising results. However, the author did not compare the predictions with established experimental results, making judging the tool's accuracy difficult. Additionally, technical limitations of the code limited the possible geometries the tool could run to simple closed-section geometries, and open-section geometries or geometries with internal ribs could not be analysed. Finally, no study was made on uncertainty quantification.

2.8 Uncertainty Quantification in Crashworthiness

Uncertainty quantification is the process of evaluating and managing uncertainties in mathematical models and simulations. In the context of crashworthiness, it is crucial because traditional safety factors cannot be used as a reliable method to account for uncertainties. Crashworthiness design relies on accurate predictions of a structure's behaviour during impact, and using safety factors alone to account for uncertainties may lead to over-designing, that is, adding unnecessary weight and materials and potentially creating an overly stiff structure that may subject occupants to dangerously high decelerations. On the other hand, failing to account for the uncertainty might result in an under-designed structure that cannot withstand the impact forces [18]. Either outcome is unacceptable from an occupant safety standpoint and highlights the critical need for structural engineers to quantify the predictive error of simulations.

By identifying and quantifying the sources of uncertainty, engineers can make more informed decisions about design parameters and material selection to ensure the structure's performance and safety. In the case of crashworthiness, the current method for quantifying uncertainty lies within the experimental assessment of the crushing properties. After conducting coupon tests, the crashworthiness parameters described in Section 2.1 are calculated for each individual test. By averaging the calculated crashworthiness parameters from coupons with a particular geometry, the crush response of that structure is defined, and standard deviations of the results can be calculated to obtain distributions describing the variation of the parameters [8, 18, 44].

However, some problems can be identified with the current approach for uncertainty quantification in crashworthiness. Firstly, as was seen in Section 2.6, crush stress is not exclusively a material property and is also affected by the crushing conditions and geometrical characteristics of the structure. Since coupons have a defined geometry, the experimentally obtained values for the crush stress are specific to that structure, material and testing conditions. Therefore, extrapolating experimental coupon level results into structural levels implies defining via coupon testing how the crush stress is affected by each factor of influence described in Section 2.6, which requires an extremely large amount of coupon testing. As a reference, testing standard ASTM D3039 [108], used for determining the tensile strength of composites, recommends testing a number of specimens large enough to establish statistical significance, typically around 18 to 30 specimens for design allowables [109]. A complete uncertainty quantification analysis for crush stress would involve crush testing 18 to 30 specimens of each tube size and for each testing condition. Considering the high cost of manufacturing composite tubes and cones, such analysis would be very expensive, time-consuming, and laborious.

Another problem identified is that the crush stress and average forces are calculated by averaging all the values along a coupon's force-displacement curve. Other properties, such as the ultimate strength, are specific points in the force-displacement/stress-strain curves and can be determined by testing several specimens and averaging all specimen's ultimate strengths. Doing the same for the crush stress would mean averaging twice, once to calculate each specimen's crush stress and another to obtain a final value from all the tested specimens. Effectively, the specimen's crush stress is obtained by averaging a large number of data points in its plateau crushing region, and the final average value is a second average amongst all specimens' averages. Calculating the uncertainty bounds for crush stress using only the averaged value of each specimen suppresses the individual variation observed during the crushing response, which always follows some form of oscillatory response due to the continuously progressing crash front. It is possible for two specimens to have very distinct responses during crushing and have the same average stress, which might cause the final obtained value to be unrepresentative of the actual behaviour of all specimens.

The current macro-scale model used by industry for crashworthiness prediction is deterministic. These models rely on a set of fixed input parameters, such as material properties, geometry, and boundary conditions, to generate a single, predicted outcome. While this approach seems straightforward, it has limitations. Comparing a single prediction from a deterministic model with a single validation experiment offers a narrow view of the model's accuracy. This restricted evaluation fails to account for inherent uncertainties in the real world.

To the author's best knowledge, uncertainty quantification on composite crash structures based on numerical or analytical models has not been studied so far in the literature. However, Catalanotti [110] successfully employed Monte Carlo simulations (MCS) for uncertainty quantification of composites, which was previously unpractical due to computational limitations, demonstrating the potential of MCS in the field of composites. By randomly sampling input variables from their experimentally obtained distributions, Monte Carlo Simulations offer a well-established methodology for propagating uncertainty through complex numerical models. While mathematically robust and guaranteed to converge, the primary limitation of MCS lies in the computationally expensive nature of achieving convergence. Typically, $10^4 - 10^5$ simulations are necessary for basic convergence, and a thorough sensitivity analysis often necessitates $10^6 - 10^7$ runs [111]. This poses a significant challenge for intricate models where individual simulations require hours of computational time. Consequently, MCS can become computationally prohibitive for complex scenarios.

2.9 Concluding Remarks

The current state of the art has revealed critical limitations in the initial design phase of composite crash structures, particularly for industries like Formula 1 racing. While advancements in numerical modelling offer powerful simulation capabilities, a crucial gap exists in the foundational knowledge required to design these structures effectively. Current practices heavily rely on in-house expertise, which, while valuable, limits applicability to new scenarios and hinders broader innovation. This approach hinders progress and necessitates a shift towards a more standardized and data-driven approach.

Furthermore, the scarcity of experimental data on the influence of key geometrical parameters, such as radius of curvature, inclination angle, and thickness, on crashworthiness metrics like Specific Energy Absorption (SEA) and crush stress, creates a significant knowledge gap. This data is vital for enriching existing analytical and numerical models, ultimately leading to more accurate predictions of a crash structure's energy absorption capabilities. Engineers can refine their design strategies and optimize crashworthiness performance by systematically collecting and analysing experimental coupon-level data.

While computationally efficient, existing analytical models suffer from limitations that hinder their broader use. Their complex input parameters are highly specific to particular crushing scenarios, reducing the model's flexibility and adaptability. Additionally, they are often restricted to analyzing simple geometries, further limiting their practical applications. Therefore, a pressing need exists for a novel analytical tool that bridges the gap between the high accuracy (but computational intensity) of numerical models and the rudimentary reliance on in-house experience. This tool should prioritize speed and ease of use in the initial design stage. Ideally, it would incorporate key geometrical parameters, material properties, and loading conditions to provide a quick yet insightful initial design. While it may not achieve the same level of detail as numerical models, this tool would significantly improve upon current design practices by offering a standardized, data-driven approach readily applicable across diverse crushing scenarios. This advancement would not only accelerate the design process but also unlock new possibilities for optimizing crashworthiness performance in composite structures.

Chapter 3

Analytical Tool for Composite Crash Structures Design

The quick development cycles in racing competition, particularly in competitive settings with high stakes like Formula 1, demand precision and speed in the design and testing of safety-critical elements like crash structures. This industry has completely changed with the advent of composite materials, especially CFRPs, which offer exceptional strength-to-weight ratios but present particular difficulties in their design and analysis. Although several numerical models have been effectively created to mimic the behaviour of crash structures under quasi-static and dynamic crushing conditions, these models can take hours or days to produce results, making them unsuitable for the rapid performance predictions required in the initial design stages. It is necessary to create a tool that can be used in the early design phases to quickly estimate the energy absorption characteristics of a crashworthiness structure. This tool would use the currently available experimental data to either assess the energy absorption capabilities of a structure with a pre-defined layup or create a layup capable of withstanding a pre-defined target load.

3.1 Software, Programming Languages and File Formats

This section provides a basic understanding of the software, programming languages and file formats chosen to develop the analytical tool.

3.1.1 Autodesk Inventor

When Autodesk Inventor, a high-end computer-aided design (CAD) programme, was first developed, its primary goal was to give engineers and product designers a comprehensive set of tools for 3D mechanical design. With time, its features have grown to include not just building and modifying complex 3D models but also modelling their behaviour, producing drawings that are ready for production, and working simultaneously with a larger group of people. Parametric modelling, which enables users to specify features based on adjustable parameters, is the fundamental feature of Inventor. Parametric modelling saves time and lowers errors by creating a fluid design environment where modifications to one feature automatically ripple through the entire model. Autodesk Inventor is also very good at assembling models; users can construct intricate products by connecting distinct models and specifying how they work together. These components can be acquired from an extensive library of standard components or developed from scratch [112].

In this work, Autodesk Inventor will be used to craft the 3D models of the geometries that will be analysed by the analytical tool. The choice of this particular software is primarily due to licensing and the author's familiarity with it. However, it should be noted that other available CAD software could be used for the model design stage.

3.1.1.1 Visual Basic Macro Editor

The Visual Basic (VB) Macro Editor is a feature of Autodesk Inventor's extensive capability. It benefits power users who want to maximise workflows and expand the program's capabilities. This section dives into the nuances of the VB macro editor, examining its utility for expert users and its place in the Inventor ecosystem.

Visual Basic for Applications (VBA) is a well-known programming language used by the VB macro editor. Users who have previously worked with VBA can quickly transfer their expertise to Inventor, lowering the entrance barrier for automation by removing the requirement to learn a whole new programming language. The primary purpose of the VB macro editor is to automate Inventor's repetitious chores. Consider a situation where a particular set of actions must be taken on multiple components or assemblies. Users can significantly reduce the time spent on repetitive tasks by condensing these processes into a single command by constructing a VBA macro [113].

A macro in VBA will be developed to automate the process of partitioning the developed CAD geometry in regular intervals.

3.1.1.2 STL Files

3D Systems, Inc. created the STL file format, a CAD format used for 3D files, in 1988. It was created for STereoLithography (STL), a type of 3D printing, and it swiftly rose to prominence as the industry standard for fast prototyping in the ensuing ten years, continuing to be used today [114]. The three-dimensional CAD model's surfaces are discretised into triangle facets in a process commonly known as tessellation. The STL is a text file, written in ASCII or binary, containing information on every tesselated facet forming the structure, specifically, the global coordinates of the three vertexes that form a facet and a unit vector normal to the facet.

The user can choose the desired resolution of the tesselations. The choice of resolution affects the size of the tesselated facets, with higher resolutions having smaller facets and a higher overall number of facets in the file, increasing its size. The higher the resolution, the more accurately the tessellated surface resembles the original 3D model. Autodesk Inventor also allows a unique definition denominated as "Brep" or Boundary Representation, which guarantees that the tesselated model contains all topological components (faces, edges and vertices) and the connections between them, which is particularly useful to represent features that do not change the overall 3D model such as surface splits. Mateus [1] studied the impact that resolution and Brep have on the geometry and found that utilising Brep over the highest resolution offered incurs, at worst, an error of 1.04%.

After being partitioned by the VBA macro, the CAD geometries will be saved as an STL file with the Brep resolution enabled to capture the splits in the structure.

3.1.2 MATLAB

Matrix Laboratory (Matlab) is a high-level programming language developed by MathWorks for engineering and scientific applications. Its matrix-based language allows users to easily manipulate matrices and arrays, making it ideal for heavy numerical computations, data analysis, visualisation, and simulations. Extensive libraries of theme-specific functions and applications, commonly called Toolboxes, are available for download to all users with a MATLAB license. They further enhance the language's functionality by allowing users to integrate their code with powerful pre-developed features without the user having deep coding knowledge. Toolboxes cover various domains, including neural networks, statistical analysis, control systems, symbolic operations and curve fitting.

Mathworks also provides an integrated development environment (IDE) for algorithm development that allows the writing, execution and debugging of the code simply and effectively and is accessible to use and understand even if the user has little programming experience, further enhancing the accessibility of the language. It is also paired with tools to create graphical user interfaces and its proprietary data visualisation environment. Mathworks guarantees backwards compatibility with every new version of Matlab, which is vital since it ensures that a script will run in any of Matlab's future versions. All these features make MATLAB a popular development language in the industry and academia.

The scripts in this work will be developed in Matlab, except the VBA macro, due to the above-mentioned reasons and the author's familiarity with the language. Additionally, some toolboxes will be utilised to aid the import of STL files into Matlab, output results into Paraview, algebraic calculations and curve data fitting.

3.1.3 Paraview

Paraview is a free and open-source software for visualising scientific data. It excels at interactive, exploratory visualisation, allowing scientists and engineers to understand complex datasets through graphical representations. Paraview handles various data formats, supporting observational data (like weather patterns) and numerical data (from simulations). Its user-friendly interface offers tools for filtering, transforming, and manipulating data to create informative visualisations like 2D plots, 3D geometric objects, and animations. Paraview also boasts features for remote visualisation over networks and can handle large datasets efficiently for smooth exploration.

VTK (Visualization ToolKit) files are a native format for Paraview, making them ideal for data exchange and visualisation. These files store scientific data in a flexible format, including points, cells, and associated data arrays. Paraview can handle ASCII (text-based) and binary VTK files, with binary formats offering faster loading times. VTK files can represent various data types, from simple points in space to complex 3D objects with intricate properties. Paraview's integration with VTK allows seamless visualisation workflows, enabling users to directly load, manipulate, and visualise their scientific data for practical analysis and communication.

The outputs of the analytical tool's simulations will be exported to Paraview for interactive 3D visualisation via the VTK file format. Initial iterations of the tool had its outputs shown directly on Matlab; however, this approach was abandoned since the resulting figures were very computationally heavy and made data visualisation slow and cumbersome. In addition to a better viewing experience, the extensive customisation options available in Paraview allow for a more professional way of conveying the simulation's results.

3.2 Analytical Tool Overview

The analytical tool consists of three distinct scripts that must be executed before a first simulation may be attempted:

- `PartitioningTool.vba macro:` After the CAD model of the desired structure is constructed, the macro containing the VBA partitioning tool needs to be run. The macro will prompt the user to specify along which axis the structure needs to be partitioned, the distance between subsequent partitions and what surfaces need partitioning. After all the parameters are set, the tool automatically partitions the CAD model to match the inputs. Once the process is finished, the partitioned CAD model can be saved as an STL with the Brep resolution and can be inputted to the analytical tool.
- `MaterialCharectirization.m:` A developed Matlab script focusing on fitting normalised crush stress parameter curves from the experimental data and bootstrapping the data to obtain confidence bounds for uncertainty quantification. The script offers several input options to customise the fitting options, bootstrapping configuration and the script's outputs. After the execution, excel files will be written containing the fitted curves and confidence bounds plotting parameters that will be used as input by the analytical tool.
- `AnalyticalTool.m:` Main Matlab script that inputs a partitioned STL model and the normalised crush stress parameter curves obtainable from the two previous scripts. The developed tool aims to predict the force-displacement curve of a thin-walled composite structure being crushed axially. The program takes the vertexes on the inputted Brep STL file and calculates the inclination, curvature and, if it is a dynamic test, the velocity in all the vertexes. It then correlates the inputted crush stress parameter curves with the calculated parameters. It assigns a value of crush stress to each vertex, which it then uses to calculate the force-displacement curve of the structure. The tool can assess a specified layup or calculate the optimal one to achieve a target load. There is also the option to choose between dynamic or static simulations. The flow chart in Figure 3.1 illustrates the correlations between the inputs and outputs of the several scripts of this tool.

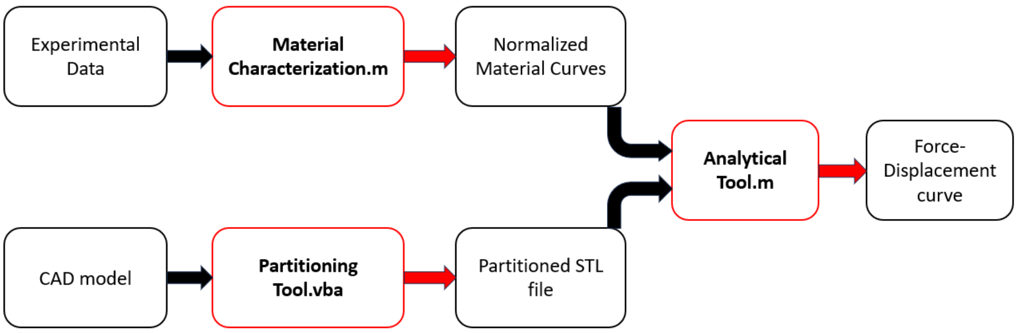


Figure 3.1: Flow chart of the analytical tool showcasing the interactions between the distinct scripts

It should be noted that some considerations need to be kept in mind when utilising the program. Firstly, the CAD structures must be constructed as zero-thickness or shell surfaces. Secondly, the structure must be drawn considering that the analytical tool will assume progressive crushing along the z-axis; that is, if the user wants to simulate the force-displacement curve of a tube being crushed axially, then the tube's axis in the CAD program must be parallel to the z-axis. Finally, the program assumes that the structure will be progressively crushed throughout its axis and does not consider the occurrence of other failure modes, such as buckling failure or bending failure. A FEM simulation and experimental testing must still be conducted on the final geometry to capture the possibility of the occurrence of geometric instabilities and certify it for crashworthiness purposes.

The developed program's purpose is not to replace established numerical models in predicting a structure's energy absorption but to aid engineers in designing new crash structures. The tool can obtain accurate predictions of the force-displacement graphs in just a few minutes of runtime, enabling engineers to iteratively make adjustments to the geometry and quickly get new predictions without the hassle and considerable time investment of setting up and running new successive FEM simulations each time a change is made. In the early design stages, engineers still rely on in-house knowledge and laborious manual assessment of the crush stress, utilising Excel for basic, subjective, deterministic and prone to human error predictions of the energy absorption. Overall, the tool offers a quick alternative to analytically predict the energy absorption of a thin-walled structure based on data from experimental coupon testing and provides confidence bounds for uncertainty quantification.

mi

3.3 Building and Partitioning of CAD Models

This section will address designing a CAD model compatible with the analytical tool and navigating the developed VBA partitioning macro. Some design considerations must be observed when constructing a CAD model for the analytical tool, namely:

1. The structure must be a "zero-thickness" or shell geometry.
2. The structure's axis must be parallel with the z-direction as the analytical tool assumes the z-axis as the crushing axis.
3. The CAD model must be partitioned at least along the z-axis utilising the partitioning tool. The resultant model must then be saved as an STL file with the Brep resolution enabled to capture the partitions in the geometry.

Additionally, since the tool assumes edge-wise progressive crushing of the laminate, any surfaces perpendicular to the z-axis (parallel to an XY plane) should be eliminated from the CAD model. Although these structures are often encountered in load-bearing elements that are also crash structures, such as the nose of an F1 car, they have little to no effect on the energy absorption during progressive crushing since the laminate is perpendicular to the crushing axis and breaks on impact [18]. The robustness of the analytical tool will allow it to run a complete simulation even if such a surface is present in the CAD model. However, the force absorbed in that region will not represent the real-world scenario, and therefore, the surface must be removed to obtain an accurate prediction.

Any geometry that adheres to the guidelines mentioned above can be utilised in the analytical tool, including open-section geometries, closed-section geometries and geometries with internal ribs or multiple closed sections.

Creating zero-thickness surfaces using Autodesk Inventor is similar to making regular geometries with thickness. The designer should construct a 2D sketch of the intended geometry and then utilise the *create* features present in the *3D Model* tab to obtain a 3D structure. The process differs once a create feature is selected; Autodesk Inventor opens a Properties window where the desired options can be chosen, and the designer must enable the option to create a surface, as is illustrated in Figure 3.2.

Once the zero-thickness model is finished, it is ready to be partitioned. The tool can partition the geometry in planes perpendicular to the z, y, and x-axis. Reference planes must be created for each direction the user wishes to partition. Partitioning along the z-direction requires the user

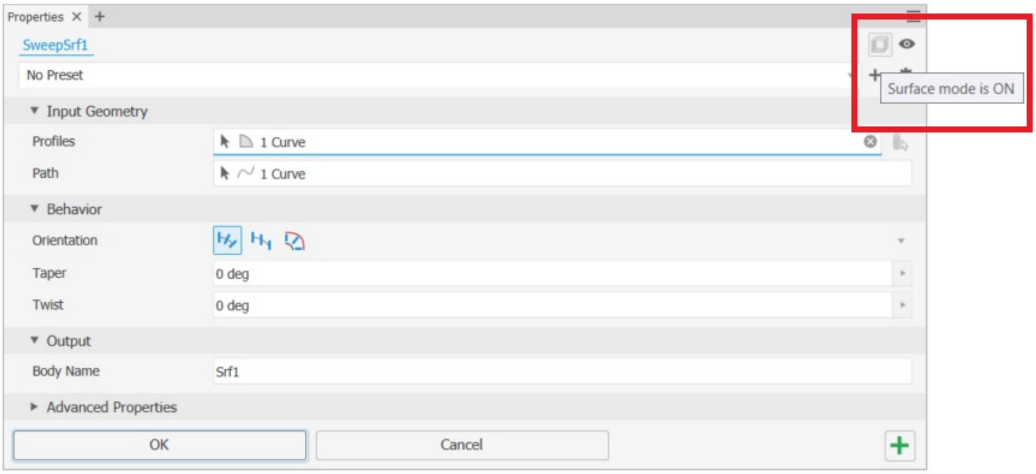


Figure 3.2: Autodesk properties window to sweep a sketch, with the option to create a surface highlighted

to create a plane perpendicular to the z-axis (XY plane) tangent to the point containing the most negative z-coordinate and rename the plane to "WP0_z". This process can be repeated along the x and y-directions if the user wishes to partition along those axes, and the created planes should be named "WP0_x" and "WP0_y", respectively. Recall that partitioning along the z-axis is required for the tool to function while partitioning along the x and y-axis is optional but can increase the final prediction's accuracy. It is worth mentioning that the *Measure* tool in the *Tools* tab can be used to aid with the creation of the planes.

Once the required planes are created, the user can then go to the *Tools* tab in Autodesk Inventor, under *Macros* and run the macro with the name `Partition_v2`. Upon doing so, a prompt asks the user if he desires to partition along the z-axis. If "Yes" is selected, a new prompt will appear for the user to enter the value in millimetres of the offset between subsequent partition planes. Afterwards, a window prompt will ask for the limit z coordinate, and the macro will partition the structure until the specified value is reached. A final prompt will then ask how many surfaces to partition and to pick those surfaces. This process is then repeated for the x and y directions. Only after all the prompts are answered does the tool partition the geometry.

Selecting "No" on the first prompt will automatically skip to the next direction. Alternatively, pressing "Cancel" will terminate the macro without making any partitions in case a mistake is made or the parameters need to be altered. It is recommended that the user writes down what parameters were chosen in the partitioning of the structure, especially the offset for the partitions in the z-axis, which is a parameter that needs to be inputted into the analytical tool in conjunction with the STL model.

Some care needs to be exercised when choosing a value for the offsets, as selecting a value that is too small will cause the STL file to contain excessive numbers of facets, increasing the analytical tool's runtime. If the chosen value is too large, the prediction becomes inaccurate, as the tool will not be able to capture changes in geometry finely. From experience with the tool, the optimal values for the offsets range from 3 mm to 10 mm, depending on the geometry.

After the tool runs, a message box instructs the user to delete undesired partitions. Sometimes, some of the automatic partitions may create local singularities and cause the tool to assign excessive values to the facet in question. If this behaviour is observed, the problematic partitions may be manually deleted. It should be noted that partitions along the z-axis may not be deleted, as doing so will cause an error in the analytical tool's code. Alternatively, the user could choose not to delete any partitions, and the experience shows that the final effect on the force-displacement curves is minimal since the elements that cause singularities are very small and, therefore, have little impact on the total force.

The analytical tool offers the option to simultaneously run a model in two distinct parts with different layups, for example, running the SIS's outer shell with an 8-ply laminate and the SIS's rib with a 4-ply laminate of the same material. To separate a model into two parts, the user must partition the whole CAD model using the partitioning tool macro and then alter the visibility of the surfaces only to show the surfaces that share the same laminate. Then, an STL file containing those surfaces can be saved. Afterwards, the visibility may be reversed to show the surfaces with the different laminate, and then that model can be exported as an STL.

During the development analytical tool's development, several CAD models were developed to test features, debug the code and for comparison with experimental results or FEM simulations. Models of tubes and cones were created to test the tool's ability to calculate the curvature and inclination. A tube with two internal ribs and a half-tube model was designed to test the tool's ability to handle multiple closed and open-section geometries. A pre-existing CAD model of the SIS of an F1 car was used for result verification, which was then separated into smaller parts to test the tool's ability to compute several parts with distinct laminates simultaneously, Figures 3.3 to 3.7 show some of the partitioned CAD models described.

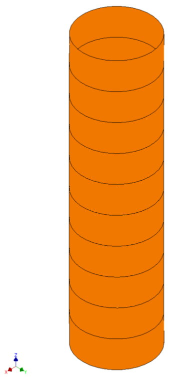


Figure 3.3: Partitioned tube CAD model

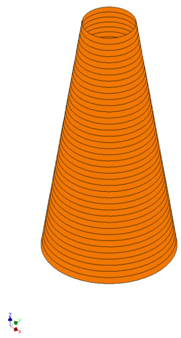


Figure 3.4: Partitioned cone CAD model

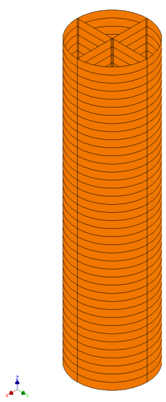


Figure 3.5: Partitioned tube with internal ribs CAD model

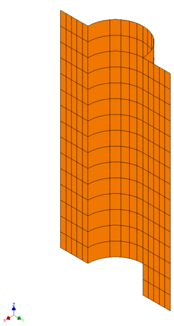


Figure 3.6: Partitioned half-tube CAD model

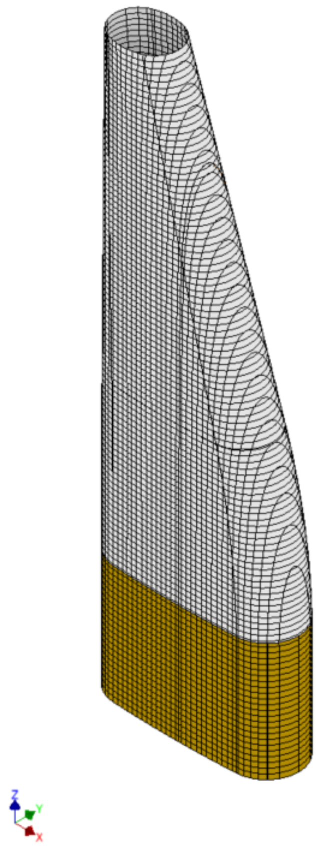


Figure 3.7: Partitioned SIS CAD model

3.4 Material Charecterization

The first iterations of the analytical tool had the fitting of the crush stress parameter curves from the experimental data integrated into the script. At those stages, the bootstrapping and uncertainty quantification still had not been developed, and it was feasible to have all the code in the same script. With the addition of uncertainty quantification in the form of bootstrapping, the code for fitting the curves grew dramatically in both complexity and runtime. The decision was then made to separate the code into two distinct scripts, one just for the analytical prediction and another exclusively dedicated to fitting and bootstrapping the experimentally obtained material data.

Developing an entirely new script allowed the inputs to be more complex and increased the overall customizability without overloading the user with too many inputs to run a single script. Additionally, having everything done in a single script caused the entire material characterisation process to be executed every time a new simulation was run, even if no changes were made to the material. Separating the scripts allowed the user only to run the material characterisation code when a new material or changes to an existing material are made, saving time and allowing the user to save output files to create a material library over time, allowing the user to test new materials quickly just by changing the input file in the analytical tool.

The finished `MaterialCharacterization.m` script takes as input four Excel files containing the experimental material data, specifically, the variation of crush stress with curvature, crushing velocity, inclination and laminate ply count. The available data is then fitted and bootstrapped to obtain confidence bounds, and four distinct Excel files are written, containing information on the normalised crush stress parameter curves for each of the four parameters of influence. The output files can then be directly used on the analytical tool script.

It should be noted that during the development of this work, no experimental data for the variation of the crush stress with inclination or laminate ply count was available for the materials studied. Due to this, the sections of code regarding the treatment of this data were left commented out to save computational time despite being fully functional. Theoretical curves based on available literature were utilised directly inside the analytical tool to mimic the effects of inclination and laminate thickness despite the lack of available experimental data.

It was mentioned before that the crush stress parameter curves would have to be normalised before inputting into the analytical tool script. This happens because the tool assigns a value for the crush stress at every facet vertex based on the calculated factors of influence described in Section 2.6. To assign a value, it must be known beforehand by what factor the crush stress increases or decreases when a singular parameter changes. Crush stress parameter curves can be normalised by dividing all the ordinate data by a particular reference value, effectively producing curves detailing by what factor the reference crush stress changes the parameter of influence changes. The reference state chosen for this work is an 8-ply flat-plate specimen crushed quasi-statically, meaning that the radius of curvature crush stress parameter curve will be normalised by the value of the crush stress when the radius tends to infinity (a flat plate is a curve with an infinite radius of curvature), the velocity plot by the crush stress value when $v = 0$ m/s, the inclination plot by the crush stress value when $\theta = 0°$ and the ply count plot by the crush stress value when the ply count equals 8.

Two additional features were coded in this script. When first importing new material data, choosing the best type of fit cannot be automated and must be manually determined by the user instead. To aid in this process, the option to intentionally stop the code midway through its operation was added to allow the user to use the Curve Fitter application, which is included in Matlab as part of the Curve Fitting Toolbox. Additionally, for results verification, an option was added to discretise the fitted radius of curvature vs crush stress parameter curve into bands of crush stress spanning different radii of curvature. This feature was added so that the crush stress parameter curve could be imported into an ABAQUS FEM model with the CZone plugging installed.

3.4.1 Input Control of the Material Characterization Script

Below is a list of all the user-defined inputs required to run the MaterialCharaterization.m script and a brief explanation of their function. It should be noted that the inputs "FilenameCurvature," "SampleNumberCurvature," "TypeFitCurvature," and "NormalizeCurvature" are repeated for the velocity, inclination, and ply count inputs, but their function remains the same.

- `FlagUnknownFit`: Set to False by default. If `FlagUnknownFit` is set to True, the code will not bootstrap but will stop midway through its execution. It will also provide tools for manually testing curve fitting types to the data using the Curve Fitter Application. Additionally, the following error message is displayed in the Command Window: "Code stopped prematurely because FlagUnknownFit==true. This is normal operation to allow the use of the Curve Fitter app. If the bootstrapping process is desired, set FlagUnknownFit to false and assign values to `TypeFitCurvature`, `TypeFitVelocity`, `TypeFitInclination` and `TypeFitPlyCount` as required.”

- `NumberBootstrap`: Set to 1000 by default. Number of times that curves will be fitted during the bootstrapping process of the material data. It can be set higher for more precision but will increase runtime.
- `ConfidenceBound`: Set to 0.95 by default. It can range from 0.01 to 0.99 and determines the confidence bound obtained after completing the bootstrapping process.
- `OutputToExcel`: Set to true by default. Determines whether or not to save the outputs into Excel files. If set to true, after the tool finished running, four Excel files named “DataCurvature.xlsx”, “DataVelocity.xlsx”, “DataInclination.xlsx”, “DataPlyCount.xlsx” will be created. The files will be overwritten if they already exist when running the tool.
- `PlotCurves`: Set to true by default. Determines whether or not figures containing the fitted and bootstrapped crush stress parameter curves are plotted.
- `FilenameCurvature`: Name of the Excel file and file extension containing the experimental data of the crush stress variation with the radius of curvature. The data’s format should be a table with two columns, with the first column containing the radius of curvature of the tested specimens and the second column containing the corresponding average crush stress. The file must be in the same folder as the script and remain closed during execution.

  Example: FilenameCurvature=’InputCurvatureHybrid.xlsx’
- `SamplesNumberCurvature`: Number of specimen samples tested per size for the radius of curvature data. For the bootstrapping process, the number of samples tested must be the same across all the tested sizes for the curvature data.

  Example: To obtain the experimental data for the variation of the crush stress with the radius of curvature, four different tube sizes with increasingly large radii were tested, and each size was tested five times. In this case, SamplesNumberCurvature=5. However, if one of the sizes were only tested three times, the results of two samples for each tube size would have to be removed, and SamplesNumberCurvature=3.
- `TypeFitCurvature`: Type of fit that best fits the available crush stress vs radius of curvature experimental data. If not known, set `FlagUnknownFit` to true and determine the best fit using the Curve Fitter Application.

  Example: TypeFitCurvature=’power2’
- `NormalizeCurvature`: Set to true by default. Determines whether or not to normalise the curvature data. The variable must be set to true if the outputted Excel files are to be used in the analytical tool script.
- `FlagDiscretizeCurvature`: Unique to the curvature data and is set to False as default. Determines whether or not to discretise the fitted curvature crush stress parameter curve into segments of a specified variation. Only to be used if the crush stress parameter curve is intended to be imported into an ABAQUS FEM model with the CZone plugin.
- `VariationDiscretization`: Only required if `FlagDiscretizeCurvature`=true and is set to 5 as default. Expressed in percentage (0-100), limits the discretisation bound by the variation specified.
- `DiscretizationPlotType`: Only required if `FlagDiscretizeCurvature`=true and is set to 0 as default. Determines which of the three fitted curves is discretised, where 0=Fit from

data, 1=Lower Bound Fit, 2=Upper Bound Fit.

- `BaseNumberPlies`: Unique to the ply count data. Number of plies of the specimens used to obtain data for the crush stress variation with radius of curvature, inclination and velocity. In this work was set to eight since all the tests were performed with eight-ply specimens. This variable determines what value for the crush stress is used to normalise the ply count data. Since it was set to eight for this work, the ply count data was normalised by the value of the crush stress when the ply count was eight.

Regarding the variable `SamplesNumberCurvature`, it is understood that possibly deleting good experimental data is not ideal, and the code could be altered to remove this limitation. However, doing so will also require the user to manually change data in the input files to match the new format, which might be laborious, time-consuming, and prone to human error. Due to this, the choice was made to keep the current format and ask the user to delete the extra samples in each set if necessary.

Finally, the program requires the following Toolboxes to be installed:

- Statistics and Machine Learning Toolbox
- Curve Fitting Toolbox
- Symbolic Math Toolbox

3.4.2 Using the Curve Fitting Application

As stated, the user could use the Curve Fitting application to find the best fit for the experimental data. To do so, the user should assign any value to the `TypeFit` variables so the script does not immediately produce an error due to a lack of an assigned variable. After the code breaks and the error message is displayed, the user can open the Curve Fitter program by going to the Apps section in Matlab's IDE.

Having opened the Curve Fitting application, the user can insert the required experimental data by going to the *Data* tab and selecting the *Select Data* button. After doing so, an input box prompts the user to choose the intended data. The script automatically creates vectors containing the x and y experimental data for each influence parameter to allow easy input onto the Curve Fitter application. Figure 3.8 shows the process of selecting the radius of curvature experimental data onto the prompt, and Figure 3.9 shows the data displayed in the application alongside a first-degree polynomial fit (poly1).

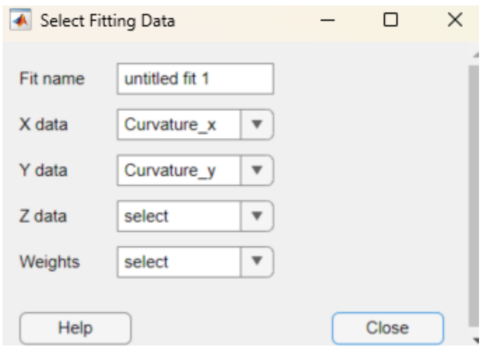


Figure 3.8: Input box showing the selected RoC experimental data

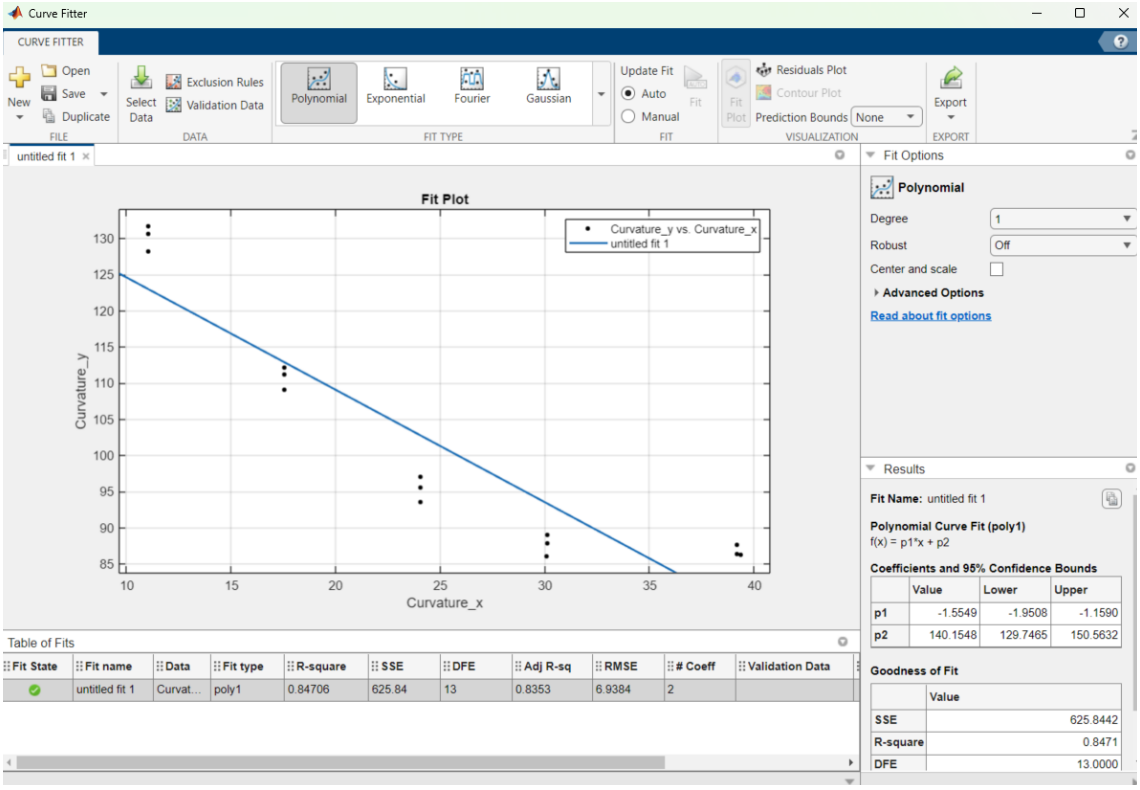


Figure 3.9: Curve Fitter app displaying the RoC experimental data and a poly1 fit

With the data selected, the user can try different fit options by choosing a different fit in the *Fit Type* tab. The current fit may also be modified by adjusting the options in the *Fit Options* tab (right side menu in Figure 3.9). Parameters defining the goodness of fit can also be found in the bottom right of the application (see Figure 3.9).

Supposing that particular data points are found to be outliers that are unrepresentative of the material's behaviour and negatively affect the fit's goodness. In that case, there is an option to temporarily exclude these data points to check for improvements. To do so, the user must open the experimental data vector and find the index of the problematic data points. Then, those indexes must be written into the appropriate exclusion vectors (`ExcldCurvature`, `ExcldVelocity`, `ExcldInclination` and `ExcldPlyCount`) found in the *Inputs for the Curve Fitter* code section, re-run the code and re-open and re-select the data onto the Curve Fitter application. If the goodness of fit improves, the user must open the relevant input file and delete the whole outlier set to maintain a constant number of samples throughout all the tested sizes.

To fit the inclination, velocity and ply count experimental data, it is a matter of finding the best-fitting function and inputting the fit type's designation into the relevant TypeFit parameter (`TypeFitVelocity`, `TypeFitInclination` and `TypeFitPlyCount`). The fit type designation can be found in the Curve Fitter's *Results* window inside brackets in front of the fit's name (see Figure 3.9).

Choosing a fit type for the radius of curvature experimental data offers an added consideration. In addition to selecting the best fit, the fitted curve should plateau to a value when the radius of curvature approaches infinity. This is because the curvature's data will be normalised by the crush stress value at the plateau to simulate a flat coupon. From experience, the only type of fit to meet both criteria is the *power2* fit, whose mathematical expression is given by Equation (3.1).

$$f(x) = a \cdot x^b + c \tag{3.1}$$

,where a,b and c are the fitting parameters. Interestingly, if the coefficient b is negative, $a \cdot x^b$ tends to zero when x tends to infinity, and the coefficient c corresponds to the plateau.

3.4.3 Fitting, Normalising and Bootstrapping the Experimental Material Data

This section describes how the `MaterialCharacterization.m` script obtains the crush stress parameter curves from the experimental data through fitting and bootstrapping. The script first reads the inputted Excel files specified in the `Filename` variables, and their data is stored in a matrix. The experimental data of each influence parameter is then fitted according to the `TypeFit` variables, and the normalising factors are obtained according to the established reference of an 8-ply flat specimen tested quasi-statically. All the experimental crush stress values are then divided by the correspondent normalising factor, and the new normalised data is fitted to get the parameters for the normalised curves.

For the specific case of the RoC crush stress parameter curve, the normalised factor is obtained differently. Since the RoC curve is fitted through a *power2* fit whose b coefficient is negative, the coefficient c is the crush stress value at the plateau. The normalising factor for the RoC was defined as $1.01c$, and the inverse function of the RoC crush stress parameter curve was used to obtain the radius of curvature at which $1.01c$ occurs. The normalising factor and the respective RoC are saved in the `DataCurvature` output file under the variable names `flat_CrushStress` and `flat_transition`, respectively, to be used in the analytical tool script. The remainder of the process is the same as above, where the crush stress values are divided by the normalising factor, and the data is refitted.

The data matrices must be rearranged before bootstrapping the material's experimental data can be started. To this effect, the script separates the available data for each influence parameter into groups of similar x to provide the required sampling sets for bootstrapping. The x and y data are stored in the group's sampling matrix.

With the required data prepared, the bootstrapping process is then started. The script randomly picks several samples from each available sampling group equal to the value specified in `SamplesNumber` input variables. Subsequently, a curve is fitted to the sampled data with the same fit type as the one specified in the `TypeFit` variables, and the fitting coefficients are saved in a matrix. The described operation corresponds to a single bootstrapped curve, and this process is repeated until the number of bootstrapped curves matches the amount specified in the $NumberBootstrap$ input. It should be noted that sampling picking is done with the repositioning of values, meaning that the same data point may be picked more than once. Additionally, the RNG seed in the script is fixed to allow for the replication of results.

Once all the necessary curves have been bootstrapped, the next step is to order them and find the curves corresponding to the specified confidence interval. The matrix containing all the coefficients describing the bootstrapped curves is ordered based on the coefficient of most relevance, which itself is dependent on the chosen fit type. A function denominated as `GetType.m` was created to automate this process, containing information about the ordering parameter and the total number of parameters for several data fits, specifically *poly1* to *poly4*, *exp1*, *exp2*, *power1*, and *power2* fits. After the matrix is ordered, the confidence bounds can be obtained in the following manner, assuming that a 95% confidence interval was specified in the `ConfidenceBound` input: the script separates the central 95% of the matrix's rows, leaving out the first 2.5% of rows and the last 2.5%. Then, the bootstrapped curves corresponding to the confidence interval are defined as the first and last curves of the matrix containing the central rows.

The confidence bounds fitting coefficients are then saved into an Excel file alongside the fitting type and the fitting coefficients obtained from directly fitting the experimental data. For the specific case of the RoC data, the Excel file also includes the variables `flat_transition` and `flat_CrushStress`.

3.4.4 Curvature Data Discretization

The code for the curvature data discretisation was created to aid the import of the RoC data to an ABAQUS FEM model, which was used for verification purposes by running numerical simulations with the same material parameters as the ones used for the analytical tool. The code creates a text file with crush stress values assigned to bounds of radii of curvature. To use the discretisation function, the input variable `FlagDiscretizeCurvature` must be set to true and `NormalizeCurvature` must be false.

The code for creating bounds defines a crush stress limit according to the specified input `VariationDiscretization`. The limit is defined according to Equation (3.2).

$$\texttt{Limit} = \texttt{flat\_CrushStress} \times (\texttt{VariationDiscretization} \times 0.01) \tag{3.2}$$

The code starts analysing the curve at the `flat_transition` x-coordinate, whose corresponding y-coordinate is `flat_CrushStress`. Once the crush stress of the crush stress parameter curve exceeds the value of `flat_CrushStress+Limit`, a boundary is created there, and the corresponding crush stress is saved in the `CrushIntervals` variable. The code then looks for the coordinates where `CrushIntervals+Limit` is exceeded, creates a new boundary and saves the new crush stress in the `CrushIntervals` variable. The described process is repeated until the whole curve is discretised.

An additional feature had to be coded in the discretisation process. ABAQUS only allows the boundary x-coordinates to be integers, meaning that a bound must be at least a unit wide. This is not an issue for large values of x since the crush stress parameter curve there decreases slowly, but for small values of x, the rapid decrease would create bounds less than a unit wide. Code was developed that would join together adjacent bounds until a bound at least a unit wide could be formed. A final bound was created from x=`flat_CrushStress` to infinity to accommodate the assignment of crush stress values to straight sections.

3.4.5 Crush Stress Parameter Curves and Convergence Analysis

The experimental results from Moreira [8] were utilised as inputs for the `MaterialCharacterization.m` script. Moreira [8] studied the behaviour of CFRP and hybrid FFRP/CFRP tubular coupons quasi-statically and dynamically, allowing the creation of crush stress vs RoC and crush stress vs crushing velocity crush stress parameter curves with recourse to the developed script.

Insufficient experimental data existed to accurately assess the effects of inclination and laminate thickness, so theoretical curves were utilised instead to simulate those factors' effects. Further experimentation with conical coupons and coupons with different thicknesses is required before enough data can be inputted into the script.

3.4.5.1 Carbon Fibre Crush Stress Parameter Curves

During his studies, Moreira [8] utilised coupons made of a CFRP pre-preg with E745 epoxy resin and IM7 carbon fibres from supplied by the company Toray®. The plies featured a balanced 5 Harness-Satin weave (5HS) with an areal weight of 280 g/m^2. It should be noted that since the weaves were balanced, the ply's mechanical properties were similar in the transverse and longitudinal directions. The tubular coupons were made with an 8-ply laminate with a $[45°/0°/45°/0°]_S$ stacking sequence. Table 3.1 shows the dimensions of all tubes manufactured, where D_i is the internal diameter and t is the thickness. From the eight manufactured sets for each tube size, five were tested quasi-statically, and the remaining three were tested dynamically.

Table 3.1: Dimensions of the manufactured CFRP tubes (Adapted from [8]).

Sample	Sets	D_i [mm]	t [mm]
T-C-QI-R1	8	19.57(±0.45)	2.47(±0.05)
T-C-QI-R2	8	32.21(±0.05)	2.50(±0.02)
T-C-QI-R3	8	45.28(±0.02)	2.51(±0.03)
T-C-QI-R4	8	57.29(±0.05)	2.52(±0.03)
T-C-QI-R5	8	75.81(±0.08)	2.51(±0.02)

The experimental data obtained from the tests performed by Moreira [8] can be found in Appendix A.1.1 already in the necessary format to be inputted into the `MaterialCharacterization.m` script. The process by which the tables were obtained from the available experimental data is also documented in Appendix A.1.1. Note that this process heavily depends on how the experimental data was obtained. Therefore, no single procedure can be created to adapt the data into the format that the script requires. The tool will run as long as the input formats detailed in Section 3.4.1 are respected.

Before obtaining the final crush stress parameter curves, it is first essential to know how many bootstrapped curves need to be done by the script. Therefore, the available RoC experimental data were bootstrapped 100, 500 and 1000 times, and the results can be observed in Figure 3.10. Additionally, Table 3.2 shows the total runtime for each bootstrapping process.

Table 3.2: Differences in runtime

Number of Bootstraps	Runtime [s]
100	4.01
500	13.8
1000	33.7

As can be seen from Figure 3.10, the difference in the confidence bounds obtained when varying the number of bootstraps made is very slight, and from Table 3.2, it can be seen that the runtime steadily increases with the number of bootstrapped curves. However, since the overall runtimes are fast, the choice was made to use 1000 bootstrapped curves for all further material characterisation to guarantee the most accurate results whilst maintaining reasonable runtimes. Table A.3 in Appendix A.1.2 contains the coefficients resulting from the 100 bootstrapped RoC curves ordered by the c parameter for completeness.

Having confirmed the suitability of the bootstrapping process, the final crush stress parameter curves for the IM7-E745 5HS CFRP laminate, which show the variation of crush stress with radius of curvature and impact velocity, can be obtained. Figure 3.11 and 3.12 show the respective curves. Documented in Appendix A.1.3 can be found those same curves with the values normalised as well

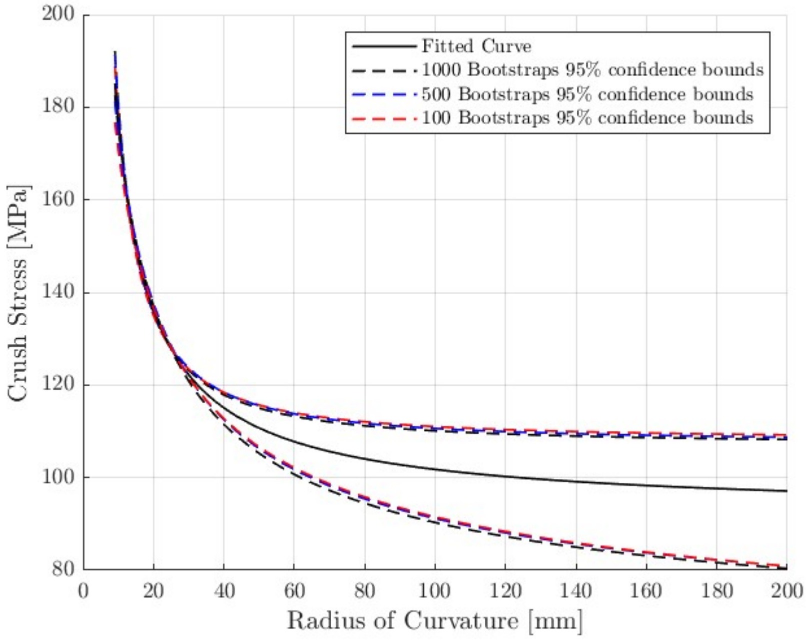


Figure 3.10: Comparison in confidence bounds with 1000, 500 and 100 bootstraps

as the information contained in the `DataCurvature` and `DataVelocity` output files.

Finally, Figure 3.13 documents the discretised RoC crush stress parameter curve that will be inputted into FEM simulation for results verification. Table A.6 found on Appendix A.1.4 contains the bounds and corresponding assigned values of crush stress. That same data can be found in the outputted text file from the discretisation process.

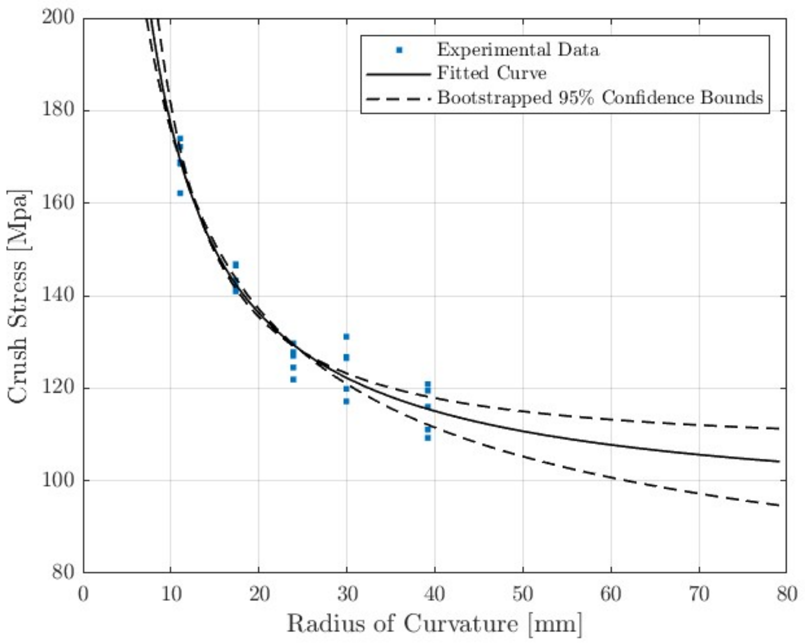


Figure 3.11: Variation of crush stress with radius of curvature for a CRFP laminate

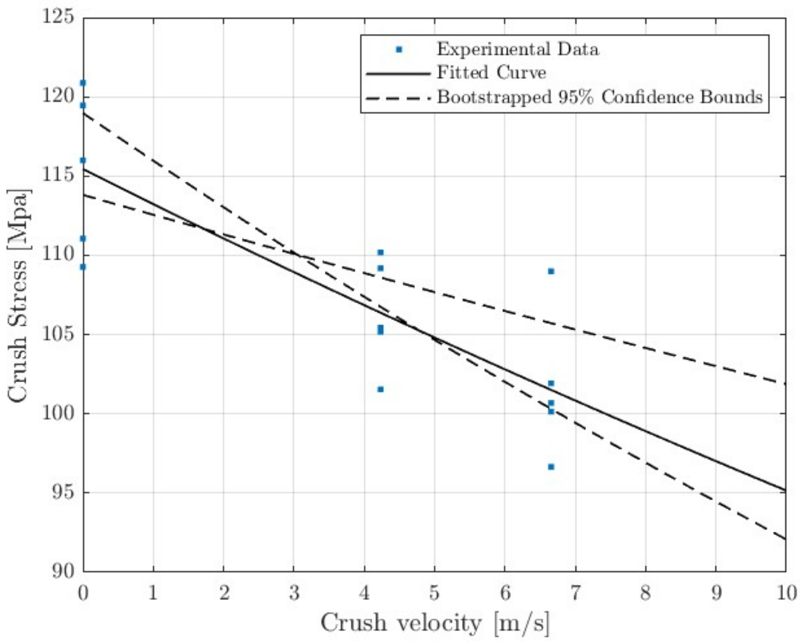


Figure 3.12: Variation of crush stress with crushing velocity for a CRFP laminate

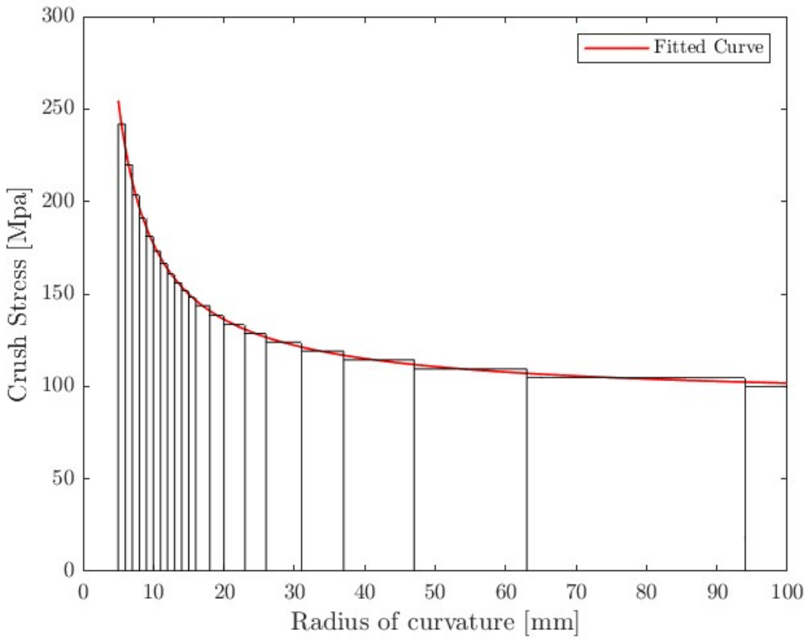


Figure 3.13: Discretized RoC curve for a CFRP laminate

3.4.5.2 Hybrid Crush Stress Parameter Curves

Moreira [8] also performed tests on hybrid tubular coupons whose data will be used to obtain the crush stress parameter curves through the utilisation of the `MaterialCharacterization.m` script. The laminate was a 50-50 hybrid between the IM7-E745 5HS CFRP material from the previous section and an FFRP material. The FFRP plies are made of an epoxy-based matrix supplied by Angeloni® with the designation IMP503Z-HT and the flax fibre reinforcements supplied by

Bcomp® with the designation APX 300T. The FFRP plies are arranged in a 2x2 balanced twill weave.

The hybrid laminate utilised in manufacturing the tubular coupons was an 8-ply laminate whose four outer plies were CFRP, and the four inner plies were FFRP. All the plies in the laminate were aligned along the 0° direction except the two outermost plies, which were aligned in the 45° direction, forming a $[45°/0°/0°/0°]_S$ layup. Table 3.3 documents the sizes of the manufactured hybrid laminate tubes.

Table 3.3: Dimensions of the manufactured hybrid tubes (Adapted from [8]).

Sample	Sets	D_i [mm]	t [mm]
T-H-50 CO B-R1	3	19.24 (±0.01)	2.89 (±0.01)
T-H-50 CO B-R2	3	32.27 (±0.01)	2.84 (±0.02)
T-H-50 CO B-R3	3	45.29 (±0.01)	2.84 (±0.01)
T-H-50 CO B-R4	3	57.37 (±0.04)	2.85 (±0.01)
T-H-50 CO B-R5	3	75.67 (±0.16)	2.83 (±0.01)

The experimental data obtained from the tests performed by Moreira [8] can be found in Appendix A.2.1 already in the necessary format to be inputted into the `MaterialCharacterization.m` script. The process of adapting the available data into the required input format by the script was the same as the one performed for the CFRP laminate and is detailed in Appendix A.1.1.

The data was inputted, and the script was used to bootstrap 1000 times to obtain the confidence bounds. Figures 3.14 and 3.15 show the absolute RoC and velocity crush stress parameter curves. In Appendix A.2.2, the normalised curves can be found, as well as the data contained on the `DataCurvature` and `DataVelocity` output files to be used in the analytical tool script.

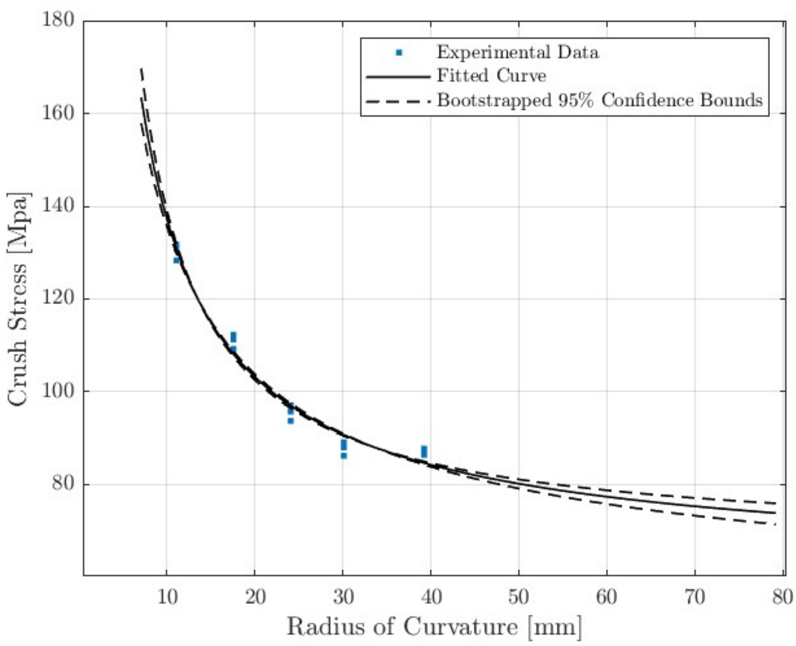


Figure 3.14: Variation of crush stress with RoC for a Hybrid laminate.

It can be seen from Figures 3.14 and 3.15 that the experimental data for each tube size was much less scattered than for the CFRP data, producing tighter confidence bounds. Additionally, since only three specimens per size were tested, the bootstrapping was less effective since fewer

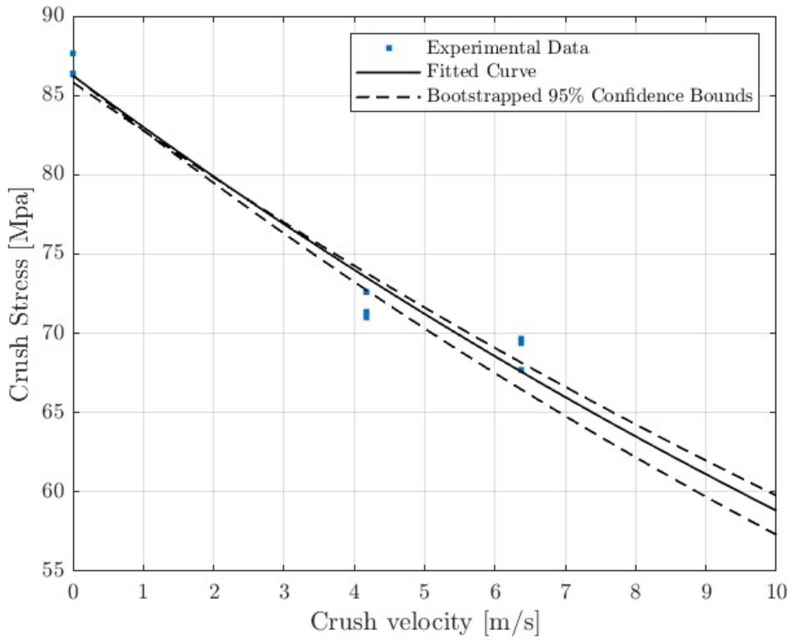


Figure 3.15: Variation of crush stress with crushing velocity for a Hybrid laminate.

total combinations of bootstrapped curves were possible. Additionally, for the RoC curve, the crush stresses were similar for the two largest tube sizes tested, indicating that a plateau was experimentally reached. However, the fitted curves continue to decrease past the data and eventually plateau at a value 40% lower than the experimental data suggests.

To force the appearance of a plateau reflecting the experimental data, the data for the R5 tube sizes was duplicated at $x = 80$ mm. This approach might not be very scientifically sound due to the lack of available experimental data for larger tube radii. However, it was done to simulate what might be observed once experimental data is available. The results section will compare the predicted force-displacement curves of a hybrid SIS obtained from the analytical tool for the two Hybrid RoC curves. Figure 3.16 shows the absolute curve obtained with the duplicated R5 data, and in Appendix A.2.3 can be found the normalised curves, the input file and the output `DataCurvature` file obtained.

3.4.5.3 Crush Stress Parameter Curves with Inclination and Thickness (Ply Count) Effects

This section documents the theoretical curves utilised to account for the lack of experimental data to obtain the fitted and bootstrapped curves from the `MaterialCharacterization.m` program.

The literature on the crushing of conical specimens reveals that inclination mainly affects the crush stress by changing the crushing mechanism from symmetric splaying to asymmetric splaying, eventually bending the laminate. It was clear from the tests conducted that up to 20° of inclination, the effect on crush stress was negligible and barely noticeable. However, for values larger than 20°, the laminate stops crushing progressively and starts to bend, causing the crush stress to drop drastically.

A normalized crush stress parameter curve for the variation of the crush stress with inclination was developed that translates the observations made from literature. The created curve is a branch

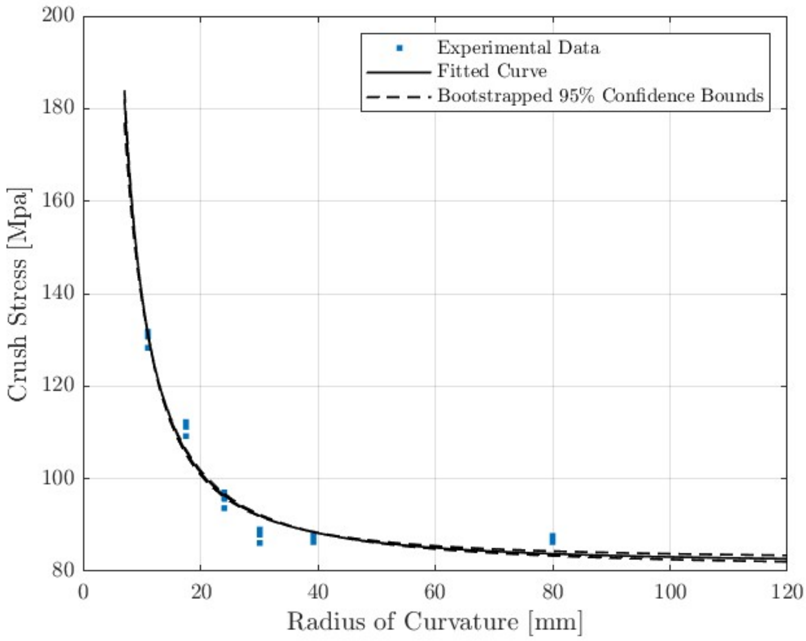


Figure 3.16: Absolute RoC curve with duplicated R5 values.

function described by Equation (3.3), which defines that until 20°, the inclination has no effect on crush stress, and for higher inclinations the performance drops by 1% per degree. Figure 3.17 shows the obtained crush stress parameter curve. Note, however, that no confidence bounds are provided since the curve is purely theoretical. Additionally, the curve was created directly in the `AnalyticalTool.m` script in a temporary section of code developed especially to consider the effect of inclination without experimental data being available.

$$\texttt{EqInclination}(\theta) = \begin{cases} 1 & , 0 \leq \theta \leq 20 \\ 1 - 0.01(\theta - 20) & , 20 \leq \theta \end{cases} \tag{3.3}$$

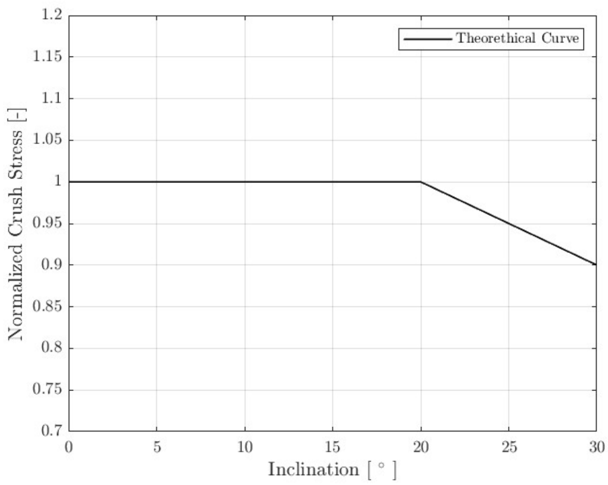


Figure 3.17: Normalised theoretical crush stress parameter curve for the inclination

The literature is not conclusive regarding the effect of laminate thickness or ply count on crush stress. It is generally understood that laminate thickness positively affects crush stress; however, it is not clear to what degree it influences it, and the overall effect appears to be minor. Further studies are required before a theoretical curve may be drawn, and therefore, the impact of thickness will not be considered in the predictions.

It should still be noted, however, that the analytical tool program was still developed to accept ply count experimental data, considering that once data is available, it can be inputted into the program and increase its accuracy.

3.5 Development of the Analytical Tool Script

The developed `AnalyticalTool.m` script aims to predict the force-displacement curve of a provided model by assigning values of crush stress based on crush stress parameter curves derived from experimental coupon data. Two main inputs are required before a simulation may be attempted: the STL model of the structure and the normalised crush stress parameter curves. The previous sections of this chapter were focused on the process of obtaining those inputs.

This section will focus on the script's workings, highlighting the process of obtaining an analytical prediction and the features implemented to broaden the tool's robustness and allow it to work correctly with the fewest possible limitations. To this effect, each primary process in the tool's code was developed in independent functions to allow for modulation, future upgrades and easy debugging. As long as the inputs and outputs of the functions composing the `AnalyticalTool.m` script are respected, the function might be freely changed to accommodate new requirements.

Following is a list, in operational order, of all the relevant functions found in the `AnalyticalTool.m` script and a brief description of their function.

- `stlread.m`: Function belonging to an older version of the Stlread Toolbox. The function can read STL files in ASCII and binary format and outputs three matrices: the V matrix, where each row contains the x, y and z global coordinates of a vertex; the F matrix, where each row contains the location in the matrix V of the three vertexes making up a facet; and the N matrix where each row contains the x, y and z coordinates of a unit vector normal to the facet. It was elected to use an outdated script version as the current version obtains the triangulation data in a format that would be cumbersome to adapt into useful data.
- `MaterialCurves.m`: Unpacks all the material data from the inputted Excel files, reconstructs the normalised crush stress parameter curves and stores them in a particular format to be used later
- `Treatment_v2.m`: The function filters and rearranges the data in the V, F and N matrices into a usable format. The function outputs two data structures. The `nodes` structure compiles all the available nodal data and has space allocated for further information to be added later. The `triangles` structure contains information on nodal connectivity and stores the required information for plotting the output variables. A `nodes` and `triangles` structure is created for every model inputted into the tool. If, for example, two parts forming a structure are inputted, two distinct `triangles` and `nodes` structures are created, one for each part. All the created `nodes` structures are stored inside the `partNodes` structure and the `triangles` structures inside the `partTriangles` structure.
- `geometry.m`: This script runs all the nodes present in the `nodes` structure and calculates the local inclination, radius of curvature and distance to adjacent nodes at all nodes. The calculated parameters are then stored in the `nodes` structure.

- `CrushStress.m`: Assigns to each node in the nodes structure a value for crush stress based on geometric parameters calculated and calculates the force-displacement curves and the prediction's confidence bounds. If the simulation is dynamic, the function performs the velocity update. If ply optimisation is desired, the script increases the local ply count until a specified target force is reached.
- `vtkwrite_mod.m` A modified version of the *vtkwrite* script included in the VTKwrite Toolbox. Allows the data to be exported in the VTK format for visualisation through Paraview.

After the tool runs, the program will produce several output files depending on the specified inputs. The tool can produce figures plotting the inputted crush stress parameter curves and graphs containing the calculated force-displacement and stress-displacement graphs for the selected geometry. The tool will also export an Excel file named *OutputTable.xlsx* containing the simulation's results and create a VTK file for each inputted model for 3D data visualisation. Finally, an output log is automatically written containing the simulation's specifications and important global parameters such as crushed mass, energy absorbed and specific energy absorbed. It should be noted that a separate VTK file will be created for each inputted part. Still, all parts can be visualised simultaneously in Paraview by individually opening all the files containing the parts, selecting them all in the *Pipeline Browser*, right-clicking and selecting the options *Add Filter>Alphabetical>Append Datasets*.

3.5.1 Input Control of the Analytical Tool Script

The following list details all the user-defined inputs that need to be specified, along with an explanation of what they control and, if appropriate, what input format must be used. Four distinct input sections were created in the script to separate the inputs into categories.

1. **Model File Input**: Input category controlling the imported STL models into the tool:
   - `ModelFilename`: Vector with the names of the STL files with the geometries to analyse. The files must remain closed during execution and must be located in the same folder as the `AnalyticalTool.m` script.

     Format: ModelFilename={'MainPart.stl','SubPart1.stl',...};
   - `initial_ply_count`= Vector containing the number of plies of the laminate for each inputted part submitted for analysis. The vector must be the same length as the `ModelFilename` vector. The ply count of the first model specified on `ModelFilename` vector will be the number specified on the first element of the `initial_ply_count` vector, the second model with the second element, and so on.

     Format:initial_ply_count=[PlyCountMainPart, PlyCountSubPart1,...], where PlyCountMainPart is an integer.

     Example: initial_ply_count=[8 4]
   - `Z_PartitionsOffset`: When partitioning the CAD model via the Partitioning tool VBA macro, the distance chosen between partitions made along the z-axis must be specified in this variable in millimetres. Recall that if two or more models are being inputted into the tool they must be partitioned together and only then separated into distinct STL files.
2. **Material Data Files Input**: Input category controlling the import of material data into the tool:

- `Ply_Thickness`: Thickness of an individual ply in millimetres.
- `Density`: Material's specific weight expressed in kg/m^3.
- `FilenameCurvature, FilenameVelocity, FilenameInclination, FilenamePlyCount`: Variables containing the Excel file names with the coefficients for the crush stress parameter curves describing the variation of the crush stress with the RoC, crushing velocity, inclination and laminate thickness, respectively. The `DataCurvature`, `DataVelocity`, `DataInclination` and `DataPlyCount` output files from the `MaterialCharacterization.m` program can be directly utilised as an input. The files must remain closed during execution and must be located in the same folder as the `AnalyticalTool.m` script.

  Example: FilenameCurvature:'DataCurvature.xlsx'

3. **Simulation Control Inputs**: Input category controlling the definitions for the simulation.
   - `DynamicAnalysis`: Set to false by default. If `DynamicAnalysis`is set to true, a dynamic simulation will run, and the program will consider the effects of crushing velocity on the assignment of the crushing stress to the structure's nodes. The tool will also automatically calculate the reduction of the crushing head's velocity as it progressively crushes the model. If the variable is false, a quasi-static simulation runs and the effect of crushing velocity on crush stress is disregarded.
   - `V0`: Only required if `DynamicAnalysis`=true. Varaible defining the impact velocity of the crushing head in m/s.
   - `ImpactorMass`: Only required if `DynamicAnalysis`=true. Varaible defining the mass of the impactor in kilograms.
   - `DropTowerFlag`: Only required if `DynamicAnalysis`=true. Distinguish between sled-impactor and drop-tower dynamic tests. If set to true, a drop-tower test is simulated, and the program considers the effect of gravity on the crushing head's velocity update process. Otherwise, a sled-impactor test is performed, and the gravitic acceleration is disregarded.
   - `optimisation`: Variable that determines whether the optimisation process is to be performed. If `optimisation`=false, then the value specified in `initial_ply_count` is used for the whole structure. Otherwise, if the variable is true, the optimisation process is performed, and the program will continuously increase the ply count until the specified target force is achieved.
   - `TargetForce`: Only required if `optimization`=true. Varaible specifying the target force in KN to be achieved by the optimisation process

4. **Plotting Input Control**: Input category controlling the outputs generated by the tool.
   - `MaterialCurves_Plot`: If the variable is set to true, the tool will open figures plotting each crush stress parameter curve inputted. Useful for visual verification that the intended curves were utilised.
   - `DisplacementGraphs_Plot`: If the variable is set to true, the tool will open figures plotting the force-displacement and crush stress-displacement graphs of the model submitted to analysis. If `DynamicAnalysis`=true, the crushing head velocity vs displacement graph will also be plotted and if `optimisation`=true, a ply count vs displacement graph will be plotted.

- CrushStressAndPlyCount_plot: If the variable is set to true, the tool will include in the output VTK file the required information for 3D visualisation of the crush stress and ply count distribution throughout the model.

Since no experimental data for the inclination and ply count effects on crush stress was available, the inputs FilenameInclination and FilenamePlyCount were left commented. In addition, some temporary code had to be devised to cover the lack of data. The permanent code was left commented out, and the temporary excerpts are marked with accompanying instructions to restore the code's full functionality once experimental data is available. As mentioned, the inclination effects on crush stress will be simulated via a theoretical curve built directly inside the program that reflects the literature's findings for similar materials.

3.5.2 Creating the nodes and triangles Structures

Structures in MATLAB function as a way to organise related data under a single entity. They act like containers with named compartments (fields) that can hold different data types (numbers, text, arrays, and even other structures) within each field. Structures can be created using the *struct* command, and data can be accessed or manipulated using dot notation (.), specifying the structure name and field name. Structures allow efficient grouping and management of complex data sets with various data types.

As previously mentioned, the nodes and triangles structures were developed as a central means for storing and accessing information about the individual nodes' characteristics and facet information, respectively. Storing all the data in the same place provides an excellent basis for implementing new features, increases the program's modularity, facilitates debugging and dramatically simplifies the code's complexity and number of variables in each script. However, creating the structures requires considerable computational power and is responsible for much of the program's total runtime. The function Treatment_v2.m was developed to make the structures.

Since all the fields in the nodes and triangles structures are the same, creating a separate structure for each triangle and node is unnecessary. Instead, all the data can be combined into a larger structure, where each node's or triangle's information corresponds to an individual row that can be accessed via index notation. For example, the command nodes(37) presents all the available information regarding the node found on row 37.

The function first allocates the names and dimensions of each field found in the triangles and nodes structures. The list below documents all the pre-allocated fields.

1. triangles structure:

    - triangles(i).coords: Contains a 3x3 matrix where each column contains the x, y and z global coordinates, in millimetres, of each node forming facet i.

    - triangles(i).normal Contains a column vector with the u, v and w components of the unit vector normal to the facet i.

    - triangles(i).indexes: Contains a row vector where the first component corresponds to the index position in the nodes structure of the node whose global coordinates correspond to those found in the first column in the triangles.coords matrix. Similarly, the second component corresponds to the index of the node found in the matrix's second column, and the third component corresponds to the third column.

2. nodes structure:

- `nodes(i).coords`: A column vector containing the x, y and z global coordinates, in millimetres, of the node i.
- `nodes(i).neighbours`: A 3xn matrix, where each column contains the x, y and z global coordinates, in millimetres, of the n nodes neighbouring node i.
- `nodes(i).normals`: A 3xn matrix, where each column contains the u, v and w components of the normal unit vectors of the n triangles that include node i.
- `nodes(i).curvature`: A floating-point number describing, in millimetres, the calculated radius of curvature in node i. The calculation of the RoC will be performed in the `geometry.m` function, so the value of 0 is pre-allocated.
- `nodes(i).inclination`: A floating-point number describing, in degrees, the calculated inclination in node i. The inclination calculation will be performed in the `geometry.m` function, so the value of 0 is pre-allocated.
- `nodes(i).distances`: A floating-point number describing, in millimetres, the sum of half of the total distance to the neighbouring nodes of node i with the same z-coordinate. The distance calculation is performed in the `geometry.m` function, so the value of 0 is pre-allocated.
- `nodes(i).ply`: An integer defining the number of plies in node i. The value for the number of plies will be assigned on the `CrushStress.m` function, so the value of 0 is pre-allocated.
- `nodes(i).CrushStress`: A floating-point number describing, in megapascal, the crush stress value assigned to node i. The assignment of the crush stress value will be done in the `CrushStress.m` function, so the value of 0 is pre-allocated.

Additionally, recall that one `nodes` and `triangles` structure is created for each part submitted for analysis. All the created structures can be found in the `partNodes` and `partTriangles` super-structures, and the data within them can be accessed via index notation. For example, the command `partNodes(1)` outputs the entirety of the `nodes` structures relative to the first model inputted in the `ModelFilename` variable.

The process by which the structures are created will now be explained. Firstly, the script directly utilises the data inside the V, F and N matrices to assign values to the `triangles.coords` and `triangles.normal` fields without any need for further computations. The remaining `triangles.indexes` field will be assigned values at a later stage.

Secondly, the function removed all the duplicate data from the V matrix, effectively leaving behind all the unique nodes, and then sorted those remaining nodes by order of descending z. A cycle starting at the first row of V begins: for each unique node, the function stores its global coordinates in the `nodes(i).coords` vector and then looks at all the `triangles.coords` matrices to find the triangles that contain the node under analysis. If a triangle contains the search node, the two remaining vertexes of the triangles are added to the `nodes(i).neighbours` matrix, and it's normal to `the nodes(i).normal` matrix.

Finally, the `triangles(i).indexes` field is filled by finding, for each triangle, the position in the `nodes` structure of the three nodes whose global coordinates can be found in `triangles(i).coords` matrix.

Note that, as a consequence of the described process, the `nodes` structure is ordered by descending z. Since a structure is crushed from top to bottom, the nodes in the `nodes` structure are displayed by the order in which they are being crushed. Figures B.1 and B.2 in Appendix B show a part of the nodes and triangles structures after the execution of the `Treatment_v2.m` function to demonstrate the described process (the specific inputs used do not matter since the tables are only meant to demonstrate the function's capabilities).

3.5.3 Calculating the Curvature, Inclination and Distance at all Nodes

The `geometry.m` function was created to calculate the inclination, radius of curvature and distances for all nodes present in the `nodes` structure. It takes as input the `nodes` structure obtained after executing the `treatment_v2.m` function and the variable flat_transition from the RoC material data file after the information has been unpacked into a usable form by the `MaterialCurves.m` function. After the execution of `geometry.m` function, the same `nodes` will be outputted but with values assigned for the `distance`, `curvature` and `inclination` fields.

3.5.3.1 Inclination

The unit vectors normal to the facets were used to calculate the surface's inclination relative to the z-axis. Inclination is a surface property, but the tool requires data on the nodal level, which represents a point in a surface. There is no mathematical formulation for the inclination of a point, so the tool must average the inclination of the surface surrounding the node to obtain an approximated value. To this effect, the `geometry.m` function runs all the nodes and, for every individual node i, accesses the `nodes(i).normals` field, which contains all the unit vectors normal to the triangles that node i is a part of. For each normal vector, the program then calculates the inclination via Equation (3.4) and then averages all the values and stores the data in the `nodes(i).inclination` field.

$$\theta = \arcsin |w| \tag{3.4}$$

, where θ is the inclination in degrees and $|w|$ is the absolute value of the z-component of the unit vector normal to the surface. Note that this only works because the length of the vector is one, which simplifies the equation of the sine as it's being divided by one.

The process of calculating the surface's inclination can be verified by submitting conical specimens for analysis and comparing the values obtained to the expected values. If the cone's height and top and bottom radii are known, the inclination of the cone's surface can be calculated by hand via Equation (3.5).

$$\theta = \arctan \frac{R_b - R_t}{H} \tag{3.5}$$

, where H is the cone's height, and R_b and R_t are the cone's bottom and top radii, respectively.

A cone model with 200 mm of height, 100 mm bottom diameter and 40 mm top diameter was constructed and run through the analytical tool program. Figure 3.18 shows, via Paraview, the obtained distribution of assigned inclination values to the nodes. Calculating the theoretical value for the inclination via Equation (3.5), the value of $\theta = 8.53°$ was obtained. By referencing the image and the model's *nodes* structure, the inclination was observed to range from 8.30 to 8.70 degrees. It can be concluded, therefore, that the tool is correctly assigning values for the inclination. The slight variation can be attributed to approximation errors originating in the tesselation of a curved surface to several flat surfaces when creating the STL files. It should be noted that the relative error is small enough not to impact the tool's accuracy.

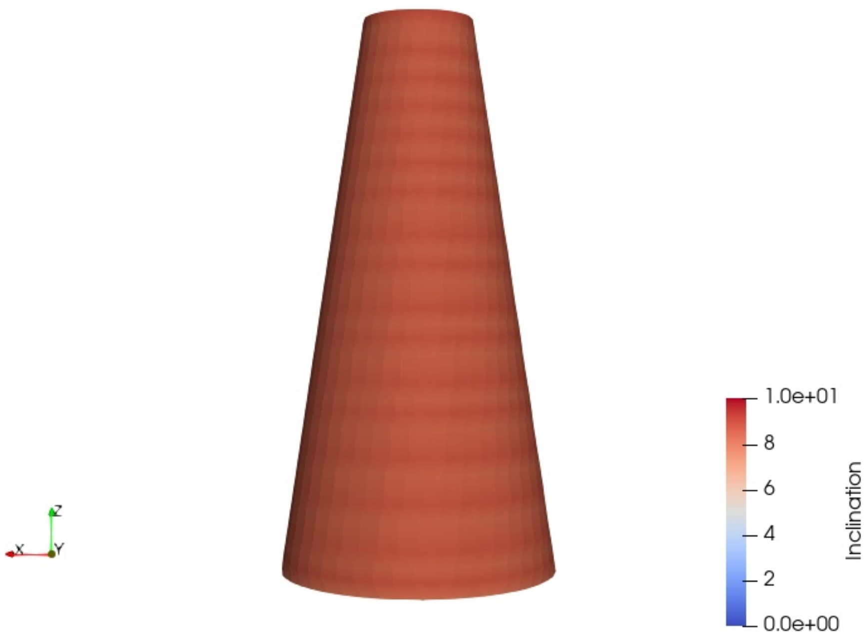


Figure 3.18: Calculated values of inclination for a cone model

3.5.3.2 Distance

Calculating the distance between nodes is necessary to know the perimeter length of the model's cross-section. The thickness can be calculated by multiplying the number of plies with the ply thickness, and in conjunction with the cross-section's perimeter length, the area of the cross-section can be calculated, which in turn is necessary to calculate the force to crush the structure at each cross-section.

Since the force is calculated for each cross-section, the only relevant nodal distances are the distances between neighbouring nodes with the same z-coordinate. However, simply calculating the distances between nodes and adding them up would not work because this process would be repeated for the next node, and the same distance would be counted twice. To solve this, the distance to the neighbouring node is divided by two, and the other half is accounted for when the process is repeated for the neighbouring node.

For each node i, the neighbouring nodes' coordinates are accessed via the information stored in the `nodes(i).neighbours` matrix and the nodes with the same z-coordinate as node i are separated into a different matrix called `same_level_nodes`. The program then calculates all the distances from node i to the nodes in the `same_level_nodes` matrix, sums them up, and divides the total by two. The resulting distance is stored in the `nodes(i).distances` variable. If the node has no neighbours in the same z-coordinate, the symbolic value of 0.001 is assigned to avoid errors in the script.

The same conical model in Figure 3.18 can be used to verify that the nodal distances are being correctly assigned. As mentioned, the top diameter was 40 mm, giving a total perimeter of $40\pi = 125.66$ mm at the cone's top. From the resulting `nodes` structure, it was seen that the tool assigned a distance of 4.3238 mm to the 29 nodes located at $z = 200$ mm, producing a total cross-sectional perimeter of $4.3238 \times 29 = 125.41$ mm. The difference between the calculated and theoretical values was 0.25 mm, representing a relative error of 0.2% and confirming the correct functioning of the distance calculation process.

3.5.3.3 Radius of Curvature

For a section with constant radius R, the curvature k (Menger curvature) along the whole section is given by equation (3.6). However, since the crush stress parameter curve of the variation of the crush stress with the radius of curvature uses R, the curvature k will not be used.

$$k = \frac{1}{R} \tag{3.6}$$

Suppose three consecutive points A, B and C, with coordinates (x_A, y_A), (x_B, y_B) and (x_C, y_C), respectively. The radius of curvature of the curve containing points A, B, and C can be calculated using Equation (3.7):

$$R = \frac{\overline{AB} \times \overline{BC} \times \overline{CA}}{4A_{ABC}} \tag{3.7}$$

, where $\overline{AB}$, $\overline{BC}$, $\overline{CA}$ are the distances between the points A and B, B and C, and C and A respectively and A_{ABC} is the area of the triangle formed by the points A, B and C.

The Shoelace formula (Equation (3.8))can be used to calculate the area of any n-sized polygon from the coordinates of its vertexes.

$$A = \frac{1}{2}\sum_{i=1}^{n}\begin{vmatrix} x_i & x_{i+1} \\ y_i & y_{i+1} \end{vmatrix} \tag{3.8}$$

In this case, the Shoelace formula was applied with $n = 3$ to calculate the area of the triangle defined by the three consecutive points A, B and C. Equation (3.9) shows the utilised expression and its simplification.

$$\begin{aligned} A_{ABC} &= \frac{1}{2}\sum_{i=1}^{3}\begin{vmatrix} x_i & x_{i+1} \\ y_i & y_{i+1} \end{vmatrix} = \frac{1}{2}\left(\begin{vmatrix} x_A & x_B \\ y_A & y_B \end{vmatrix}\begin{vmatrix} x_B & x_C \\ y_B & y_C \end{vmatrix}\begin{vmatrix} x_C & x_A \\ y_C & y_A \end{vmatrix}\right) \\ &= \frac{1}{2}[(x_B - x_A)(y_C - y_B) - (y_B - y_A)(x_C - x_B)] \\ &= \frac{1}{2}[x_A(y_B - y_C) + x_B(y_C - y_A) + x_C(y_A - y_B)] \end{aligned} \tag{3.9}$$

Similar to inclination, the radius of curvature cannot be calculated at a single node since it is a property of a continuously defined curve. Therefore, the neighbouring nodes must be used to approximate a curve from which to calculate the radius of curvature. The curves obtained must be parallel to the XY planes, so only the neighbouring nodes located in the same z-coordinate as the search node will be used for the RoC calculation.

The `geometry.m` script runs through all the structure's nodes, and for each node, i finds and stores the coordinates of the neighbours with the same z-coordinate as node i. From here, the expression in Equation (3.9) is used to calculate the area and the $\overline{AB}$, $\overline{BC}$, $\overline{CA}$ distances are calculated. Finally, the radius of curvature is obtained via Equation (3.7), and the value is stored in the appropriate nodes(i).curvature field.

An upper and lower bound had to be imposed for the allowable values of the RoC. In the fitted RoC crush stress parameter curve (see Figure A.1), the values nearing x=0 tend exponentially to infinity, which would translate to an unrealistic crush stress assignment by the analytical tool, so a lower limit was imposed that if the calculated RoC is less than 5 millimetres, the program would instead store the value 5. On the other hand, the area of the ABC triangle for flat sections would be zero since all points would lie flat on a line and Equation (3.7) would give an error due to the division by zero. The feature was implemented that if the area is zero or the calculated value is greater than the `flat_transition` RoC, then the stored RoC value will be the `flat_transition` RoC. Recall that the definition of the `flat_transition` is the RoC upon which a plateau is achieved on the crush stress parameter curve, meaning that for any larger RoC, the corresponding crush stress is the same, and no error is incurred by the implementation of the upper bound.

However, the described curvature assignment process only works if the node has two, and only two, neighbouring nodes in the same z-coordinates, which is not always the case. Alternative methods had to be developed to calculate the RoC to account for the other possible cases.

We start with the case where the node has three or more neighbouring nodes in the same z-coordinate, which occurs in geometric intersections, for example, the intersection point of an internal rib with the structure's shell. Since there is an intersection of geometries, there is not a single value that can define the RoC at that point; in fact, for this example, there can be three possible distinct values for the radius of curvature, depending on the nodes selected for the curvature calculation. Suppose that node 0 has three neighbours on the same z-coordinate: node 1, node 2 and node 3 (see Figure 3.19). Three curves containing node 0 can be characterised:

- Curve 1-0-2
- Curve 1-0-3
- Curve 2-0-3

Each of the previously defined curves has a distinct RoC, and since each one of them symbolises a possible RoC value at that node, node 0 was assigned the average of the three possible values.

The code counts the number of neighbouring nodes in the same z-coordinate as node i, and if the count is greater than two, then all possible curve combinations containing node i are obtained, all the possible radii of curvature calculated, and the average stored into the `nodes(i).curvature` field. Note that this will cause a lower value to be assigned to the intersection node than the surrounding surfaces, effectively creating a local reinforcement effect in geometric intersections.

Another exception is made if the code only finds a single neighbouring node at the same z-coordinate, which might happen at the free edges of open-section geometries. In this case, there is not enough available information from which to calculate the radius of curvature, so the tools assign instead to node i the same RoC of the neighbouring node. Suppose it so happens that the neighbouring node only has node i as its neighbour. In that case, both nodes are assigned the `flat_transition` RoC, which will cause a discontinuity in the 3D plot of the RoC field but is the most conservative option available since there is not enough data to calculate the radius of curvature.

The final exception implemented was in the case of no neighbouring nodes at the same z-coordinate as node i being found, which might randomly happen throughout more complex ge-

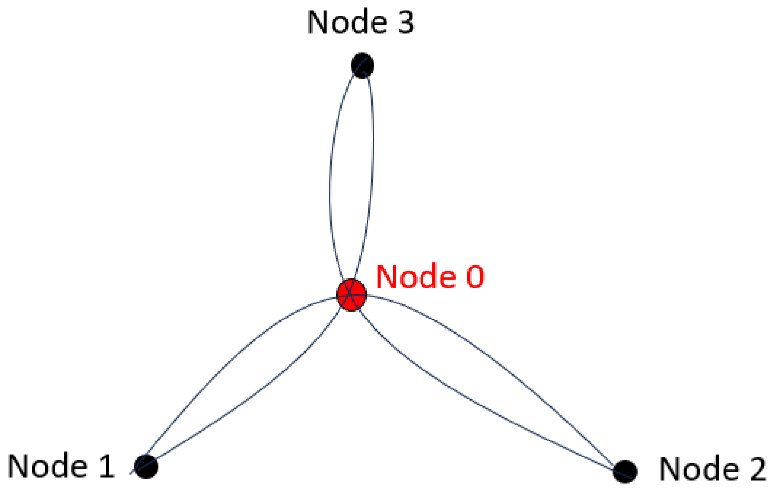


Figure 3.19: Diagram of the curvature in geometric intersections

ometries due to the tesselation process when creating an STL file. In this case, node i is assigned the radius of curvature of the neighbouring node with the highest z-coordinate. Since the `nodes` structure is ordered by descending z, it is guaranteed that the node above node i had already been previously assigned a value for RoC and, therefore, the program could always assign a value.

The same method of utilising different model geometries can be used to verify the correct implementation of the radius of curvature code. Submitting a simple tubular specimen, the program would be expected to assign the tube's radius for all nodes since there is no geometry variation across the tube's length. Figure 3.20 shows the RoC field for a tube with a 50 mm diameter, and it can be seen that the tool correctly assigned the tube's radius to the nodes. Note that the scale's data range was adjusted for better visualisation of the intended details for all the images used for verification. If the range were left as original, the model would appear monochromatic because of the broad range and geometric details could not be observed.

Inputting the same cone that was used for the inclination verification, it would be expected to see a linear variation between 20mm RoC at the cone's top and 50mm at the bottom. Indeed, from Figure 3.21, a linear colour shift can be seen, and the value at the cone's extremities matches the correct dimensions.

A model of a tube with two internal ribs in a cross-pattern was submitted for analysis to verify the code's ability to handle geometric intersections, and a half-tube model was submitted to verify the ability to handle open edges. The results can be seen in Figures 3.22 and 3.23. Additionally, in Figure 3.22, note that on the intersecting edges, lower values of RoC were assigned, representing the previously mentioned reinforcement effect caused by the averaging of the RoCs.

3.5.4 Assigning Crush Stress Values to the Structure's Nodes

The large final function `CrushStress.m` addresses, in order of execution, the assignment of the number of plies to the nodes, the assignment of crush stress values to the nodes, calculating the

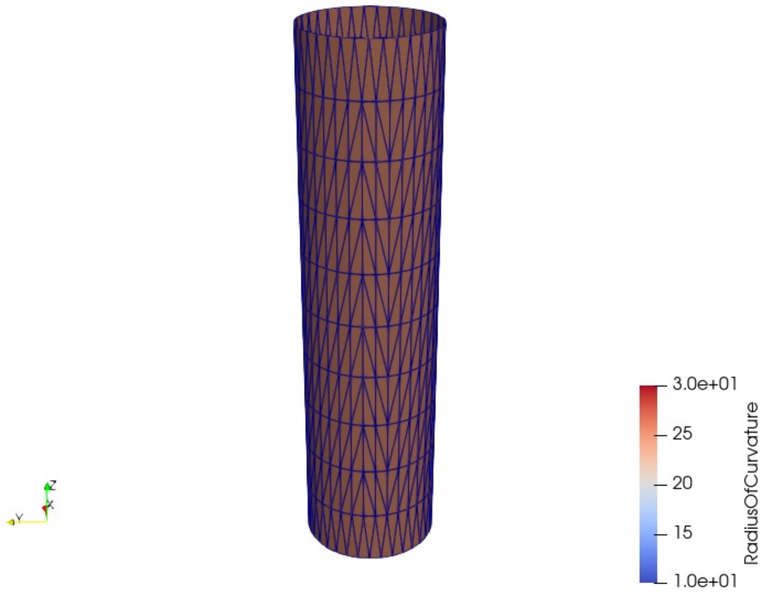


Figure 3.20: Radius of curvature distribution for a tubular model

total force, number of plies optimisation (if the `optimisation` variable is set to true), updating the velocity (if it is a dynamic simulation), plotting the displacement graphs, and constructing the excel output table.

Unlike the previous functions addressed where, if multiple parts were submitted for analysis, the functions would be executed once for each part, the `CrushStress.m` function is only executed once and runs all parts simultaneously. This is because some of the calculations performed, such as the velocity update or the total force calculation, are impacted by each part's effect on them, and, therefore, all the parts' data needs to be available to obtain an accurate result.

Additionally, the `CrushStress.m` function does not run a loop across the `nodes` structure like the previous scripts. Instead, the function runs a loop through all sets of nodes that form complete cross-sections across the structure's length. This is because to calculate the force-displacement curve, it is necessary to know the crush stress at every displacement and the cross-sectional area. As mentioned in Section 3.5.3.2, the cross-sectional area is obtained by multiplying the perimeter by the thickness and the distances between neighbouring nodes are already stored in the `nodes` structure. However, this method requires a group of nodes to exist within the same z-coordinate that spans the entire structure's perimeter. Since it's known that the Brep definition on STL files forces the creation of nodes in a surface split, it can be concluded that at every z-coordinate that the structure was partitioned, a complete cross-section of nodes can be found, and the force could be calculated.

In essence, the code runs through all z coordinates where the structure has been partitioned, assigning the number of plies and crush stress values to the nodes found there. It then calculates the force and, if necessary, optimises the ply count and updates the velocity. Recall that the model was partitioned along the z-axis in constant intervals specified in the `Z_PartitionsOffset` variable, meaning that points in the displacement graphs can be obtained every `Z_PartitionsOffset` intervals. For example, if a structure was partitioned along the z-axis at every 10 mm, the tool will obtain predictions for the crushing force at 10 mm displacement intervals. Reducing the z-axis partition offset when partitioning the model will provide more points on the displacement graphs, albeit at the cost of increased runtimes.

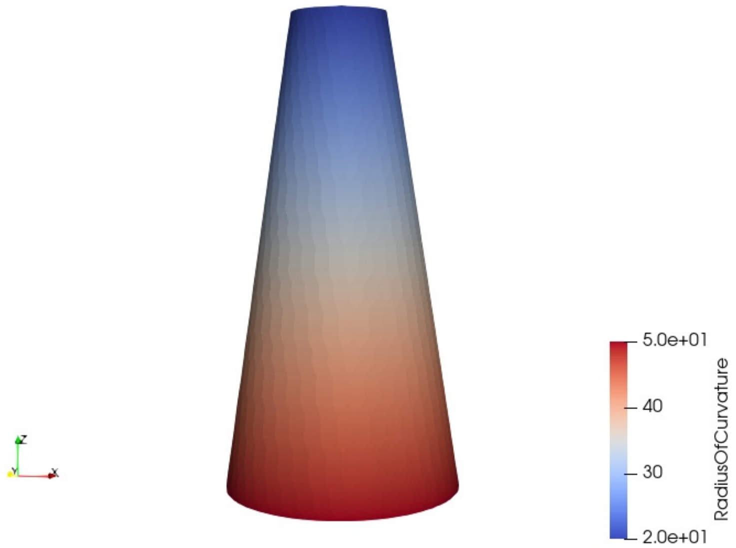


Figure 3.21: Radius of curvature distribution for a conical model

For each cross-section, all the required and calculated data is stored in a unique structure called `LayerStructure`, which is later utilised to create the displacement plots and import back into the `nodes.ply` and`nodes.CrushStress` structure fields the assigned nodal values for the ply count and crush stress, respectively. If the simulation is dynamic, a variable called `velocityLayer` stores the velocity at the layer under analysis. Once the subsequent layer is analysed, its value is updated to reflect the impactor's deceleration.

3.5.4.1 Ply Count and Crush Stress

For every z-coordinate containing a z-partition, the program runs through all `nodes` structures in the `PartNodes` super-structure to find all nodes forming the complete cross-section. It then stores all the data in `LayerStructure`. Each part's data is stored in a unique row in the `LayerStructure` structure, which can be accessed via index notation. For example, the command `LayerStructure(i)` outputs all the stored information regarding the part corresponding to the i-th index in the `ModelFilename` vector.

At this point, the tool assigns a ply count for each part in the `LayerStructures` structure according to the `initial_ply_count` input vector. If the `optimisation` variable is false, then the assigned ply count is stored back in the appropriate `nodes.ply` structure field. However, if `optimisation`=true, the code will change the assigned ply count so that a target force can be achieved and only after this process is the ply count data stored back into the *nodes* structures. The workings of the optimisation process will be explained later.

The crush stress is assigned via the normalised crush stress parameter curves. These curves represent by what factor the crush stress increases or decreases when one of the parameters, for example, the inclination, changes in relation to a reference value, which for the inclination is $\theta = 0°$. The crush stress at the node is first asserted by multiplying together all the geometrical factors of influence at the node's location. Equation (3.10) shows the equation used to determine the crush stress:

$$\sigma_c = \texttt{flat\_CrushStress} \times F_{\text{RoC}} \times F_\theta \times F_{\text{PC}} \tag{3.10}$$

where σ_c is the crush stress in MPa, `flat_CrushStress` is the crush stress value at the plateau of

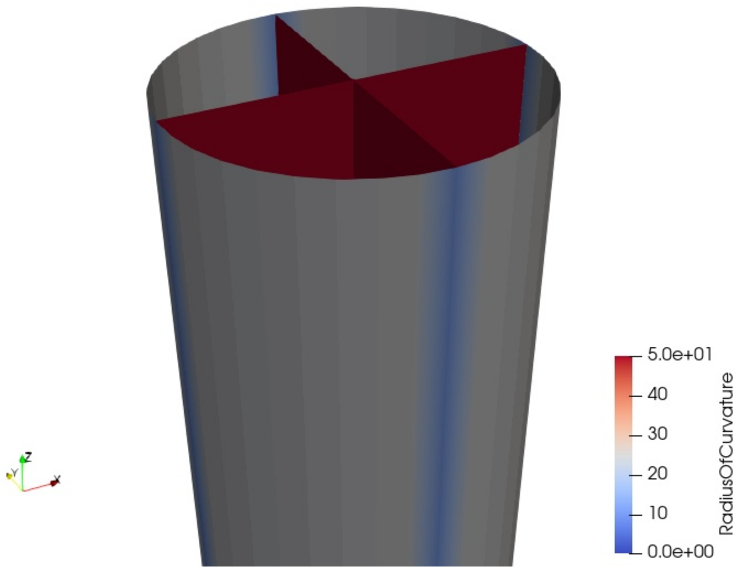


Figure 3.22: Zoomed on the radius of curvature distribution for a tubular model with internal ribs

the absolute RoC curve, in MPa, and F_{RoC}, F_θ and F_{PC} are the geometrical factors of influence of the radius of curvature, inclination and ply count, respectively. For a cross-sectional node i, contained in part k contained in the `LayerStructure` structure, the influence factors can be obtained according to Equation (3.11):

$$\begin{cases} F_{\text{RoC}} = \texttt{EqCurvature(x=LayerStructure(k).nodes(i).curvature)} \\ F_\theta = \texttt{EqInclination(x=LayerStructure(k).nodes(i).inclination)} \\ F_{\text{PC}} = \texttt{EqPlyCount(x=LayerStructure(k).nodes(i).ply)} \end{cases} \tag{3.11}$$

where `EqCurvature`, `EqInclination`, and `EqPlyCount` are the equations for the RoC, inclination and ply count normalised crush stress parameter curves obtained from the `MaterialCharacterization.m` program.

If the simulation is dynamic, the crush stress value obtained in Equation (3.10) is updated to consider the velocity effect per Equation (3.12):

$$\sigma_c = \sigma_c \times F_V \tag{3.12}$$

where F_V is the velocity's factor of influence, which can be obtained via Equation (3.13).

$$F_V = \texttt{EqVelocity(x=velocityLayer)} \tag{3.13}$$

This process is repeated for all the nodes in the `LayerStructure` structure. Similar to the ply count assignment, if `optimization=false`, the data is then stored in the appropriate `nodes.CrushStress` structure fields. However, if `optimization=true`, the change in the ply count will cause the F_{PC} to change, and the value of the crush stress needs to be recalculated to reflect the new ply count before being stored into the appropriate `nodes.CrushStress` structure fields.

The crush stress's upper and lower confidence bounds are calculated after executing the code section containing the optimisation process, which assigns the final ply counts to the nodes. The process described above is repeated, but instead of using the normalized crush stress parameter

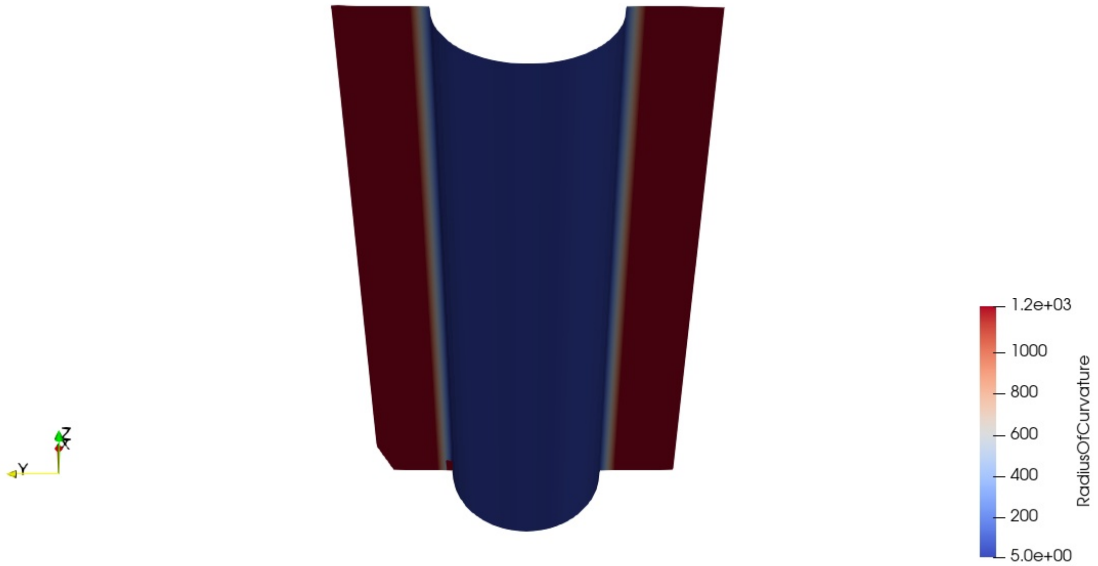


Figure 3.23: Radius of curvature distribution for a half tube model

curves equation (`EqCurvature`, `EqInclination`, `EqVelocity` and `EqPlyCount`) for the calculation of the factors, the bootstrapped upper and lower bound equations are used.

Code was implemented to obtain an average crush stress for the cross-section to plot the structure's stress-displacement graph. To compute the average crush stress, a weighted average of all nodal crush stress values along the cross-section was performed, per Equation (3.14):

$$\sigma_{av} = \frac{\sum_{i=1}^{n} (\sigma_{c_i} \times d_i)}{\sum_{i=1}^{n} d_i} \tag{3.14}$$

, where σ_{av} is the average crush stress, in MPa, σ_{c_i} is node i's assigned crush stress and d_i is node i's assigned distance. To obtain the upper and lower bounds for the average crush stress, Equation (3.14) was again used, but σ_{c_i} was replaced with the assigned upper and lower crush stress bounds at each node.

Note, however, that from this process, only nodes located at the model's z-partitions have been assigned values for the crush stress and ply count. If the `CrushStressAndPlyCount_plot` variable is false, the nodes missing the values can be ignored. However, if the variable is true, all nodes must be assigned values to be compatible with Paraview. For the nodes lying between cross-sections, a process identical to the assignment of the RoC when there are no same z-coordinate neighbours was used. The code finds the highest z-coordinate neighbour and assigns the same crush stress and ply count to the node in question. Since the *nodes* structures are ordered by descending z, it is always guaranteed that the neighbour already has a value assigned to them.

The same cone model used for the RoC and inclination verifications was submitted for analysis to verify that the code works correctly. A quasi-static simulation with an 8-ply carbon fibre laminate was performed. Figure 3.24 and 3.25 show the crush stress and ply count field distributions, respectively, and Figure 3.26 shows the calculated stress-displacement graph.

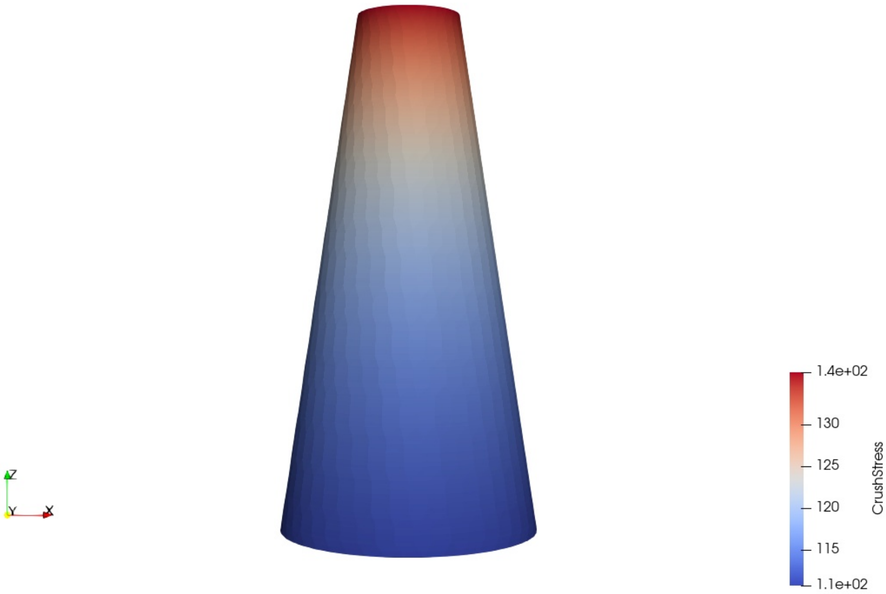


Figure 3.24: Cone model crush stress distribution

Since it is known from the fitted RoC curves of the carbon fibre experimental data that the crush stress decreases with the radius of curvature, it would be expected on a conical model for the crush stress to decrease from the cone's top to its bottom since the radius of curvature increases. From referencing Figure 3.24 and 3.26 it can be seen that this is the case, suggesting that both the crush stress and average crush stress are being correctly calculated. In addition, the crush stress confidence bounds were plotted in Figure 3.26, confirming that the code is working correctly. Finally, an 8-ply laminate was assigned to the whole structure, and Figure 3.25 shows eight plies across the entire structure, suggesting that the ply count is also being correctly assigned.

3.5.4.2 Force

All the necessary data is already available to calculate the total force required to crush the cross-section under analysis. For every part k, the force necessary to crush the individual part can be calculated, in kN, using Equation (3.15):

$$F_k = \frac{\sum_{i=1}^{n}(\sigma_{c_i} \times d_i) \times t_k}{1000} \tag{3.15}$$

$$F_k = \frac{\sum_{i=1}^{n}(\sigma_{c_i} \times d_i) \times \texttt{Ply\_Thickness} \times \texttt{PlyCount}_k}{1000}$$

where F_k is the force to crush part k, σ_{c_i} is the crush stress assigned to node i, d_i is the distance assigned to node i, t_k is the laminate thickness in mm for part k, `Ply_Thickness` is the thickness of a ply in mm (defined in the material input section) and `PlyCount` is the number of plies at that

Figure 3.25: Cone model ply count distribution

cross-section for part k. To calculate the total force to crush the cross-section, the contribution of each part needs to be added, see Equation (3.16):

$$F_{total} = \sum_{k=1}^{n_k} F_k \tag{3.16}$$

where n_k is the total number of parts submitted for analysis.

If `optimization=false`, then the calculated value in F_{total} is stored and used for the velocity update and as a point in the force-displacement graph. However, if `optimization=true`, then the increase in the number of plies will change both the laminate thickness and the assigned crush stress, and the described process needs to be repeated once the final ply count to achieve a specified target force is found. That final force value obtained after the optimisation will be used for the velocity update and as a point in the force-displacement graph.

Obtaining the upper and lower force bounds for the total force can be done by repeating the same process described in Equations (3.15) and (3.16) but by replacing σ_{c_i} with the upper and lower bounds for the assigned crush stress.

Figure 3.27 shows the obtained force-displacement graph for the same 8-ply carbon fibre cone quasi-static simulation made for verifying the crush stress calculation process. The force-displacement curve obtained increases along the displacement due to the increase in the cross-sectional area despite the decrease in crush stress seen in Figure 3.26. The force-displacement curve obtained follows the trend documented in the conical specimen crush testing literature, where the force gradually increases with the displacement.

As another means of verification, the total force was calculated by hand using Equation (3.17), where σ_{av} is the average crush stress in MPa and A_c is the area of the cross-section in mm^2.

$$F_T = \frac{\sigma_{av} \times A_c}{1000} \tag{3.17}$$

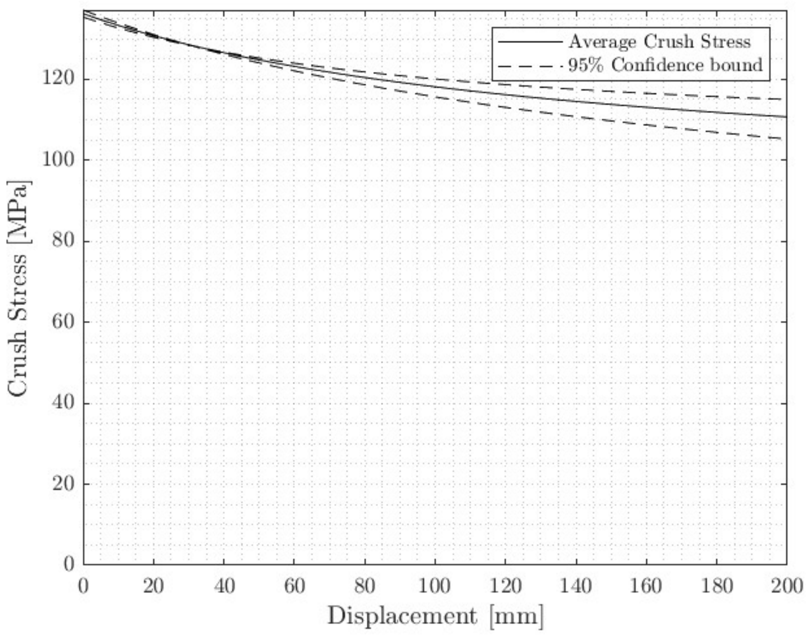


Figure 3.26: Stress-displacement curve of a quasi-static, 8-ply carbon fibre cone simulation

In the cone simulation, the calculated average crush stress at zero displacement was 136.179 MPa. The area of the cross-section can be obtained by multiplying the perimeter by the thickness. The cone's radius is 20mm at zero displacement, giving a total perimeter of 40π. The *Ply_Thickness* variable used in the simulation was set to 0.3 mm, and the cone had an 8-ply laminate, so $A_c = 40\pi \times 0.3 \times 8 = 301.59$ mm^2. Using Equation (3.17), the hand-calculated value for the total force is 41.2549 kN. The analytical tool obtained a value for the total force of 41.2582 kN, a relative error of 0.008%.

Finally, Equation (3.18) demonstrates that both Equations (3.15) and (3.17) can be used to calculate the total force (if the structure is composed of a single part).

$$
\begin{aligned}
F_T &= \frac{\sum_{i=1}^{n}(\sigma_{c_i} \times d_i) \times \texttt{Ply\_Thickness} \times \texttt{PlyCount}}{1000} \\
&= \frac{\sum_{i=1}^{n}(\sigma_{c_i} \times d_i) \times \sum_{i=1}^{n} d_i \times \texttt{Ply\_Thickness} \times \texttt{PlyCount}}{\sum_{i=1}^{n} d_i \times 1000} \\
&= \frac{\sigma_{av} \times \sum_{i=1}^{n} d_i \times \texttt{Ply\_Thickness} \times \texttt{PlyCount}}{1000} \\
&= \frac{\sigma_{av} \times P \times \texttt{Ply\_Thickness} \times \texttt{PlyCount}}{1000} \\
&= \frac{\sigma_{av} \times A_c}{1000}
\end{aligned}
\tag{3.18}
$$

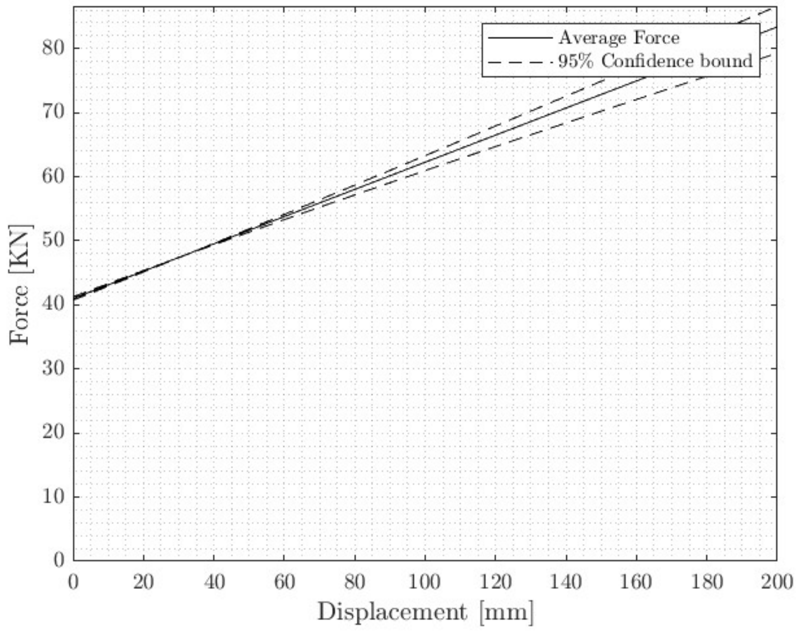


Figure 3.27: Force-displacement graph of a quasi-static, 8-ply carbon fibre cone simulation

3.5.4.3 Ply count Optimisation

Before the optimisation code is executed, the tool assigns a ply count for each part in the `LayerStructures` structure according to the `initial_ply_count` input vector. That ply count is then used to calculate the crush stress and total force. Suppose the `optimization` variable is false. In that case, the user wants a prediction for a structure whose parts have the ply counts specified `initial_ply_count` vector and no further changes to the assigned ply counts are needed, so the tool skips the optimisation code. It uses the previously calculated values for crush stress and force to plot the displacement graphs, obtain the confidence bounds and update the velocity.

On the other hand, suppose the `optimization` variable is set to true. In that case, the user intends to determine the minimum number of plies at each cross-section needed so that the structure can withstand a minimum force specified in the `TargetForce` variable. For each cross-section, after the total force is calculated for the parts with the initial ply counts assigned, the program compares the total force to the target force. If the value is smaller, the program adds another ply to all the parts, increasing both the laminate's thickness and the ply count factor (F_{PC}) and, therefore, requiring new nodal crush stress values and total force values to be recalculated using Equations (3.11), (3.10), (3.15) and (3.16). It then checks the new total force value against the target force. If it is less, the cycle repeats until the target force is achieved, at which point the newly obtained values for the ply counts, nodal crush stresses, and total force are saved, and the optimisation process is concluded.

The function then uses the updated values to calculate a new average crush stress, the upper and lower bounds for the stress and force-displacement bounds, and, if it is a dynamic simulation, update the velocity. A new graph detailing the number of plies across the displacement for every part is also plotted.

Note that the optimisation code is only able to increase the ply count and cannot decrease it, meaning that the ply counts specified in the `initial_ply_count` input vector are, effectively, the minimum allowable number of plies for each part's laminates. Additionally, the tool can only increase the ply count for all the parts simultaneously, meaning that it is unable to maintain a ply count fixed for a specific part whilst optimizing the others.

The same quasi-static simulation for an 8-ply carbon fibre conical model was submitted to verify that the optimisation code works correctly. However, the `optimization` variable was set to true, and a target force of 70 kN was specified. Figures 3.28 and 3.29 show the obtained force vs displacement and the number of plies vs displacement graphs, and Figure 3.30 provides a 3D visualisation of the ply count assignments.

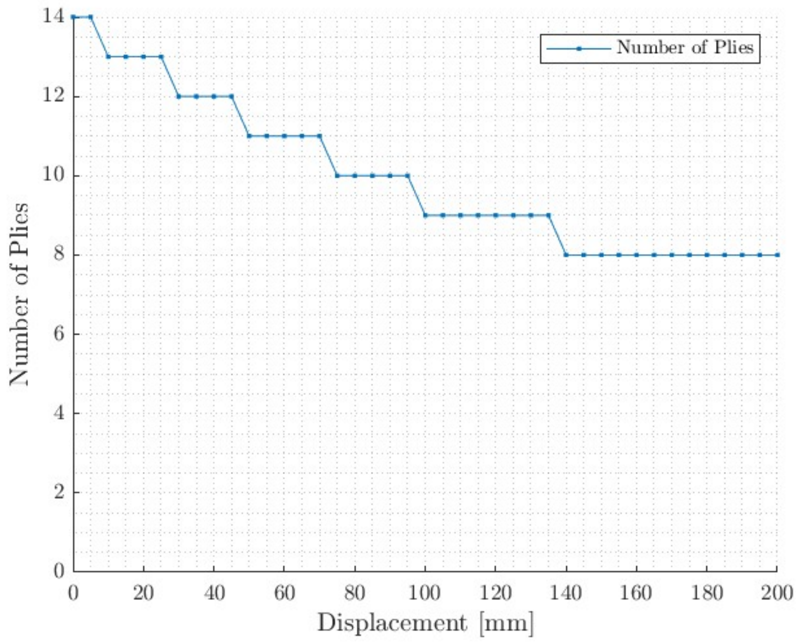


Figure 3.28: Cone model optimised for 70 kN ply count vs displacement graph

Observe how, across the displacement, the ply count needed to withstand 70 kN decreases. It had already been observed that on a standard static simulation, the force increased with the displacement due to the increase in cross-section. For an optimised cone, it follows that fewer plies would be assigned along the displacement since the increasing cross-sectional area already naturally increases the area without the addition of plies. Additionally, a note from Figure 3.28 that at the bottom of the cone, fewer plies would be able to withstand the specified 70 kN force; however, since the cone had been defined to initially have eight plies, the laminate's ply count could not decrease past eight plies.

3.5.4.4 Crushing Head Velocity Update

If the simulation is dynamic, the final operation performed before another cross-section is analysed is to update the crushing head's velocity. As the crushing head moves downwards and progressively crushes the structure, its kinetic energy is gradually absorbed, causing the crushing head to decelerate along the displacement. This deceleration needs to be accounted for to maintain the tool's accuracy. The crushing head's mass is specified in the `ImpactorMass` variable, and the velocity at which it impacts the structure is defined in the `V_0` variable.

Two different ways of calculating the deceleration were implemented into the tool and are controlled by the `DropTowerFlag` variable. If the variable is true, the program simulates a drop tower test by assuming a free-falling mass subjected to the acceleration of gravity impacting the

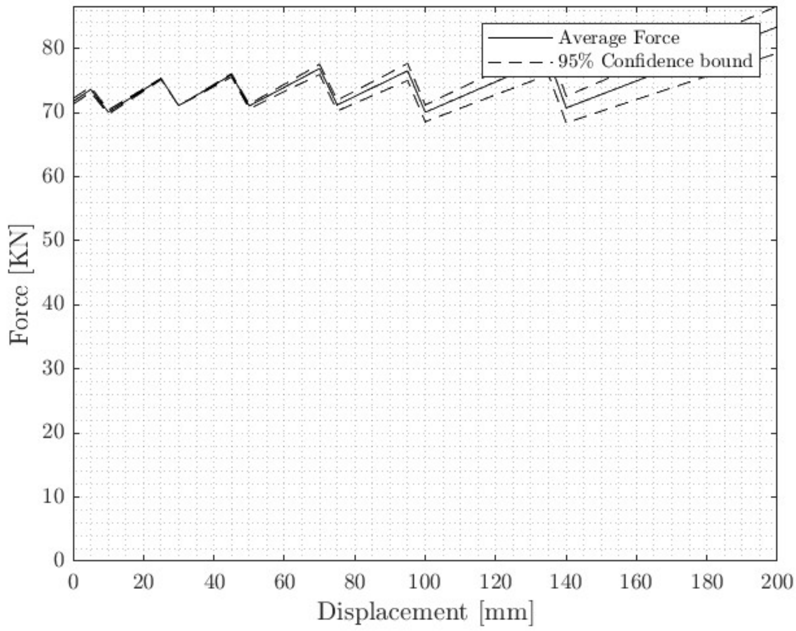


Figure 3.29: Cone model optimised for 70 kN force vs displacement graph

structure at velocity `V_0`. On the other hand, if `DropTowerFlag` is false, then the program simulates a sled-impactor test by assuming a horizontally sliding mass impacting the structure at velocity `V_0`. Since the mass horizontally slides on rails, gravity's acceleration is disregarded for sled-impactor tests.

It should be noted that the effects of air resistance and contact friction are not considered for the deceleration calculations. Earlier iterations of the program attempted to consider friction; however, its implementation proved difficult since current testing machines have no way of measuring the energy losses due to friction, nor are the dynamic friction coefficients known, making it impossible to predict friction losses realistically. Even so, predictions were made considering arbitrarily large friction coefficients, and the results obtained indicated that the energy losses due to friction were negligible, accounting for 0.5% of the total energy absorbed. The increased code's complexity and necessary user input to enable the friction calculation was deemed excessive, considering that the energy loss was insignificant even for the worst-case scenario, and the feature was removed from the current code's version.

The free-body diagram found in Figure 3.31 was used to calculate the deceleration of the impactor, considering a drop tower test. M is the impactor's mass, W is the impactor's weight, F_t is the previously calculated total force, and Q_Z is the resultant force indicating the direction of motion.

From the free body diagram, Equation (3.19) was used to calculate the impactor's acceleration. Note from the sign convention that if the acceleration a is positive, the impactor is accelerating, and if it's negative, it is decelerating. Since $W << F_T$, the impactor decelerates rapidly.

$$\begin{aligned} W - F_T &= Q_Z \\ M \times g - F_T &= M \times a_z \\ a_z &= g - \frac{F_T}{M} \end{aligned} \tag{3.19}$$

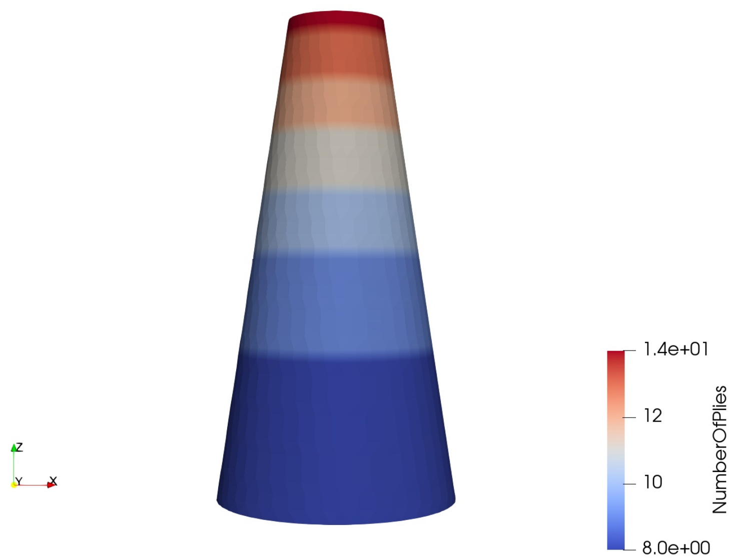


Figure 3.30: 3D Ply count distribution for a cone model optimised for 70 kN

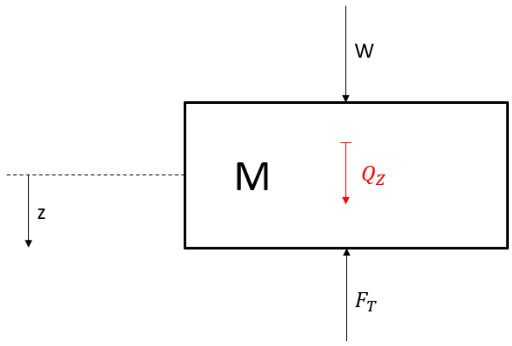


Figure 3.31: Free body diagram of a drop tower's impactor

The free-body diagram for the sled-impactor test is the same as the one used for the drop tower but without the weight component included. With this, Equation (3.20) was used to calculate the acceleration of the sled impactor.

$$\begin{aligned} -F_T &= Q_Z \\ -F_T &= M \times a_z \\ a_z &= -\frac{F_T}{M} \end{aligned} \tag{3.20}$$

Between cross-sections, the impactor is said to have a uniformly accelerated movement, whose equations of motion for the position and velocity as a function of time can be seen in Equation (3.21).

$$\begin{cases} z(t) = z_0 + v_0 t + \frac{1}{2} a_z t^2 \\ v(t) = v_0 + a_z t \end{cases} \tag{3.21}$$

, where z_0 and v_0 are, respectively, the position and velocity when $t = 0$. The equation of motion as a function of time can be rearranged to obtain a single equation for the velocity as a function of displacement, which is documented in Equation (3.22).

$$v(\delta) = \sqrt{v_0^2 + 2a_z\delta} \tag{3.22}$$

, where δ is the displacement, which is given by $\delta = z - z_0$. Using Equations (3.19) and (3.20), the expressions obtained for a_z can be placed into Equation (3.22) to obtain the velocity update functions for both the drop-tower and the sled impactor.

Equation (3.22) was used for the velocity update in the code. The updated velocity is the value obtained from the equation, v_0 is the current velocity at that cross-section, and δ is the crushing head displacement, which is the distance between consecutive partitions, which corresponds to the value stored in the `Z_PartitionsOffset` input variable. The expression for a_z is chosen between Equations (3.19) and (3.20) depending if `DropTowerFlag` is true or false.

If the updated velocity reaches zero at any point, the crushing head stops moving, and the simulation ends. Effectively, the cycle running through all the cross-section breaks and the remaining nodes in the `nodes` structures are intentionally left with the value of 0 assigned to the crush stress and ply count.

A dynamic sled-impactor simulation with a mass of 100kg impacting the structure at 10 m/s was submitted to verify the code for the velocity update. The conical model used the same 8-ply carbon fibre laminate as before. The optimisation setting was turned off. Figures 3.32, 3.33, and 3.34 show the resulting velocity, force and stress-displacement curves, respectively, and Figure 3.35 shows the 3D visualization of the assigned nodal crush stress values.

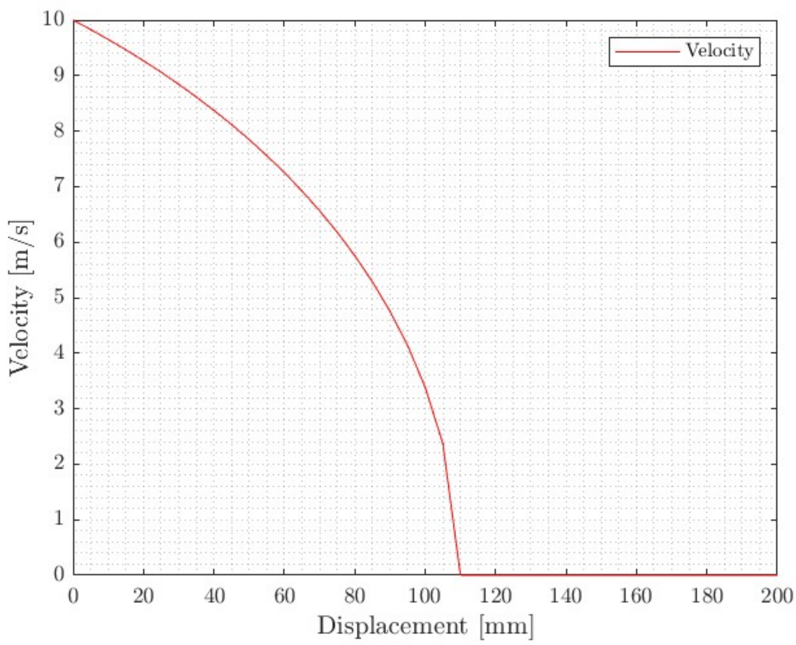


Figure 3.32: Velocity-displacement curve of a dynamic, 8-ply carbon fibre cone simulation

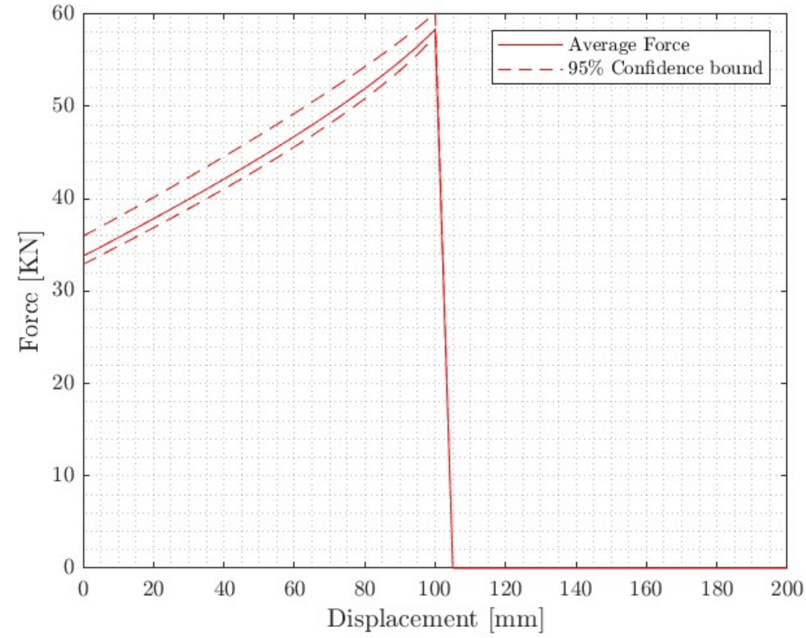


Figure 3.33: Force-displacement curve of a dynamic, 8-ply carbon fibre cone simulation

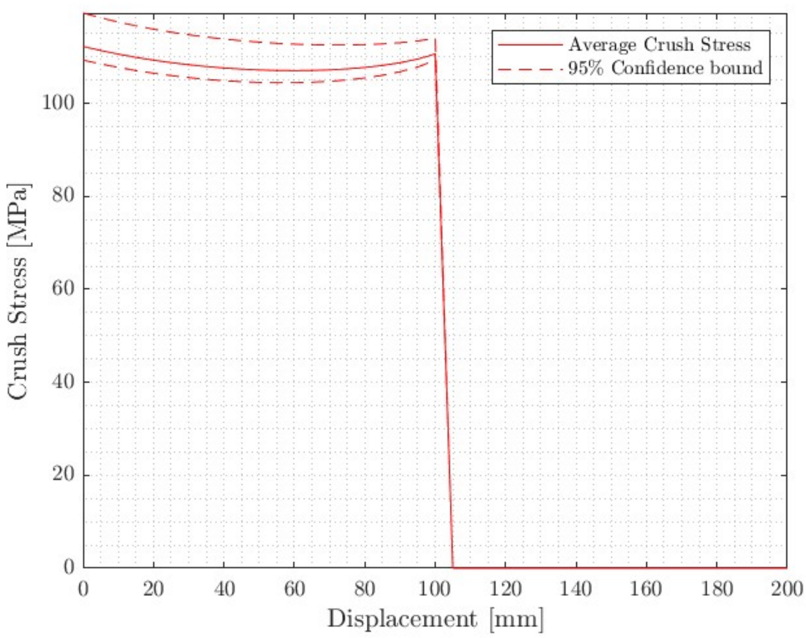


Figure 3.34: Stress-displacement curve of a dynamic, 8-ply carbon fibre cone simulation

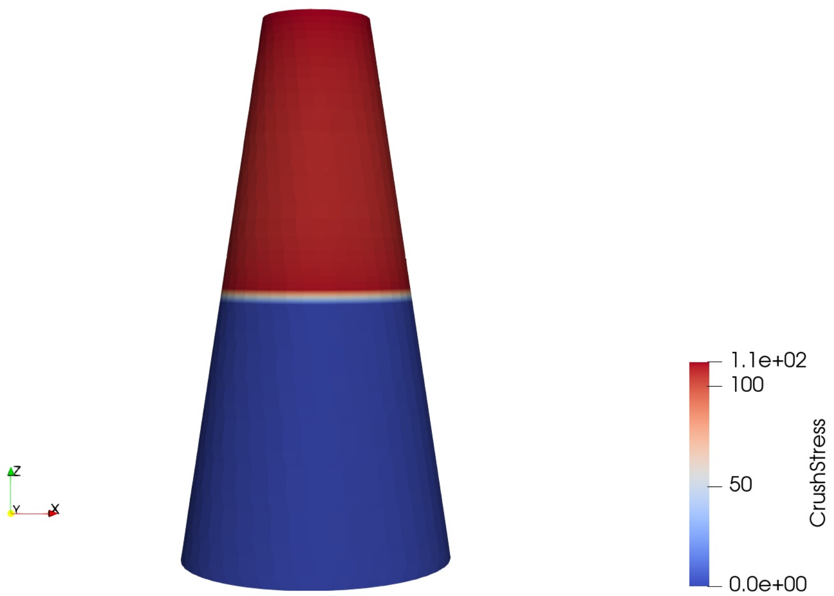


Figure 3.35: 3D crush stress field for a dynamic, 8-ply carbon fibre cone simulation

The velocity-displacement graph shows that the crushing head's velocity exponentially decreases until it stops moving after 105 mm of displacement. Since the system's total mechanical energy remains constant throughout the crushing test, the energy absorbed by the structure being crushed corresponds to the decrease in the crushing head's kinetic energy. The kinetic energy is a function of the square of the velocity, meaning that a linear reduction in kinetic energy would translate to a highly exponential decrease in the velocity. The curve obtained matches the expected behaviour, and the program seems to update the velocity correctly.

The crush stress plot (Figure 3.34) also changed to reflect the effect of velocity on the crush stress. Comparing the dynamic crush stress plot with the quasi-static results (Figure 3.26), it can be seen that not only is the dynamic average crush stress lower at each cross-section, but the overall shape of the graph changed. In the static simulation, the crush stress continuously decreases with displacement due to an increase in RoC. However, in the dynamic simulation, not only is the RoC effect present, but the constantly reducing velocity of the crushing head also gradually increases the average stress, resulting in a "belly" shape in the crush stress-displacement curve.

Additionally, note how in Figures 3.33, 3.34 and 3.35 once the velocity reaches zero, all the subsequent parameters were assigned the value of zero since the crush stress ended at 105 mm displacement.

3.5.4.5 Output Table and Output Log

After the execution of the `CrushStress.m` function, the code compiles all the information used to create the displacement plots into a table called *OutputTable* and exports it into an Excel file called *OutputTable.xlsx*. If a file with the same name already exists when a new simulation is submitted for analysis, the old file is deleted, and a new file is created. The table's format automatically adapts to the specified inputs to include all the relevant data. For example, if the simulation is static, no information on the velocity-displacement data is included in the *OutputTable*; however, if the simulation is dynamic, this data is automatically included. Figure 3.36

shows an excerpt of the *OutputTable* of the dynamic cone simulation used to verify the velocity update code.

1	2	3	4	5	6	7	8	9
Displacement [mm]	Force [KN]	Force Lower Bound [KN]	Force Upper Bound[KN]	Crush Stress [MPa]	Crush Stress Lower Bound [MP	Crush Stress Upper Bound[MP	Ply count	Crushing Head Velocity [m/s]
0	33.7863	32.9050	35.9490	112.2457	109.3177	119.4306	8	10
5	34.7810	33.8788	36.9782	111.3737	108.4846	118.4095	8	9.8296
10	35.7913	34.8663	38.0248	110.5966	107.7383	117.4983	8	9.6511
15	36.8091	35.8601	39.0792	109.9077	107.0740	116.6861	8	9.4638
20	37.8443	36.8703	40.1511	109.3012	106.4882	115.9638	8	9.2673
25	38.8893	37.8902	41.2317	108.7726	105.9780	115.3241	8	9.0608
30	39.9540	38.9298	42.3303	108.3181	105.5413	114.7604	8	8.8436
35	41.0315	39.9830	43.4388	107.9350	105.1769	114.2677	8	8.6148
40	42.1315	41.0600	44.5666	107.6214	104.8844	113.8416	8	8.3732
45	43.2481	42.1558	45.7061	107.3762	104.6642	113.4789	8	8.1178
50	44.3913	43.2809	46.8667	107.1992	104.5178	113.1771	8	7.8469
55	45.5561	44.4315	48.0418	107.0916	104.4479	112.9348	8	7.5587
60	46.7535	45.6194	49.2411	107.0554	104.4584	112.7514	8	7.2511
65	47.9801	46.8427	50.4590	107.0942	104.5555	112.6275	8	6.9212
70	49.2482	48.1152	51.7062	107.2139	104.7473	112.5649	8	6.5655
75	50.5576	49.4390	52.9789	107.4228	105.0461	112.5675	8	6.1790
80	51.9207	50.8294	54.2859	107.7335	105.4692	112.6413	8	5.7554
85	53.3541	52.3070	55.6387	108.1652	106.0424	112.7966	8	5.2851
90	54.8704	53.8907	57.0409	108.7486	106.8071	113.0504	8	4.7537
95	56.5089	55.6304	58.5187	109.5368	107.8340	113.4326	8	4.1364
100	58.3260	57.6030	60.1015	110.6364	109.2650	114.0043	8	3.3852
105	60.4892	60.0298	61.8877	112.3325	111.4794	114.9297	8	2.3721
110	0	0	0	0	0	0	0	0
115	0	0	0	0	0	0	0	0
120	0	0	0	0	0	0	0	0
125	0	0	0	0	0	0	0	0
130	0	0	0	0	0	0	0	0
135	0	0	0	0	0	0	0	0
140	0	0	0	0	0	0	0	0

Figure 3.36: Excerpt of an output table for a dynamic, 8-ply carbon fibre cone simulation

Once the tool finishes running, an output log is automatically printed into MATLAB's command window to provide the user with an overview of the simulation. The log includes a summary of the simulation's settings chosen and a quick summary of the structure's crushing performance by calculating and displaying the energy absorbed (EA), the crushed mass, and specific energy absorption (SEA), with confidence bounds included. Like the *OutputTable*, the log's format automatically adapts to showcase the most relevant information in accordance with the inputs specified.

The total energy absorbed during crushing is calculated by integrating the force-displacement curve across the displacement and the lower and upper bounds by integrating the respective bounds. To obtain the crushed mass, the tool calculates the area of each triangle in the `triangles` structure and multiplies the area by the number of plies and the ply thickness to calculate the volume. The laminate's specific mass (specified in the `Density` input variable) is then multiplied by the total volume of all triangles to calculate the crushed mass. The SEA can be obtained by dividing the total energy absorbed by the crushed mass.

Knowing that the magnitude of the cross-product between two vectors is the area of the parallelogram between them, the area of a triangle knowing the 3D coordinates of its vertexes can be calculated according to Equation (3.23), supposing that the triangle vertexes are points A, B and C.

$$Area = \frac{1}{2} \cdot |\vec{AB} \times \vec{AC}| \tag{3.23}$$

Figure 3.37 shows the output log written for the dynamic cone simulation used to verify the velocity update code.

```
Command Window
Elapsed time is 6.521710 seconds.

Stl file models run with corresponding initial number of plies:
    Cone_50_20_200.stl ,with an inital ply count of 8 plies

Analysed at 5 mm intervals

Type of analysis: Dynamic
    Drop Tower: Off
    Impactor mass: 100 kg
    Initial impact velocity: 10 m/s
    Final crushing head velocity: 0 m/s

Ply count optimization: Off

Crushed mass: 0.070748 kg
Energy Absorbed [KJ]: 4.94 (4.83, 5.18)
Specific Energy Absorbed [KJ/kg]: 69.78 (68.27, 73.24)
fx >>
```

Figure 3.37: Ouput log written for a dynamic, 8-ply carbon fibre cone simulation

Chapter 4

Preparation of a Side Impact Structure CAD Model for Submission Into the Analytical Tool

This chapter focuses on adapting a pre-existing CAD model for an upper Side Impact Structure currently employed in Formula 1 cars into the final model for analytical predictions. To this effect, the effect of partitioning on the tool's accuracy will be analysed to determine how to partition the model. Afterwards, the entire process of partitioning the model will be showcased, and the final discretisations for the RoC and inclinations will be shown.

The details on how a SIS is manufactured will be reviewed in the following chapter from the work by Dalli [18]. For now, it is just necessary to know that the SIS's outer shell and the SIS rib have different laminates, with the outer shell having an 8-ply laminate and the rib a 4-ply laminate. Recall from previous chapters that if the structure has parts with different laminates, each part's STL file must be inputted separately into the analytical tool. Therefore, after partitioning the whole structure, it was necessary to divide the existing SIS CAD model into two separate parts: the rib and the outer shell. Figures 4.1 and 4.2 show the unpartitioned full CAD model utilised and its section view to showcase the internal rib. It should be noted that the SIS's base laminate (dark yellow part in Figure 4.1) is composed of 14 plies; however, since the experimental and numerical tests made by Dalli [18] were terminated before the crushing plate reached the SIS's base, it was not necessary to create another part for the base.

One thing worth noting is that when the model is separated into two STL files, the intersection edge between the rib and the outer shell is lost. The whole structure will still be plotted and calculated correctly, but since the parts are separated, the reinforcement effect at intersection nodes no longer occurs. However, the loss of the intersection reinforcement effect will not have much impact on the final prediction since the effect of one node compared to the whole cross-section is minimal.

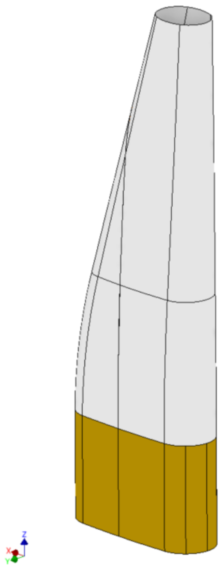


Figure 4.1: SIS CAD model

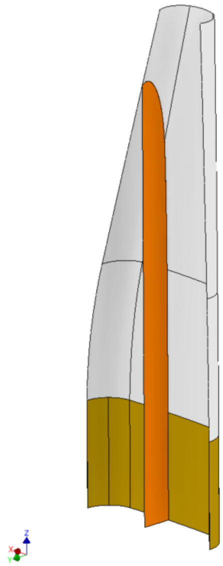


Figure 4.2: SIS half-section view CAD model

4.1 Evaluating the Effect of Partitioning on Accuracy

When creating the STL files required for the analytical tool, it was mentioned that the CAD model must be partitioned along the z-axis using the developed `PartitioningTool.vba` macro and, optionally, along the x and y-axis. This section is focused on evaluating the effect that the partitions made have on the final predictions by showcasing the differences between an x and y partitioned and unpartitioned model and the effects of the size of the partitions on the final results and total runtime.

The SIS model was used for the evaluations since it is the model studied during this work with the highest degree of geometrical complexity, and it is the model from which the analytical predictions obtained will be compared with pre-existing experimental and numerical results. Therefore, it is crucial to understand how the SIS geometry must be partitioned to get the best predictions from the `AnalyticalTool.m` program.

Partitioning along the x and y axes effectively imposes a grid onto the structure where nodes must be created. This is important since it limits the distance between neighbouring nodes and, since the processes of calculating the inclination and radius of curvature use information from the neighbouring nodes, having too large a distance between nodes can create errors, similar to how, in FEM models, having large elements may negatively impact its accuracy.

Two static simulations of the SIS were run to understand the differences between choosing or not to partition along the x and y axes. One of the models was only partitioned every 5 mm along the z-axis, whilst the other was partitioned along the x, y and z axes every 5mm. Since this section aims to capture the partitioning effect and not to present predictions for the real SIS, a single STL file of the whole SIS (outer shell+rib) was submitted for analysis and run with 8-plies so the intersection effect may be captured. The intersection further increases the structure's geometrical complexity and provides a better case for establishing conclusions.

Figures 4.3 and 4.4 show the tesselation of the SIS and the RoC discretisation for the unpartitioned model along the x and y axes. Figures 4.5 and 4.6 plot the same variables for the model

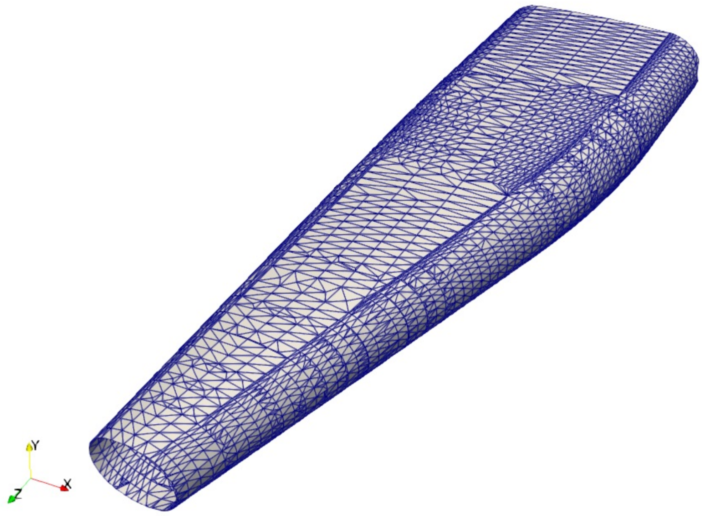


Figure 4.3: Tesselation of the unpartitioned SIS

partitioned along the x and y axes.

From the analysis of the RoC discretisations obtained for both the simulations (Figures 4.4 and 4.6), it is immediately apparent that the discretisation of the RoC was significantly better for the partitioned model, especially on the SIS's flat sections and around the intersection edge. Comparing the tesselations of the two models, (Figures 4.3 and 4.5) note how the nodes in the partitioned model are much smaller and structured than in the unpartitioned model. Since the tool uses the neighbour's information to calculate the RoC, the large distances between nodes in the unpartitioned model cause the RoC to be calculated along a large edge of the cross-section instead of locally, forming large gradients and discontinuities along the structure that affect the final prediction's accuracy.

The calculated SEA of both simulations was used as a baseline to quantify the influence of partitioning along the x and y axes despite its value having no real significance since the simulated structure does not represent the real SIS. It would be expected that since both simulations were run from the same CAD model and under the same conditions (quasi-static with 8-ply carbon fibre laminate), the SEA values obtained would be similar. The obtained value for the partitioned model was 73.59 kJ/kg, whilst for the unpartitioned model was 70.24 kJ/kg, a relative error of 4.55%. However, it should be noted that partitioning the model increased the number of nodes from 4290 to 7290 and the number of facets from 8586 to 14579, which led to an increase in the runtime of 29.914 s, from originally 13.860 s to 35.774 s.

From the analysis, it can be concluded that the model must be well partitioned to minimize errors in the analytical prediction. The distance between partitions should be chosen so that all geometric features, such as changes in curvature or inclination, occur along multiple partitions so that the analytical tool can obtain smooth geometrical transitions and avoid discontinuities. However, choosing a value that is too small for the partitions will create an excessive number of nodes and facets, dramatically increasing runtime. For the final SIS model, used to compare with experimental and numerical results, it was decided to partition every 3 mm along the x and y axes to guarantee a good resolution.

It was already seen in the introduction to Section 3.5.4 that the distance between partitions

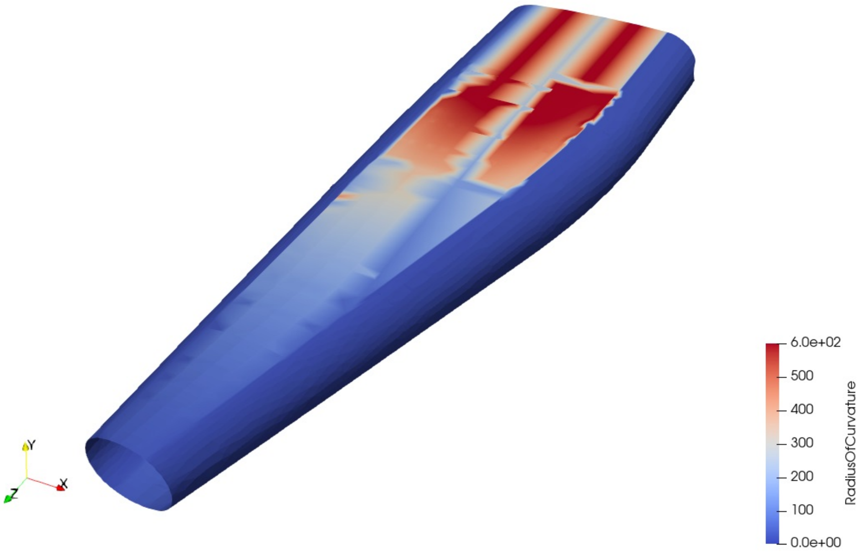


Figure 4.4: RoC discretisation of the unpartitioned SIS along the x and y directions

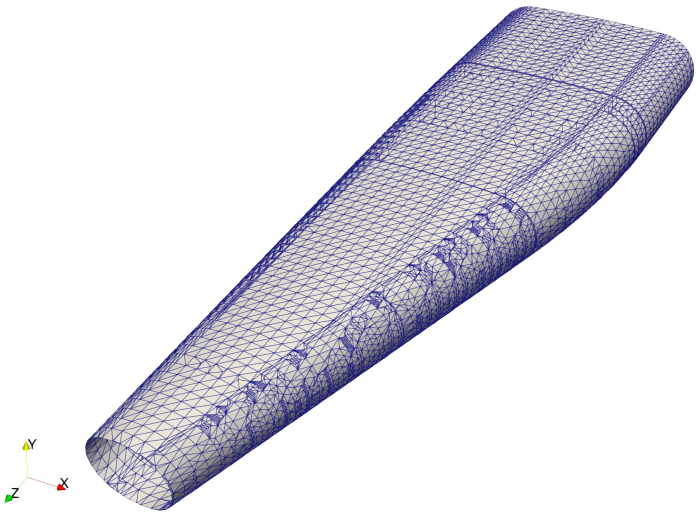


Figure 4.5: Tesselation of the partitioned SIS

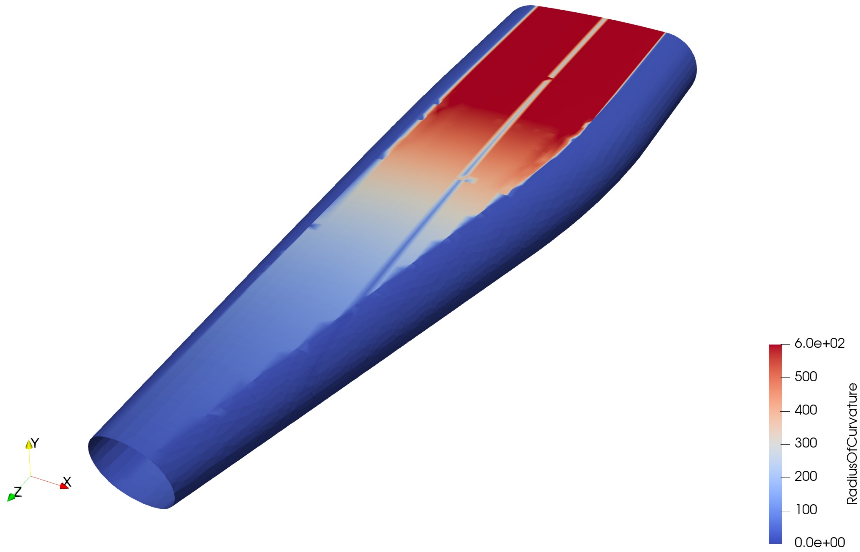


Figure 4.6: RoC discretisation of the partitioned SIS

along the z-axis dictates the interval at which the complete cross-sections needed to calculate the force-displacement curve are created, that is, if the structure is partitioned every 5 mm, then the tool will analyse the structure every 5 mm. Therefore, the value chosen for the distances between z partitions must be small enough to capture cross-sectional changes along the structure's length; however, choosing too small a value will also cause too many nodes to be formed and increase runtime. For the final SIS model, the distance between partitions along the z-axis was defined as 3 mm, a small enough value to accurately capture changes in cross-section along the SIS's length but not small enough to cause excessive runtimes.

4.2 Partitioning the Model

Having determined the values from which to partition the CAD model, this section showcases the process of partitioning the SIS, starting with the CAD model shown in Figures 4.1 and 4.2.

The first step was to create the reference planes (WP0_z, WP0_y, WP0_x) from the origin planes. The WP0_z was created tangent to the SIS' top section by creating a plane parallel to the XY origin plane offset by a distance of 207.604 mm. The WP0_y reference plane was created tangent to the most y-negative surface of the SIS by creating a plane parallel to the XZ origin plane offset by a distance of 52.835 mm. Finally, the WP0_x reference plane was created tangent to the most x-negative surface of the SIS by creating a plane parallel to the YZ origin plane offset by a distance of 23.755 mm. Figure 4.7 shows the CAD model after creating the reference planes.

With the reference planes defined, the `PartitioningTool.vba` macro was executed to partition the model along the x, y and z axes. All the input parameters chosen are documented in the list below. Figures 4.8 and 4.9 show the CAD model after the macro finished running with and without the partitioning planes.

1. Partitions along the z axis:
    - **Partitions Offset**: 3 mm

4.2. Partitioning the Model

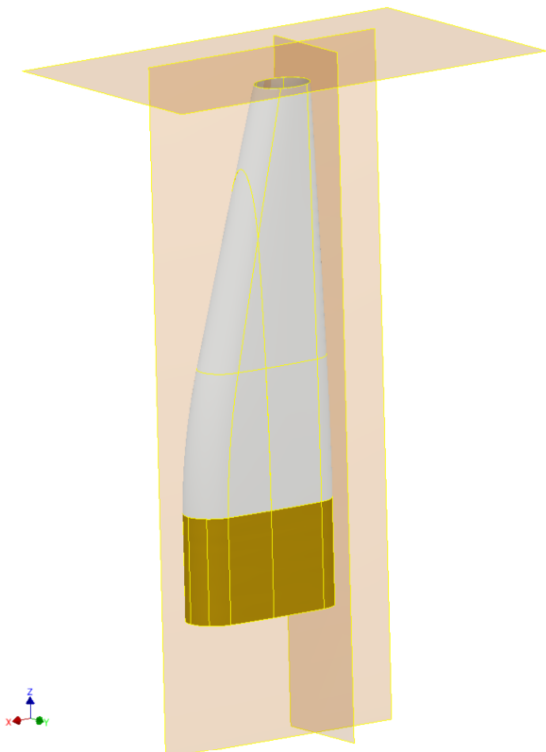


Figure 4.7: SIS CAD model showing the created reference planes

- **Distance to final partition**: -345 mm
- **Number of surfaces to partition**: 2 (Rib + Outer Shell)

2. Partitions along the y axis:
   - **Partitions Offset**: 3 mm
   - **Distance to final partition**: 34 mm
   - **Number of surfaces to partition**: 1 (Rib)
3. Partitions along the x axis:
   - **Partitions Offset**: 3 mm
   - **Distance to final partition**: 111 mm
   - **Number of surfaces to partition**: 1 (Outer Shell)

After partitioning, the model can be saved onto two STL files, one for the outer shell and another for the rib. To save the outer shell, the visibility of the boundary patch forming the rib, denominated "Boundary Patch 3", was hidden, leaving only the outer shell showing. The resulting geometry was then exported as an STL file with the Brep definition enabled. To save the rib's STL, the boundary patch's visibility was re-enabled, and the Surface Body feature containing the outer shell geometry, denominated "Srf13", was hidden. The resulting geometry was then exported as an STL file with the Brep definition enabled.

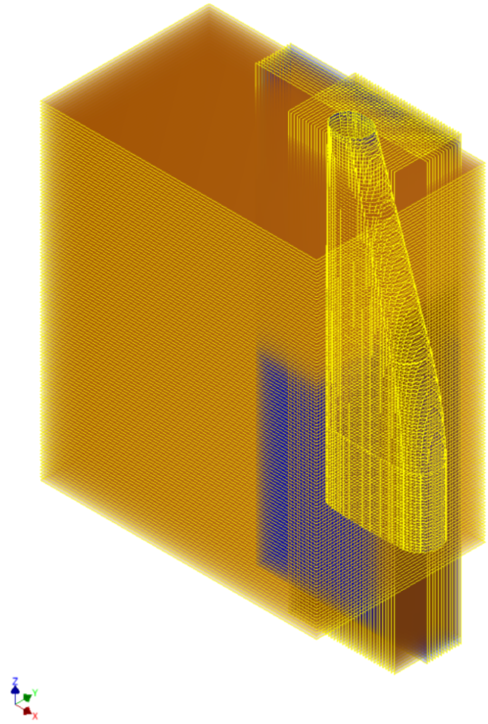


Figure 4.8: Partitioned CAD model showing the partitioning planes

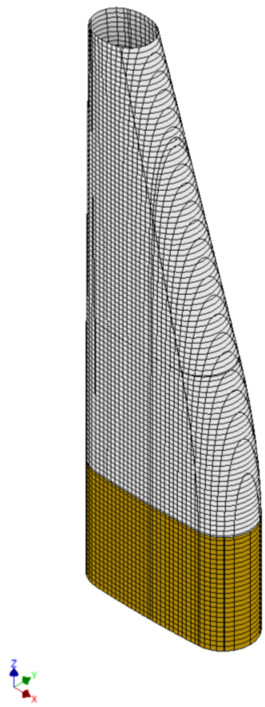


Figure 4.9: Partitioned CAD model

4.3 Final SIS Model Discretisations

Having obtained the final model's STL files for comparing the analytical tool's results with the numerical and experimental results, they can be inputted into the `AnalyticalTool.m` program and a simulation submitted to observe the discretisations obtained. The simulation parameters are irrelevant since the RoC and inclination discretisations depend only on the geometry, and the values assigned will be the same if the simulation is quasi-static or dynamic. Figures 4.10 and 4.11 show the full view and section view of the RoC discretisation, and Figures 4.12 and 4.13 show the inclination discretisation.

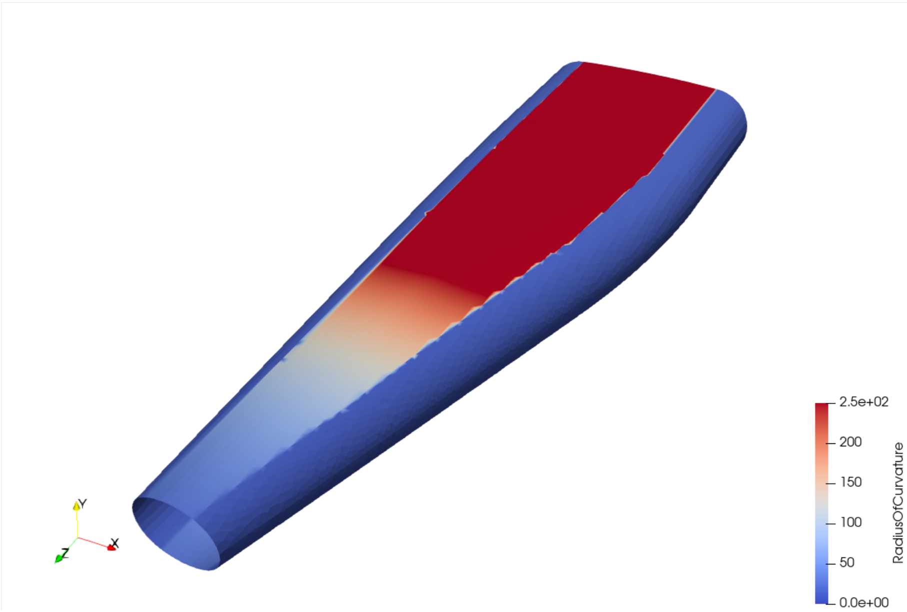


Figure 4.10: RoC discretisation for the final SIS model

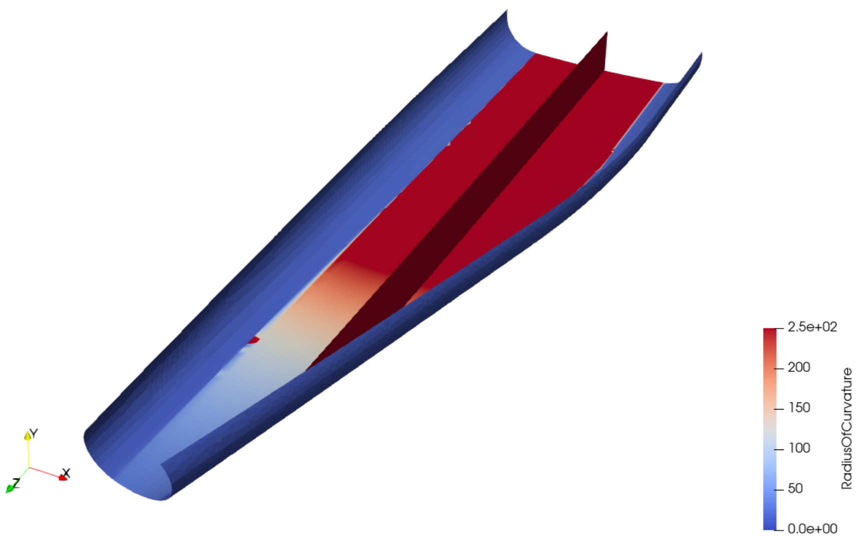


Figure 4.11: RoC discretisation for the final SIS model: section view

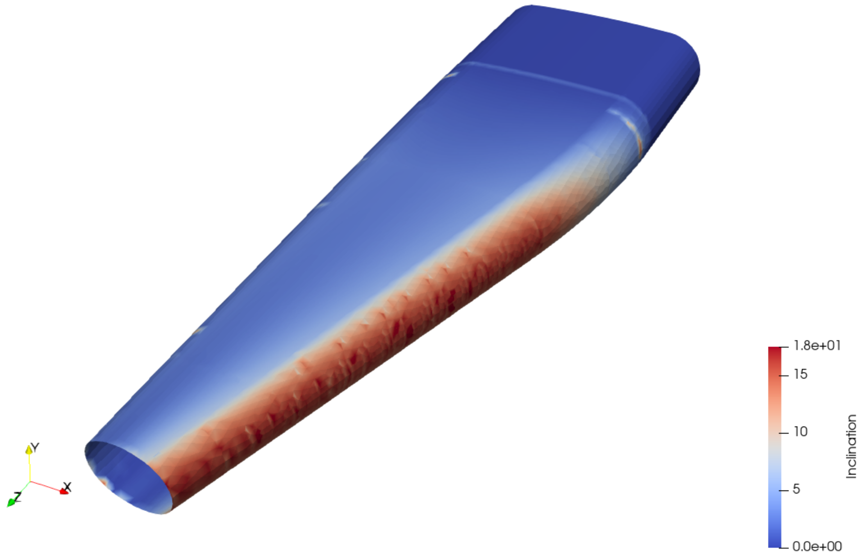


Figure 4.12: Inclination discretisation for the final SIS model

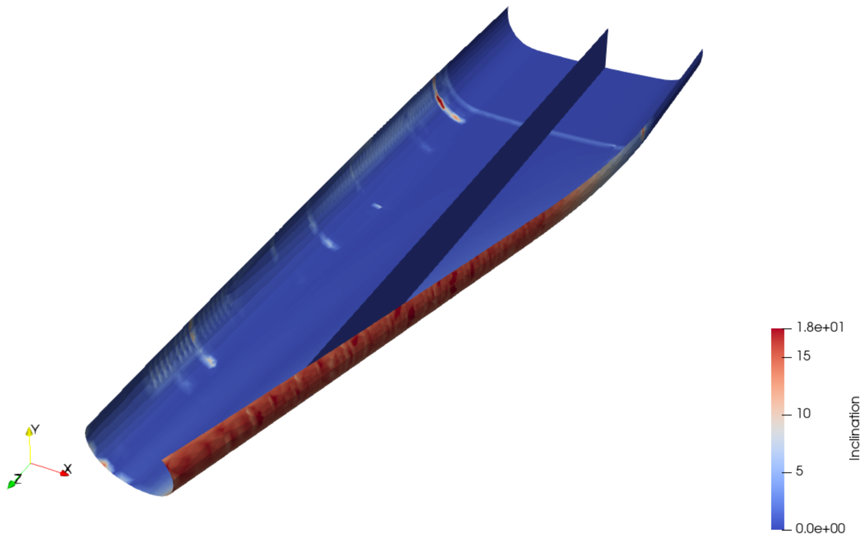


Figure 4.13: Inclination discretisation for the final SIS model: section view

From Figures 4.10 to 4.13, it can be seen that the tool was able to accurately calculate and assign values for the RoC and inclination throughout the model's nodes and create a smooth distribution. Some discontinuities can be observed for both the inclination and the RoC in the backside of the SIS, which results from highly concentrated clusters of nodes creating localized singularities. However, these discontinuities either fall outside of the z partitions and do not influence the analytical prediction or their influence is so localized that it does not alter the predicted result in any significant way. Should problems be observed at this stage, the whole CAD model must be re-partitioned, or some of the x and y partitions could be deleted and new STL files made.

Chapter 5

SIS Analytical Results and Discussion

This chapter presents the analytical predictions obtained using the developed Analytical Tool for the upper Side Impact Structure (SIS) employed in Formula 1 car. The analytical predictions will be validated against experimental and numerical results obtained from previous research. Dalli [18] conducted a comprehensive study of quasi-static and dynamic experimental tests on a carbon fibre SIS. The author also performed numerical simulations using a macro-scale model in ABAQUS for static and dynamic crush tests. The carbon fibre material used in the experiments and simulations is the IM7-E745 5HS CFRP laminate, whose crush stress parameter curves were obtained in Section 3.4.5.1.

Furthermore, recent unpublished internal work by Dalli involved a quasi-static test of a hybrid SIS using the same testing methodology as the carbon structure. The results of this test will also be compared to the analytical tool's predictions. The hybrid material used in this test is the same laminate studied by Moreira [8], whose crush stress parameter curves were obtained in Section 3.4.5.2.

5.1 Review of the Experimental and Numerical Data

Before presenting the results from the analytical tool, it is first essential to define how the experimental and numerical results were obtained by Dalli to establish a baseline for comparison. To this effect, a review of the relevant sections of Dalli's work [18] will be presented. Readers interested in more detailed information can refer to the original work.

5.1.1 SIS Geometry and Manufacturing

Balanced 5HS plies (equal mechanical properties in the 0° and 90° direction) were used to form the laminates in the SIS. The structure comprises 14 plies at its base, tapering to 8 plies within its crushing region. The layup at the base is $[+45°/0°/+45°/-45°/-45°/+45°/0°/0°/-45°/90°/90°/45°/0°/+45°]$, and plies 2 to 7 are drop-off plies that are gradually tapered down, effectively forming a $[+45°/0°/-45°/90°]_S$ laminate at the crushing region with a nominal thickness of 2.4 mm. The SIS's inboard and outboard layups, complete with ply numbers, are displayed in Figure 5.1a. Note that since the 5HS weave used is balanced, plies have the same properties in the 0° and 90°, so the $[+45°/0°/-45°/90°]_S$ laminate at the SIS's outer shell has the same

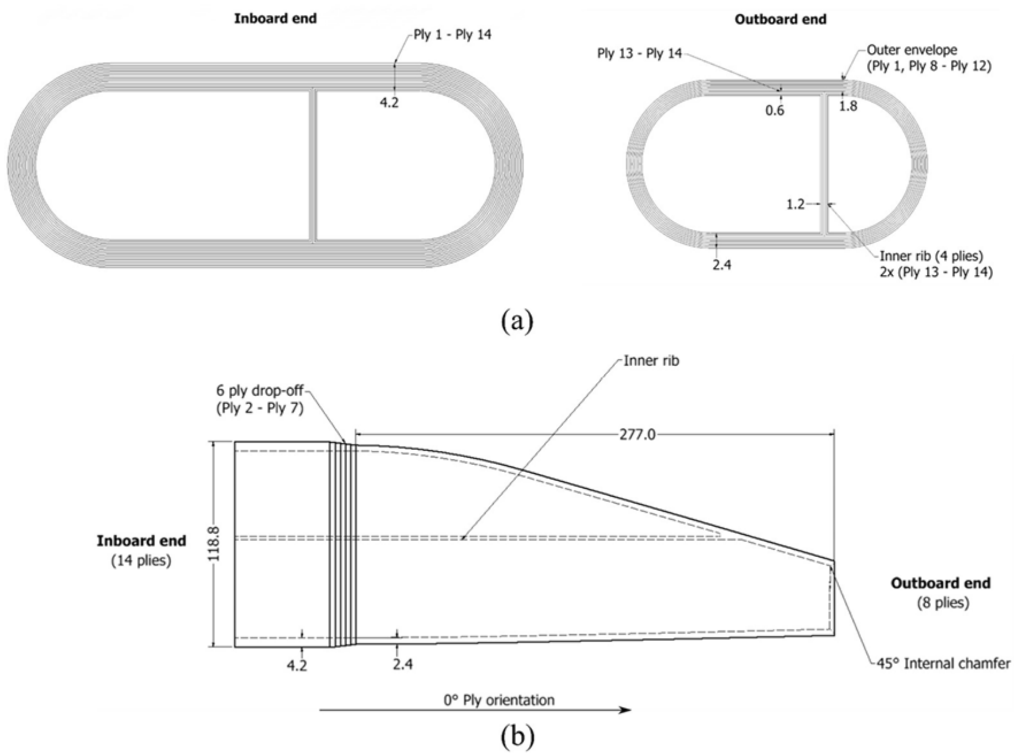


Figure 5.1: (a) Representative layup schematics of the inboard and outboard ends of the SIS; (b) cross-sectional view of the SIS, including the 6-ply drop-offs (all linear dimensions are given in mm) [18].

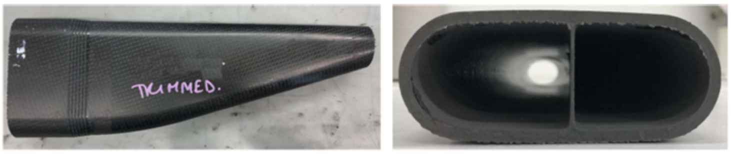
mechanical properties than the $[+45°/0°/+45°/0°]_S$ laminate used in the tubular coupons to obtain the crush stress parameter curves. The reason for rotating the plies by 90° was to improve their drapability.

The internal rib, made of plies 13 and 14, is also visible in Figure 5.1. Two internal male moulds are the foundation for the SIS layup, around which plies 14 and 13 are draped. After that, these two moulds are combined to create a four-ply inside rib. Afterwards, the final 12 plies are layered on top of plies 13 and 14 and wrapped around the joint moulds. Figure 5.1b shows the six-ply drop-off at the structure's base. A 45° chamfer was machined into the SIS's top end to promote progressive crushing, and the structure was mounted into a plastic base to allow the structures to be clamped in the experimental setups and the structure was fixed onto the base using an epoxy adhesive. Figure 5.2a shows the side and bottom views of the manufactured SIS, and Figure 5.2b shows it mounted onto the plastic base.

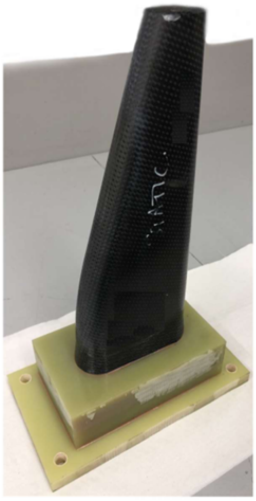
5.1.2 SIS Quasi-static and Dynamic Tests

The SISs were quasi-statically crushed on an electromechanical test machine with a 250 kN load cell and a 25 mm/min displacement-controlled loading rate. After 250 mm of crushing, the testing was terminated. Figure 5.3 depicts the quasi-static test configuration for the SIS. Three SISs were quasi-static tested.

A 780 kg sled-impactor (see Figure 5.4b) was employed to dynamically crush the SIS, with a 10 m/s target impact velocity (the impact velocity required to obtain F1 certification). Due to the expenses associated with every crush test, two SISs were crushed at once. Figure 5.4a depicts the two SIS experimental setups, complete with the metal support structure and plastic mountings.



(a)



(b)

Figure 5.2: Side Impact Structure: (a) front and bottom views; (b) structure mounted in a plastic base [18]

Figure 5.3: Quasi-static test setup for the Side Impact Structure [18]

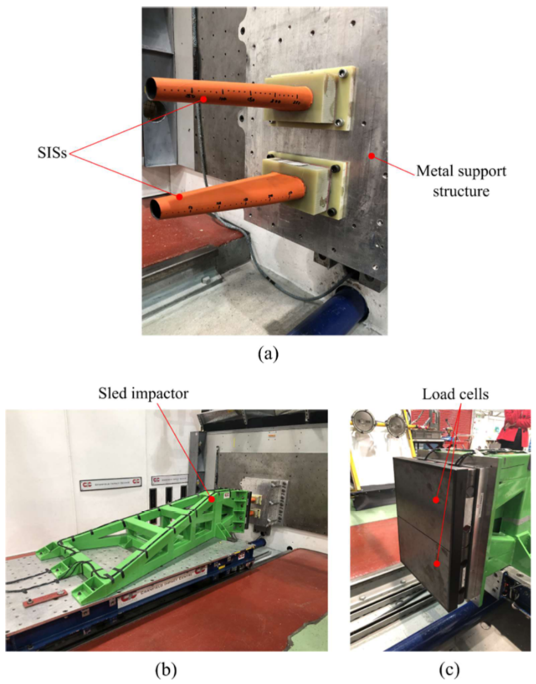


Figure 5.4: Dynamic setup for Side Impact Structure (SIS) crush test: (a) two SIS mounted on the metal support structure; (b) 780 kg sled-impactor; (c) two separate load cells to measure the individual crushing loads of both SIS. [18]

The crushing force for each SIS was measured using separate load cells with a 20 kHz sampling frequency (see Figure 5.4c).

Figure 5.5 shows the force-displacement graphs obtained for the three quasi-statically tested SISs. The gradual increase in load and the lack of a high peak force indicate that the SISs were progressively crushed, as is expected for crash structures. All three SISs exhibited similar responses. However, the load response for the third SIS (QS-3) dropped to 35 kN. The drop in force can be attributed to the test interruptions for photography, which involved unloading and reloading the structure. After resuming the test, a localised buckling failure in some outer plies was observed, which explained the load drop. Notably, the other two SIS tests continued uninterrupted up to 250mm of crushing without any signs of such failures. Therefore, the load drop in QS-3 was likely caused by the stress redistribution created by the unloading and reloading cycle during the test. Additionally, a gradual increase in crushing force can be seen along the displacement due to the increasing cross-sectional area. Figure 5.6 shows the stress-displacement curve for the three SIS quasi-static tests, which were seen to be decreasing with the displacement, which was attributed to the increase in the average radius of curvature along the SIS.

Figure 5.7 compares the load responses of two dynamic tests with the results from three quasi-static tests. Similar to the quasi-static results, neither dynamic test exhibits a significant peak force, and their responses are relatively consistent. However, after exceeding the elastic response phase, the loads measured in the dynamic tests consistently fall below those of the quasi-static tests throughout the entire 250 mm of displacement. This decrease translates to a reduction in crushing stress, which is presented in Figure 5.8. During the 250 mm of progressive crushing, the impacting mass's velocity drops from roughly 10 m/s to 5 m/s.

In recent unpublished internal work by Dalli, using the same methodologies previously described, an SIS whose outer shell is made of a hybrid laminate and internal rib of carbon was

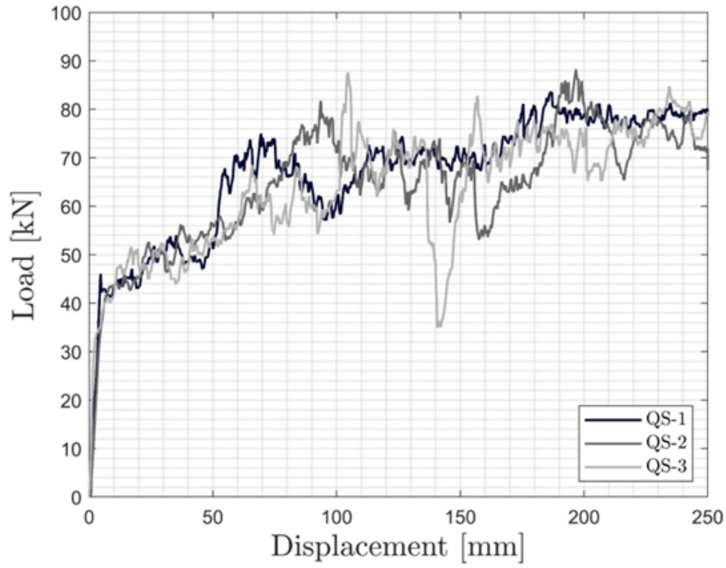


Figure 5.5: Force-displacement curves for the quasi-static tests of three Side Impact Structures. [18]

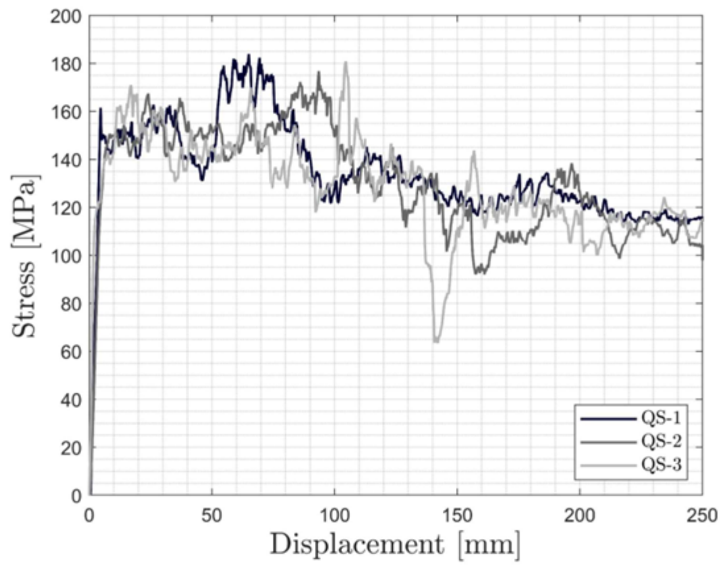


Figure 5.6: Stress-displacement curves for the quasi-static tests of three Side Impact Structures. [18]

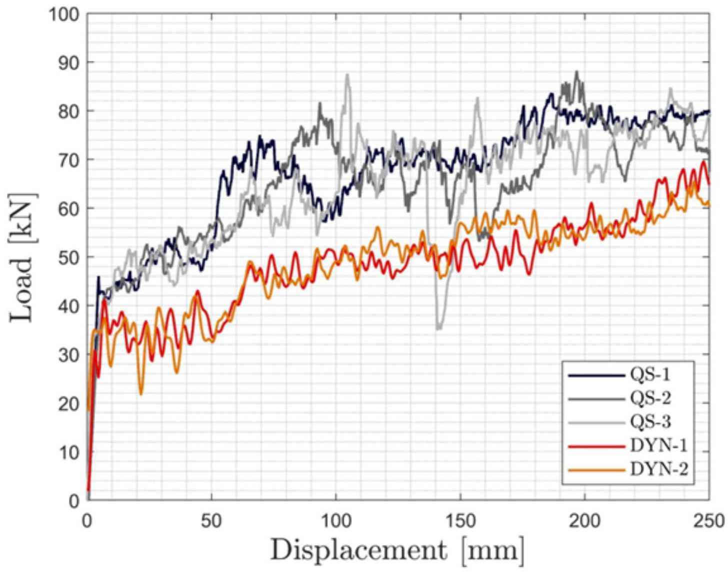


Figure 5.7: Force-displacement curves for two dynamic SIS tests overlaying three quasi-static tests. [18]

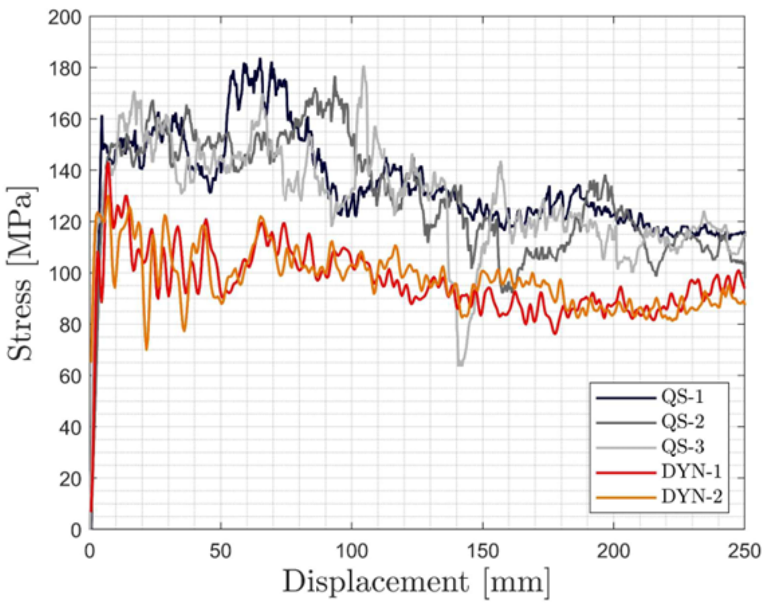


Figure 5.8: Stress-displacement curves for two dynamic SIS tests overlaying three quasi-static tests. [18]

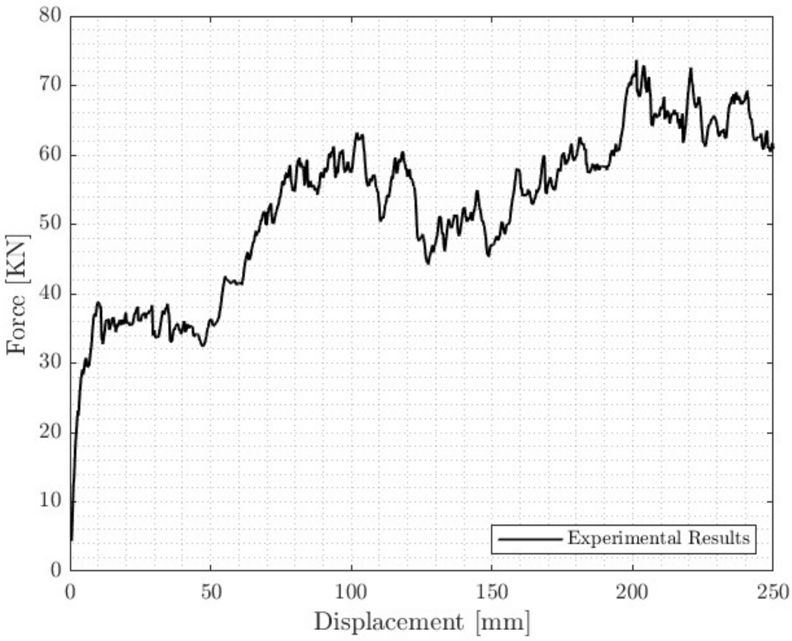


Figure 5.9: Force-displacement curve of a hybrid SIS

manufactured and quasi-statically tested. So far, only one structure has been tested, and the force-displacement graph obtained can be observed in Figure 5.9.

5.1.3 SIS Macro-scale FEM Model

The CAD model for the macro-scale analysis was constructed from the outer surface of ply 13. The SIS was modelled within Abaqus® using a built-in composite damage material model. This material model incorporates damage mechanisms to represent the SIS's behaviour under impact accurately. To simulate the progressive crushing of the SIS during a collision, the CZone®add-on program within Abaqus®was employed. CZone®allows for the definition of a crush stress parameter within the material model. Elements ahead of the crushing surface are progressively removed from the simulation when the stress exceeds this defined value. This technique effectively replicates the crushing behaviour observed during real-world impacts. Quadrilateral reduced-integration shell elements (S4R) were utilised to capture the SIS's response at the laminate level. The crushing plate was modelled as a discrete rigid body and an encastre-type boundary conditions was assigned to where the SIS was experimentally fixed within the mounting base.

For the quasi-static simulation, a smooth step velocity profile was applied to the crushing plate. The system's kinetic energy was closely monitored and maintained near zero to ensure the quasi-static condition and eliminate the inertial effects. In contrast, the dynamic simulation aimed to replicate the experimental conditions. The crushing plate was assigned half the mass of the sled impactor (390 kg) and given the same initial velocity of 10.1 m/s observed in the dynamic experiment. This approach captures the dynamic effects, such as inertia, which are crucial for analysing impacts by including the mass and initial velocity.

The assignment of the crush stress parameter to the model's surface made on Dalli's work was made subjectively by dividing the CAD model into flat and curved sections and assigning to the curved sections the crush stress obtained from tubular coupon testing and to the flat sections the crush stress from the flat coupons testing. However, in recently done internal work, Dalli

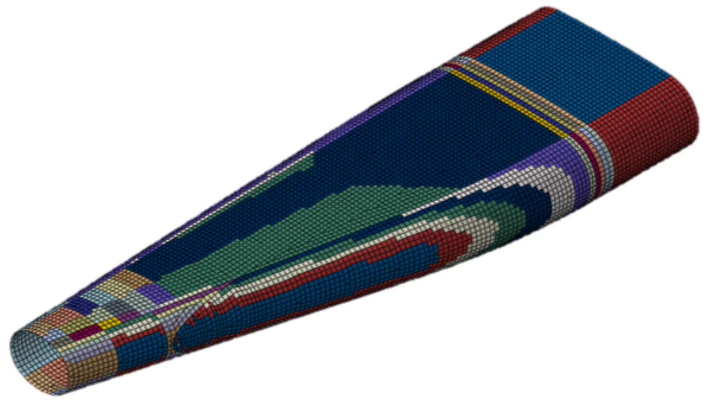


Figure 5.10: SIS RoC discretised in 5% increments of crush stress

developed a scrip that discretised the SIS's surface into several bands of radii of curvature and assigned each band a crush stress value according to the carbon discretized RoC curve obtained in section 3.4.5.1 (Figure 3.13). In Figure 5.10, the several assigned values for the RoC can be seen. For the dynamic simulation, all the assigned bands were the same, but the crush stress value for each band was multiplied by 0.84, which corresponds to the calculated velocity factor (F_V, obtained from Equation (3.13)) for the mean velocity during the experimental test. Since the crush stress assignments on the analytical tool and in the ABAQUS FEM model are based on the same RoC crush stress parameter curve, a direct comparison between the results can be made.

Figures 5.11 and 5.12 show, respectively, the simulation results of the FEM models for the quasi-static and dynamic scenarios.

5.2 Evaluating the Tool's Computational Performance

This section showcases how the computational time for a simulation varies with the total number of facets in the inputted STL files. A classic convergence analysis cannot be performed by increasing the partition size to evaluate the effect on the results or computational time. As was seen in the previous chapter, increasing the partitioning intervals past a certain point causes the RoC and inclination discretisations to become wildly inaccurate, making the obtained results unrepresentative of the inputted structure and, therefore, meaningless.

One alternative would be to keep the x and y partitioning distances and alter the z distances. However, such an approach has its problems. Increasing the size of the partitions along the z axis will create fewer cross-sections. Consequently, the displacement graphs will have fewer points to plot the curve. However, since the geometric discretisation is still the same (the x and y partitioning distance was not altered), the predictions obtained would remain the same. For example, increasing the z partitioning distance from 3 mm to 6 mm would cause the displacement graphs for the 6 mm simulation to have half as many points. However, the points sharing a common displacement (every 2 points in the 3mm graphs) would still have the same predictions since the predictions are only dependent on the geometric discretisations, which are the same for both simulations.

In addition, increasing the z partitioning distance, despite creating fewer nodes, only reduces the computational time until a certain point. The STL tesselation process still needs to populate the

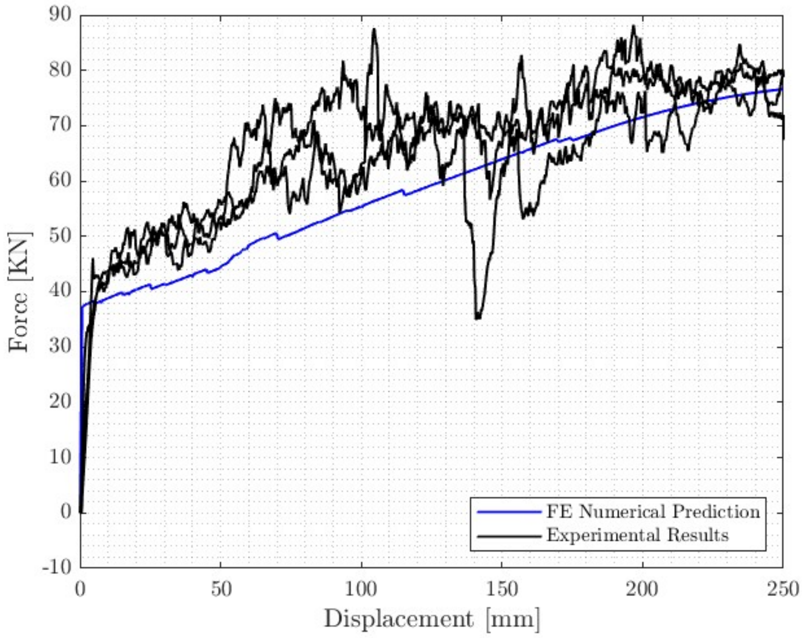


Figure 5.11: Force-displacement curve for a carbon SIS FEM quasi-static simulation with the quasi-static experimental results from [18] overlayed.

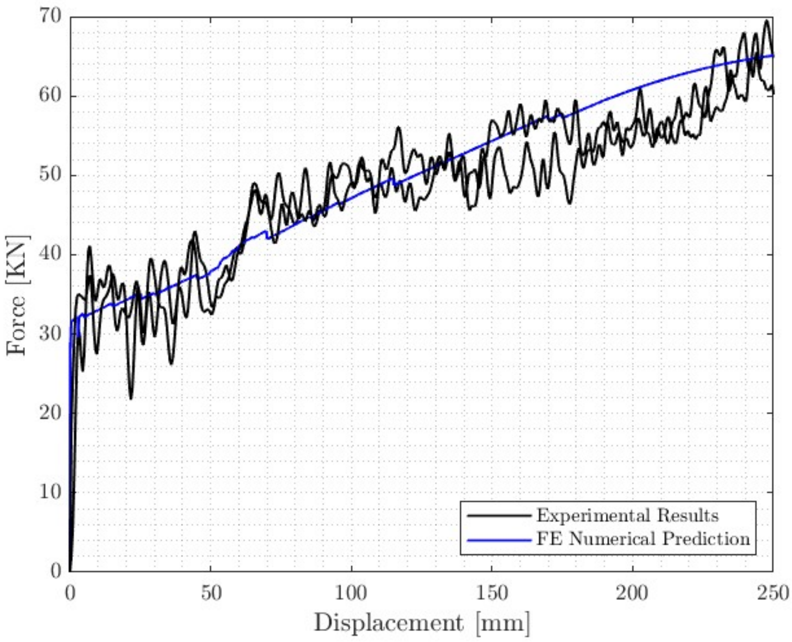


Figure 5.12: Force-displacement curve for a carbon SIS FEM dynamic simulation with the dynamic experimental results from [18] overlayed.

unpartitioned areas with nodes to accurately approximate the structure's surface. It was observed that the lack of partitions in the curved surfaces could cause the tesselation process to create an excessive amount of unorganized nodes, leading to computational times greater than for cases with smaller partitioning sizes. For these reasons, it was determined that performing a convergence analysis would be meaningless due to the tool's operating principles and dependence on the STL tesselation process.

However, from the process described above, it can be seen that the tool's computational runtime is dependent only on the total number of unique nodes and, consequently, the total number of triangles/facets. Therefore, to plot the change in runtime with the total number of nodes, several static simulations were run for several CAD models (SIS, cones, tubes...), and for each one, the runtime and number of nodes and triangles were recorded. Table 5.1 details the information for the simulations performed, and Figures 5.13 and 5.14 show, respectively, the variation of the number of triangles with the number of nodes and the variation of runtime with the number of nodes.

Table 5.1: Simulation data for the performance analysis

Number of nodes	Number of triangles	Runtime [s]
385	700	1,98
429	780	2,09
1546	3018	3,43
1842	3597	4,19
2814	5541	7,18
7290	14579	27,6
12186	23999	59,6

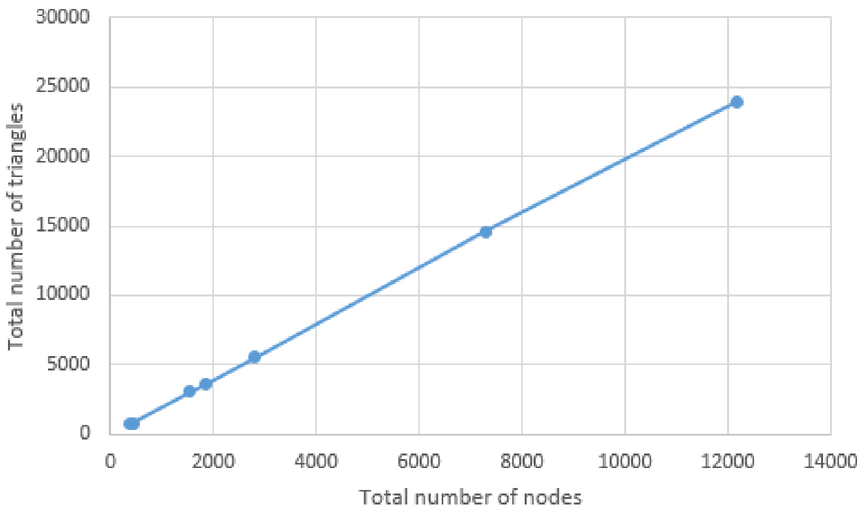


Figure 5.13: Plot of the variation of the number of triangles with the number of nodes

5.3 SIS Analytical Results

This section presents the results obtained from the AnalyticalTool.m program for the final SIS model obtained from Chapter 4. The force-displacement graph for each simulation will be directly compared with the available experimental and numerical results showcased in Section 5.1. Note that the SIS total length is 345mm, both for the CAD model and the real structure. Still, since the experimental and numerical tests were stopped after 250mm (where the SIS's base starts), the

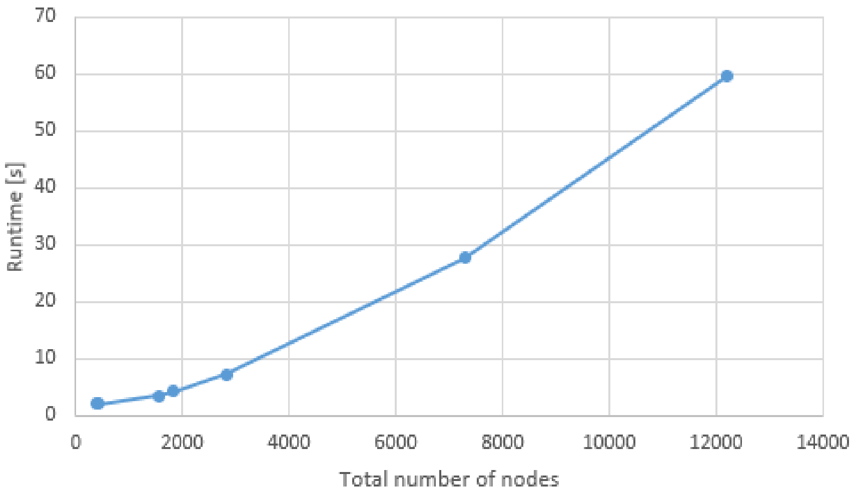


Figure 5.14: Plot of the variation of runtime with the number of nodes

force-displacement graphs from the analytical tool were also stopped after 250mm despite the force for the entire structure having been calculated.

In the previous chapter, the SIS CAD model was saved under two STL files named *SIS_OuterShell.stl* and *SIS_Rib.stl*, each containing the STL tesselations for the SIS's outer shell and rib, respectively. This was done since both parts have laminates with different numbers of plies and must be inputted separately into the analytical tool. For all the simulations conducted, the SIS's outer shell was run with eight plies and the SIS's rib with four plies to represent the real-world structure accurately.

5.3.1 Carbon Fibre SIS

This subsection focuses on the analytical predictions obtained for a carbon fibre SIS model. The simulations were run using the crush stress parameter curves obtained in Section 3.4.5.1 for the IM7-E745 5HS CFRP laminate.

5.3.1.1 Quasi-static Simulation

A quasi-static simulation for a carbon fibre SIS was submitted for analysis. To perform such a simulation, the relevant parameters in the **Simulation Control Inputs** input section were defined according to the list below. The reader is referred to Section 3.5.1 for the complete input control for the `AnalyticalTool.m` program.

- `DynamicAnalysis`=false;
- `optimization`=false;

Figure 5.15 shows the obtained force-displacement graph for the quasi-static carbon fibre SIS simulation and the numerical FEM macro-scale results obtained by Dalli. Similarly, Figure 5.16 shows the obtained force-displacement graph and its confidence bounds compared against the three quasi-static experimental tests performed by Dalli [18].

Figure 5.15 shows that the analytical tool accurately replicates the deterministic results from the FE results. The analytical simulation took 59.616s to produce results, whilst the FE simulation took 6 hours. However, the analytical prediction cannot capture failure from elastic instabilities

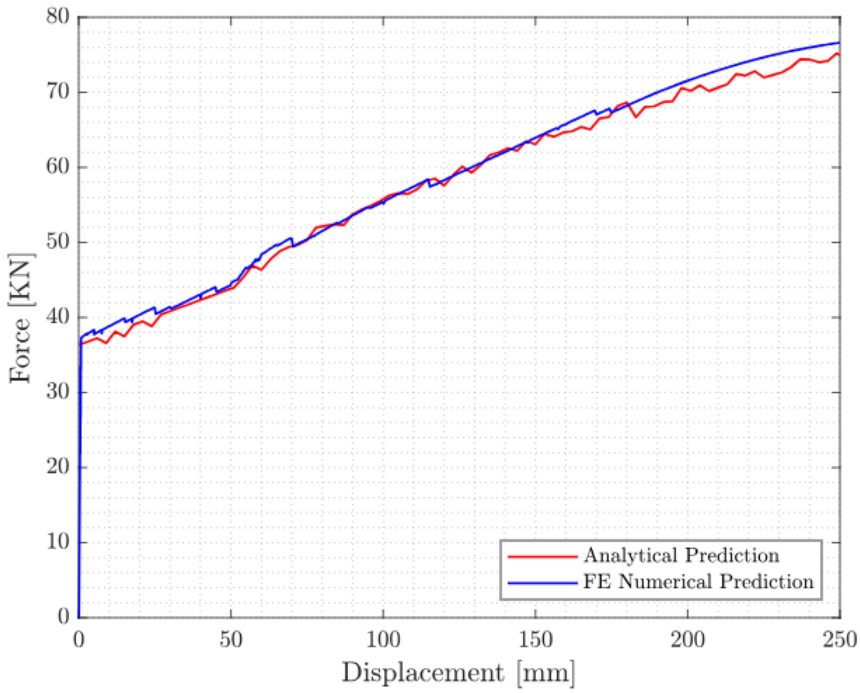


Figure 5.15: Force-displacement curves comparing the quasi-static predictions from the analytical and numerical models

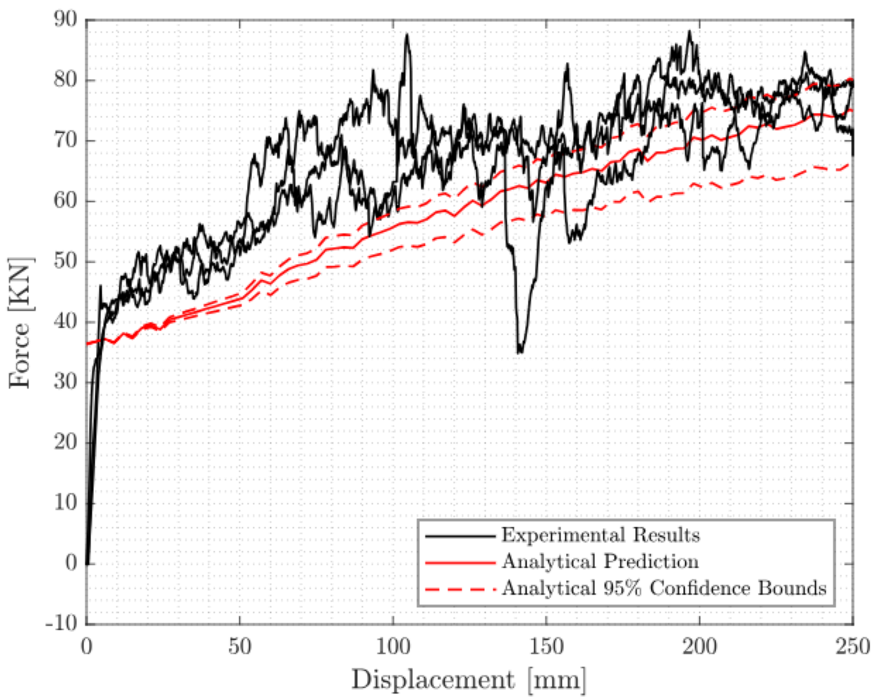


Figure 5.16: Force-displacement curves comparing the quasi-static predictions from the analytical model and the experimental results obtained by Dalli [18]

(buckling) and always assumes progressive crushing occurs. Therefore, despite being able to quickly and accurately reproduce the FE results, which is helpful for the early design stage, a numerical simulation must still be performed at a later design stage to guarantee that the structure does not fail prematurely.

From Figure 5.16, it can be seen that the tool can capture well the progressive increase in crush force over the displacement. However, the tool continuously under-predicts the crushing force compared to the experimental results. Specifically, the tool under-predicts the experimentally obtained force by approximately 13% at the start of the test, and the relative error decreases to almost 0% at the end of the crush test. Note that the tool's accuracy depends on experimental coupon data to define the crush stress parameter curves accurately, and there was not enough data to determine the inclination and crush stress parameter curves. Therefore, it is possible for the prediction to become more accurate after further coupon testing is conducted and the remaining curves are obtained.

Both the analytical tool and the numerical FEM model underpredicted the force required to crush the structure (see Figure 5.11). Since the force-displacement curves for the analytical and numerical models are in agreement, it can be said that the analytical tool at the current stage of development can match the current industry-used FEM models in its ability to predict the crushing behaviour.

5.3.1.2 Dynamic Simulation

A dynamic simulation for a carbon fibre SIS was submitted for analysis. The submitted simulation aims to replicate the dynamic testing conditions used to obtain the numerical and experimental results. Dalli [18] crushed simultaneously two SISs with a sled-impactor weighing 780 kg and with an impact velocity around 10 m/s. To replicate the testing conditions in the `AnalyticalTool.m` program, the relevant parameters in the **Simulation Control Inputs** input section were defined according to the list below.

- `DynamicAnalysis`=true;
  - `V0`=10;
  - `ImpactorMass`=390; (390=780/2, which is the weight to crush one a single SIS)
  - `DropTowerFlag`=false;
- `optimization`=false;

Figure 5.17 shows the obtained force-displacement graph for the dynamic carbon fibre SIS simulation and the numerical FEM macro-scale results obtained by Dalli. Similarly, Figure 5.18 shows the obtained force-displacement graph and its confidence bounds compared against the two dynamic experimental tests performed by Dalli [18]. Finally, Figure 5.19 compares the calculated analytical and numerical velocities with the experimentally obtained velocity.

Similar to what was observed in the quasi-static simulation, it can be seen in Figure 5.17 that the analytical tool can accurately match the results obtained from a numerical macro-scale FEM simulation but in a fraction of the total time required to execute the FEM simulation.

Figure 5.18 shows that most of the experimental data falls within the analytical confidence bounds, and the prediction accurately describes the overall structure's behaviour during crushing. The prediction improved relative to the quasi-static relation, which was also true of the numerical prediction. The results indicate that the velocity factor (F_V) accurately represents the dynamic

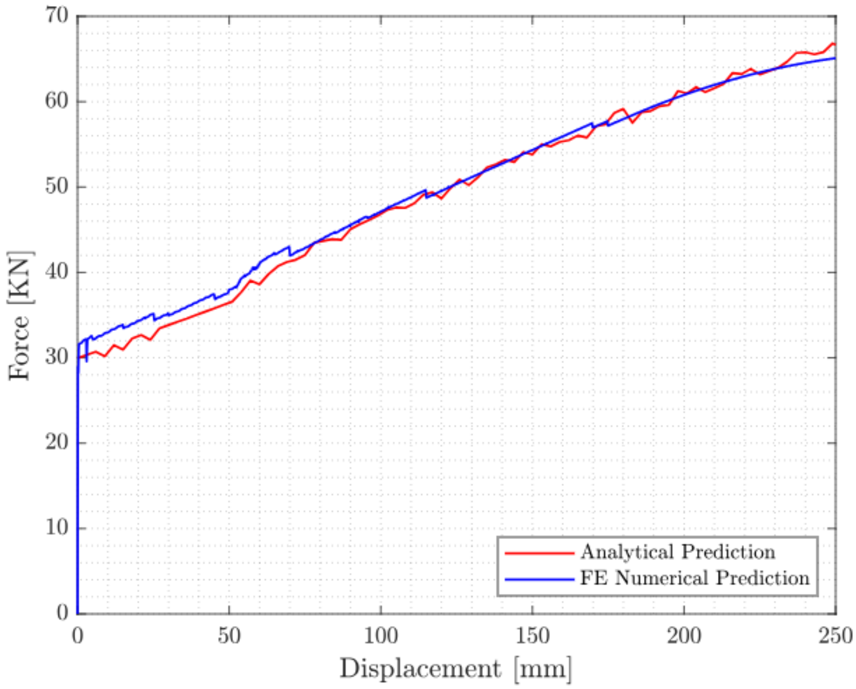


Figure 5.17: Force-displacement curves comparing the dynamic predictions from the analytical and numerical models

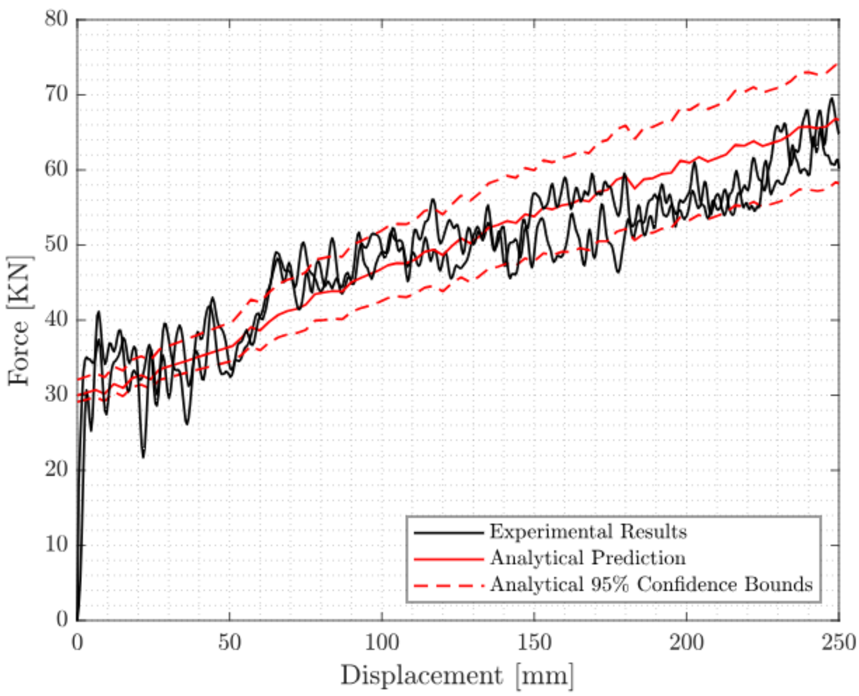


Figure 5.18: Force-displacement curves comparing the dynamic predictions from the analytical model and the experimental results obtained by Dalli [18]

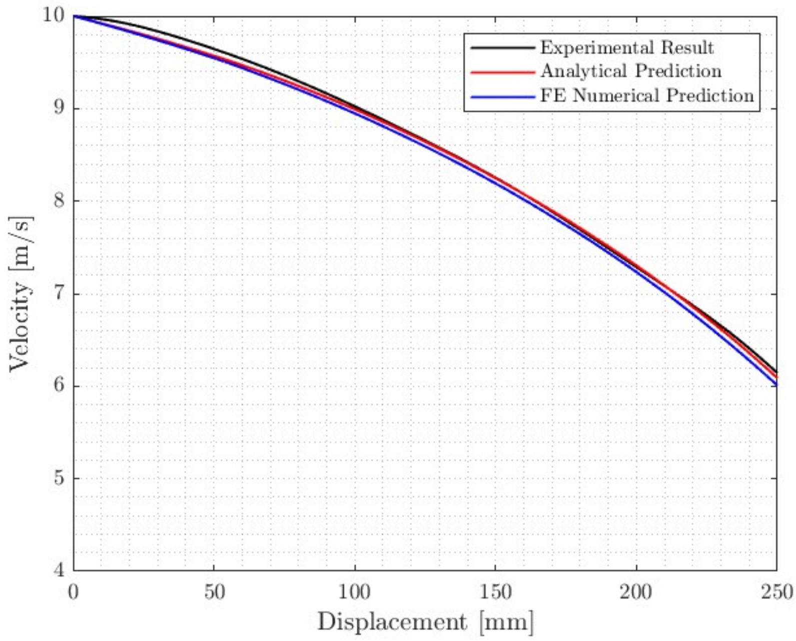


Figure 5.19: Velocity-displacement curves comparing the calculated analytical and numerical velocities with the experimentally obtained velocity by Dalli [18]

effects on the structure. However, the significant scatter of the carbon material data used to create the velocity curve might be forming an exaggerated confidence band, which might disappear if future experimental dynamic data is not so scattered.

Lastly, Figure 5.19 illustrates that the analytical crushing head velocity prediction agrees with the numerically and experimentally obtained results. Effectively, Figure 5.19 shows that the mathematical model described in Section 3.5.4.4 used to perform the velocity update adequately translates the system's dynamic behaviour.

5.3.2 Hybrid Quasi-static Simulations

This section shows the quasi-static simulations made for an SIS structure whose outer shell was made from the hybrid laminate and for which crush stress parameter curves were obtained in Section 3.4.5.2. The SIS's rib was made from carbon. The predictions are compared with the experimental results of a hybrid SIS obtained by Dalli.

At the current development stage, the program cannot run two distinct materials simultaneously. Therefore, a quasi-static simulation was run just for the outer shell with eight hybrid plies, and the resulting *OutputTable.xlsx* file was saved. Another quasi-static simulation was run just with the rib with four carbon plies, and the output table Excel file was also saved. Finally, the total force to crush the entire structure was calculated by adding the contributions of the rib and the outer shell.

For a dynamic simulation, the process described in the last paragraph would not work since the velocity update would be different for the outer shell and the rib. In that case, the tool's code would have to be altered, and the functionality to run two materials simultaneously would have to be implemented. Due to the code's structure and modularity, such a feature can be implemented relatively quickly and straightforwardly. The reason for it not being already implemented is that the hybrid experimental test was conducted near the end of the development cycle. Until then, there had been no need to implement such a feature. Due to time constraints, it was decided not to implement the feature and join both simulations manually.

Two simulations were run, one for the actual hybrid crush stress parameter curves and one for the hybrid curves with the duplicated R5 values (see Section 3.4.5.2). The latter's purpose was to force a plateau on the RoC curve, anticipating the experimental results of tubular specimens with larger radii. Figure 5.20 shows the force-displacement curve obtained for the actual hybrid curves with the experimental result overlayed. Figure 5.21 shows the same but for the hybrid curve with the duplicated R5 values.

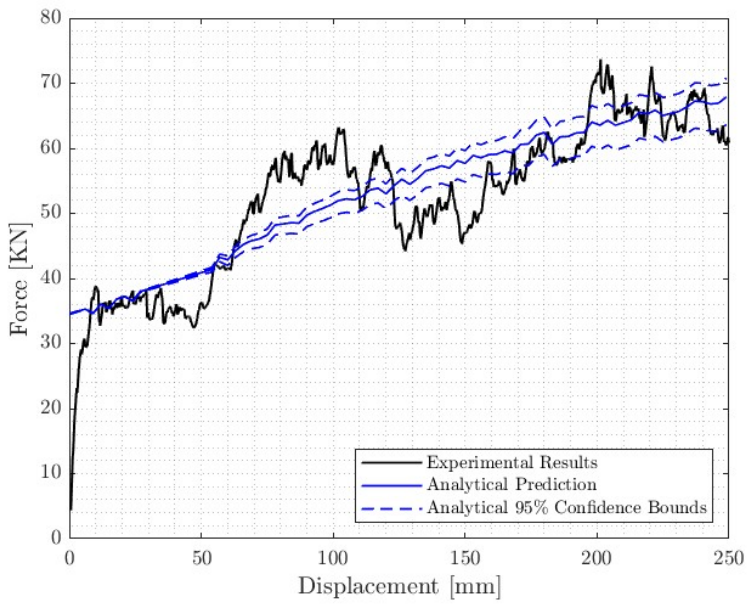


Figure 5.20: Force-displacement curve comparing the quasi-static predictions from the hybrid analytical model and the experimental result

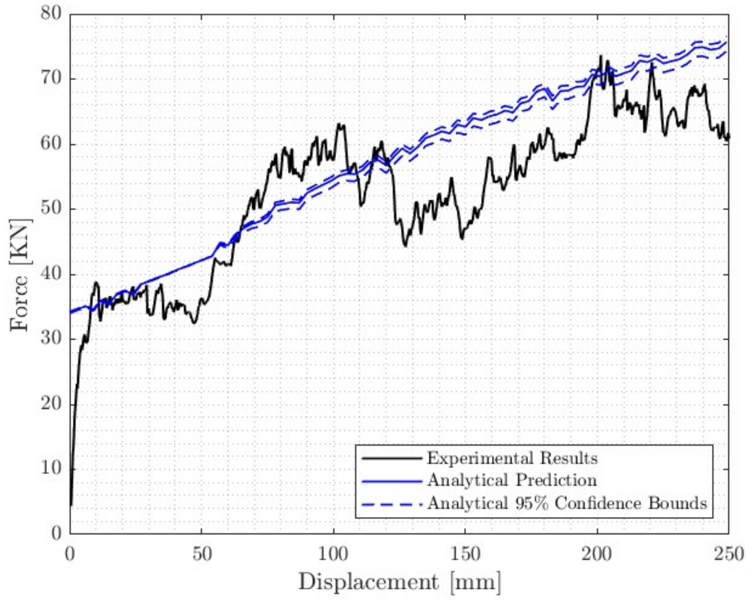


Figure 5.21: Force-displacement curve comparing the quasi-static predictions from the hybrid analytical model with duplicated R5 values and the experimental result

Analysing Figures 5.20 and 5.21, it can be observed that both predictions describe the structure's behaviour reasonably well. However, the prediction for the duplicated R5 values was overall less accurate than for the unduplicated result, indicating that the plateau has not yet been reached and the choice to copy the values was too strong. On the other hand, it is important to stress that a single specimen was tested for the hybrid structure, and it is possible that the specimen tested is not representative of the overall behaviour. Further testing of larger radii tubular coupons and hybrid structures is, therefore, recommended.

Chapter 6

Conclusions and Future Work

This chapter presents the main conclusions regarding the development of an analytical tool to assess the energy absorption characteristics of composite crash structures. Based on the results obtained and the tool's limitations, further developments to increase the tool's accuracy and add new capabilities are proposed.

6.1 Conclusions

The automotive and motorsports industries constantly strive to build lighter vehicles to achieve higher performances and gain an edge over competitors. Given this, composite crash structures are becoming more common due to their increased specific energy absorption capacities. As a result, several research projects were undertaken to evaluate the energy absorption of fibre-reinforced polymers by using coupon testing, numerical computational analysis and techniques, and testing of larger structures.

Numerical simulations are critical in the crashworthiness assessment of a structure because of the excessive costs associated with composite manufacture and testing. Several research studies on various numerical methods for this topic were carried out. It was discovered that mesoscale modelling can precisely replicate the interlaminar behaviour of composite crushing. Despite this, their extensive computation runtime and laborious calibration procedures render them unsuitable for use in industry. On the other hand, the macro-scale modelling approach has been shown to produce accurate simulation results more affordably, making them the current industry standard for crashworthiness predictions. Macro-scale models are also used to verify that the designed crash structure does not prematurely fail due to buckling or other elastic failure mechanisms. However, significant experimental testing is still required to calibrate the model, and the total runtime for these simulations is still around six to twelve hours.

Although finite element models are essential for verifying the final design, their long runtimes make them unsuitable for the initial design phase. This phase concerns selecting materials, determining preliminary laminate configurations, and identifying geometric features required to meet the crash structure's target performance criteria. Currently, the initial design stage is a heavily interactive process that typically relies on internal design expertise and a trial-and-error approach to reach a final design solution. It is clear that the current design flow for crash structures lacks a tool that allows engineers to obtain predictions of a structure's energy absorption capabilities quickly, and if the results obtained are unfavourable, change the geometry and test the new design iteration until a solution is achieved.

The review of the current literature on the testing of tubular and conical coupons, both under quasi-static and dynamic conditions, showcases parameters that affect a structure's crush stress, namely the surface's curvature and inclination angle relative to the crushing axis, the strain rate and the laminate thickness. A program called `MaterialCharacterization.m` was created that uses the available coupon level data to fit crush stress parameter curves that describe the change in crush stress with one of the influence parameters described above. The script is also able to bootstrap the inputted material data to create confidence bounds for the crush stress parameter curves used, which are used by the analytical tool to define upper and lower bounds for the prediction, providing a rudimentary level of uncertainty quantification. This work obtained curves for carbon and carbon-flax hybrid laminates using the coupon experimental data obtained by Moreira [8].

The curves obtained upon the script's execution for the carbon and hybrid materials were good overall fits to the available experimental data. The RoC curves were fitted through a *power2* fit. They captured the decrease in crush stress for gradually increasing values of RoC observed in the literature from the testing of different sizes of tubular coupons. The RoC curves plateaued when RoC tended to infinity. The curves describing the variation of crush stress with velocity were fitted with a *exp1* fit. They accurately captured the progressive decrease in crush stress with velocity seen in the literature from the dynamic experimental tests. Since insufficient experimental data was available to fit a curve for the inclination, a temporary theoretical curve based on observations from experiments conducted in cones was used instead. The effect of thickness was not considered in this work because of insufficient experimental data. Insufficient studies regarding the impact of laminate thickness prevented the provisional use of a thickness theoretical curve.

The macro `PartitioningTool.vba` was created for Autodesk Inventor to automatically partition the crash structure's CAD geometry before saving it as an STL file and inputting it for analysis. The macro is able to partition the surfaces along the x, y and z axes. Partitions along the z axis are mandatory for the tool to function since they create complete cross-sections of nodes from which the crushing force is calculated. Partitions along the x and y axes are optional and increase the tool's accuracy. A study was conducted to assess the effect of x and y partitions on a Formula 1 Side Impact Structure (SIS) model, and it was concluded that the analytical tool's ability to calculate the geometric discretisations accurately was significantly worse if no partitions were made.

The final program created is the `AnalyticalTool.m` program, which takes as input the normalised crush stress parameter curves obtained from the `MaterialCharacterization.m` and the STL files containing the partitioned CAD model for the crash structure to be analysed. The tool outputs a VTK file for 3D visualisation of results using Paraview and graphs documenting the force-displacement and stress-displacement curves. An output log containing a simulation summary is printed into MATLAB's command window, and an Excel table with the results' numerical data is created.

The tool calculates the radius of curvature and inclination at all nodes in the inputted structure. It uses that information to calculate and assign values of crush stresses based on the crush stress parameter curves. If the simulation is dynamic, the crushing head's deceleration is also automatically calculated. Finally, the tool offers the option to optimise the number of plies required at each cross-section to withstand a specified target force. In this work, the correct function of all the mentioned processes was verified, and it was concluded that all are correctly working.

The results of the analytical tool for the SIS were compared with the experimental and numerical results obtained by Dalli [18] for a carbon fibre SIS. It was seen that the tool was able to accurately replicate the deterministic numerical results for the quasi-static and dynamic simulations performed. For the experimental results, the tool could accurately reproduce the SIS's behaviour for a dynamic simulation. However, the quasi-static analytical predictions underestimated the experimental results by 13%. For the hybrid SIS, it was observed that both predictions describe the structure's behaviour reasonably well. However, because only one hybrid structure was tested, it is possible that the experimental results might not accurately reflect the structure's real behaviour.

Overall, the thesis's objectives were successfully fulfilled. The tool accurately and quickly calculated the energy absorption capabilities of a crash structure. The tool's runtime for the most complex structures tested was 61.8 s, vastly faster than the numerical models currently used in the industry, allowing engineers to quickly test and improve early-stage designs.

6.2 Future Work

The lack of experimental data to fit crush stress parameter curves is the current bottleneck limiting the tool's accuracy. More specifically, crush testing of conical specimens should be conducted to assess the effect of inclination on crush stress, and crush testing on tubular specimens with increasing laminate thicknesses should be undertaken to determine the impact of thickness on crush stress. Additionally, testing of tubular specimens with larger radii should be done to confirm the hypothesis that the crush stress plateaus when RoC tens to infinity. Developing a universal testing standard for progressive crushing of specimens is also paramount. There is no currently established method for obtaining experimental crushing data, which delays scientific discovery as different researchers have no reliable way of comparing data.

The inability to run parts of distinct materials is, currently, the most severe tool limitation. However, due to the code's high modularity, its implementation would be relatively straightforward, requiring only minor code changes in the tool's inputs and some changes to the `MaterialCurves.m` and `CrushStress.m` functions to accommodate multiple materials.

Another interesting addition would be to make the tool compatible with FEM meshed parts to allow currently existing industry models to be run in the tool. However, this would require a complete overhaul of how data is treated and stored in the tool, which would mean entirely replacing the `Treatment_v2.m` function or creating a parallel script to adapt the data into a usable format.

Finally, the control inputs are assigned directly to the variables on the program's code. Developing a Graphical User Interface to streamline the user experience and provide a more aesthetically pleasing environment would be interesting, especially if the tool is to be used in the industry.

References

[1] Mateus DG. Development of an analytical tool for rapid initial laminate design of composite crash structures. Faculdade de Engenharia da Universidade do Porto 2024;.

[2] Moropoulou A, Bakolas A, Anagnostopoulou S. Composite materials in ancient structures. Cement and concrete composites 2005;27(2):295–300.

[3] ACS. Leo Hendrick Baekeland and the Invention of Bakelite. 2022. https://www.acs.org/education/whatischemistry/landmarks/bakelite.html#top [Accessed: 15-06-2024].

[4] Dwight DW, Begum S. 1.10 glass fiber reinforcements. Compreensive composite materials II 2018;.

[5] BAE Systems. De Havilland Mosquito. 2022. https://https://www.baesystems.com/en/heritage/de-havilland-mosquito [Accessed: 15-06-2024].

[6] EPOXY RESIN COMMITTEE. EPOXIES AT A GLANCE. 2015. https://epoxy-europe.eu/wp-content/uploads/2015/06/epoxy_resin_committee_factsheet_epoxies_at_a_glance-2.pdf [Accessed: 15-06-2024].

[7] Mar-bal. History Of composites. 2018. https://www.mar-bal.com/language/en/applications/history-of-composites/ [Accessed: 15-06-2024].

[8] Moreira DFM. Development of hybrid composite coupons and experimental set-ups for dynamic crash testing. Faculdade de Engenharia da Universidade do Porto 2024;.

[9] Mangalgiri P. Composite materials for aerospace applications. Bulletin of Materials Science 1999;22:657–64.

[10] Mittal G, Rhee KY, Mišković-Stanković V, Hui D. Reinforcements in multi-scale polymer composites: Processing, properties, and applications. Composites Part B: Engineering 2018;138:122–39. URL: https://www.sciencedirect.com/science/article/pii/S1359836817335813. doi:https://doi.org/10.1016/j.compositesb.2017.11.028.

[11] Fang J, Sun G, Qiu N, Kim NH, Li Q. On design optimization for structural crashworthiness and its state of the art. Structural and Multidisciplinary Optimization 2017;55:1091–119.

[12] Kim SH, Kim H, Kim NJ. Brittle intermetallic compound makes ultrastrong low-density steel with large ductility. Nature 2015;518(7537):77–9.

[13] Zhang Y, Zhu P, Chen G, Lin Z. Study on structural lightweight design of automotive front side rail based on response surface method 2007;.

[14] Zhu G, Sun G, Yu H, Li S, Li Q. Energy absorption of metal, composite and metal/composite hybrid structures under oblique crushing loading. International Journal of Mechanical Sciences 2018;135:458–83.

References

[15] Zhou Y, Lan F, Chen J. Crashworthiness research on s-shaped front rails made of steel–aluminum hybrid materials. Thin-Walled Structures 2011;49(2):291–7.

[16] Savage G. Development of penetration resistance in the survival cell of a formula 1 racing car. Engineering Failure Analysis 2010;17(1):116–27.

[17] Scarborough C. Safety first– how the Fia foundation is ever striving to make motorsport a safer pursuit. 2010. `https://motorsport.tech/formula-1/how-the-fia-foundation-is-ever-striving-to-make-motorsport-safer` [Accessed: 1-05-2024].

[18] Dalli D. Experimental and numerical developments in predicting the crashworthiness of formula one composite structures. Phd thesis; 2020.

[19] Yan L, Chouw N. Crashworthiness characteristics of flax fibre reinforced epoxy tubes for energy absorption application. Materials & Design 2013;51:629–40.

[20] Mamalis A, Robinson M, Manolakos D, Demosthenous G, Ioannidis M, Carruthers J. Crashworthy capability of composite material structures. Composite structures 1997;37(2):109–34.

[21] Mamalis A, Manolakos D, Demosthenous G, Ioannidis M. The static and dynamic axial crumbling of thin-walled fibreglass composite square tubes. Composites Part B: Engineering 1997;28(4):439–51.

[22] Xu J, Ma Y, Zhang Q, Sugahara T, Yang Y, Hamada H. Crashworthiness of carbon fiber hybrid composite tubes molded by filament winding. composite structures 2016;139:130–40.

[23] Dalli D, Varandas LF, Catalanotti G, Foster S, Falzon BG. Assessing the current modelling approach for predicting the crashworthiness of formula one composite structures. Composites Part B: Engineering 2020;201:108242.

[24] Hsissou R, Seghiri R, Benzekri Z, Hilali M, Rafik M, Elharfi A. Polymer composite materials: A comprehensive review. Composite structures 2021;262:113640.

[25] Jweeg MJ, Hammood AS, Al-Waily M. Experimental and theoretical studies of mechanical properties for reinforcement fiber types of composite materials. International Journal of Mechanical & Mechatronics Engineering IJMME-IJENS 2012;12(04):62.

[26] Yi XS, Du S, Zhang L. Composite materials engineering, volume 1: fundamentals of composite materials. Springer; 2018.

[27] Kindervater C, Georgi H. Composite strength and energy absorption as an aspect of structural crash resistance. In: Structural crashworthiness and failure. CRC Press; 1993, p. 189–235.

[28] Isobe T, Tanaka T, Nomura T, Yuasa R. Comparison of strength of 3d printing objects using short fiber and continuous long fiber. In: IOP Conference Series: Materials Science and Engineering; vol. 406. IOP Publishing; 2018, p. 12–042.

[29] Savage G. Enhancing the exploitation and efficiency of fibre-reinforced composite structures by improvement of interlaminar fracture toughness. Engineering Failure Analysis 2006;13(2):198–209.

[30] Hamada H, Coppola J, Hull D, Maekawa Z, Sato H. Comparison of energy absorption of carbon/epoxy and carbon/peek composite tubes. Composites 1992;23(4):245–52.

[31] Jacob GC, Fellers JF, Simunovic S, Starbuck JM. Energy absorption in polymer composites for automotive crashworthiness. Journal of composite materials 2002;36(7):813–50.

[32] Kashani MH, Milani AS. Damage prediction in woven and non-woven fabric composites. In: Non-woven fabrics. IntechOpen; 2016, p. 324.

[33] Dixit A, Mali HS. Modeling techniques for predicting the mechanical properties of woven-fabric textile composites: a review. Mechanics of composite Materials 2013;49:1–20.

[34] Kim JK, Sham ML. Impact and delamination failure of woven-fabric composites. Composites Science and Technology 2000;60(5):745–61.

[35] Savage G. Formula 1 composites engineering. Engineering failure analysis 2010;17(1):92–115.

[36] FIA. FIA ENVIRONMENTAL POLICY. 2024. `https://www.fia.com/fia-environmental-policy` [Accessed: 25-06-2024].

[37] Isaac CW, Ezekwem C. A review of the crashworthiness performance of energy absorbing composite structure within the context of materials, manufacturing and maintenance for sustainability. Composite Structures 2021;257:113081.

[38] Rodríguez E, Petrucci R, Puglia D, Kenny JM, Vazquez A. Characterization of composites based on natural and glass fibers obtained by vacuum infusion. Journal of composite materials 2005;39(3):265–82.

[39] López-Alba E, Schmeer S, Díaz F. Energy absorption capacity in natural fiber reinforcement composites structures. Materials 2018;11(3):418.

[40] Dittenber DB, GangaRao HV. Critical review of recent publications on use of natural composites in infrastructure. Composites Part A: applied science and manufacturing 2012;43(8):1419–29.

[41] Stamboulis A, Baillie C, Garkhail S, Van Melick H, Peijs T. Environmental durability of flax fibres and their composites based on polypropylene matrix. Applied composite materials 2000;7:273–94.

[42] Mohanty A, Misra M, Drzal LT. Surface modifications of natural fibers and performance of the resulting biocomposites: An overview. Composite interfaces 2001;8(5):313–43.

[43] Arbelaiz A, Fernandez B, Ramos J, Retegi A, Llano-Ponte R, Mondragon I. Mechanical properties of short flax fibre bundle/polypropylene composites: Influence of matrix/fibre modification, fibre content, water uptake and recycling. Composites science and technology 2005;65(10):1582–92.

[44] Lobão VM. Development and testing of hybrid composite coupons for crush applications. Faculdade de Engenharia da Universidade do Porto 2022;.

[45] Strohrmann K, Schmeer S, Fortin G, Hamada H, Hajek M. Crashworthiness characteristics of carbon-flax composite tubes for aerospace applications. In: ECCM18-18th European Conference on Composite Materials. 2018,.

[46] Mamalis A, Manolakos D, Ioannidis M, Papapostolou D. On the response of thin-walled cfrp composite tubular components subjected to static and dynamic axial compressive loading: experimental. Composite structures 2005;69(4):407–20.

[47] Savona SC, Hogg P. Effect of fracture toughness properties on the crushing of flat composite plates. Composites science and technology 2006;66(13):2317–28.

[48] Hussein RD, Ruan D, Lu G. An analytical model of square cfrp tubes subjected to axial compression. Composites Science and Technology 2018;168:170–8.

[49] Boria S, Pettinari S, Giannoni F. Theoretical analysis on the collapse mechanisms of thin-walled composite tubes. Composite Structures 2013;103:43–9.

[50] Mamalis A, Manolakos D, Demosthenous G, Ioannidis M. Analytical modelling of the static and dynamic axial collapse of thin-walled fibreglass composite conical shells. International journal of impact engineering 1997;19(5-6):477–92.

[51] Pinho S, Gutkin R, Pimenta S, De Carvalho N, Robinson P. Fibre-dominated compressive failure in polymer matrix composites. In: Failure mechanisms in polymer matrix composites. Elsevier; 2012, p. 183–223.

[52] Ogin S, Brøndsted P, Zangenberg J. Composite materials: Constituents, architecture, and generic damage. In: Modeling damage, fatigue and failure of composite materials. Elsevier; 2016, p. 3–23.

[53] Zhou L, Zhao L, Liu F, Zhang J. A micromechanical model for longitudinal compressive failure in unidirectional fiber reinforced composite. Results in Physics 2018;10. doi:10.1016/j.rinp.2018.08.003.

[54] Maimi P, Rodríguez H, Blanco N, Mayugo J. Numerical modeling of matrix cracking and intralaminar failure in advanced composite materials. In: Numerical Modelling of Failure in Advanced Composite Materials. Elsevier; 2015, p. 175–92.

[55] Zubillaga L, Turon A, Renart J, Costa J, Linde P. An experimental study on matrix crack induced delamination in composite laminates. Composite Structures 2015;127:10–7.

[56] Jalalvand M, Hosseini-Toudeshky H, Mohammadi B. Numerical modeling of diffuse transverse cracks and induced delamination using cohesive elements. Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science 2013;227:1392–405. doi:10.1177/0954406212460974.

[57] Safri SNAB, Sultan M, Jawaid M. Damage analysis of glass fiber reinforced composites. In: Durability and life prediction in biocomposites, fibre-reinforced composites and hybrid composites. Elsevier; 2019, p. 133–47.

[58] Suppakul P, Bandyopadhyay S. The effect of weave pattern on the mode-i interlaminar fracture energy of e-glass/vinyl ester composites. Composites Science and Technology 2002;62(5):709–17.

[59] Savage G, Bomphray I, Oxley M. Exploiting the fracture properties of carbon fibre composites to design lightweight energy absorbing structures. Engineering Failure Analysis 2004;11(5):677–94.

[60] Hull D. A unified approach to progressive crushing of fibre-reinforced composite tubes. Composites science and technology 1991;40(4):377–421.

[61] Berry P. Energy absorption and failure mechanisms of axially crushed grp tubes. Ph.D. thesis; University of Liverpool; 1983.

[62] Feraboli P. Current efforts in standardization of composite materials testing for crashworthiness and energy absorption. In: 47th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference 14th AIAA/ASME/AHS Adaptive Structures Conference 7th. 2006, p. 2217.

[63] Andrulonis R, Kiser JD, David KE, Davies C, Ashforth C. Cmh-17 volume 5 ceramic matrix composites. Tech. Rep.; 2017.

[64] Jackson A, Dutton S, Gunnion A, Kelly D. Investigation into laminate design of open carbon–fibre/epoxy sections by quasi–static and dynamic crushing. Composite Structures 2011;93(10):2646–54.

[65] Bisagni C, Di Pietro G, Fraschini L, Terletti D. Progressive crushing of fiber-reinforced composite structural components of a formula one racing car. Composite structures 2005;68(4):491–503.

[66] FIA. 2024 formula 1 technical regulations. 2024. `https://www.fia.com/regulation/category/110` [Accessed: 10-05-2024].

[67] Matsumura T, Park C, Doyon D, Haftka R, Kim NH. Modeling the contribution of accident investigation to aircraft safety. In: 10th AIAA Aviation Technology, Integration, and Operations (ATIO) Conference. 2010, p. 9057.

[68] Ueda M, Anzai S, Kubo T. Progressive crushing of a unidirectional cfrp plate with v-shaped trigger. Advanced Composite Materials 2015;24(1):85–95.

[69] Bru T, Waldenström P, Gutkin R, Olsson R, Vyas GM. Development of a test method for evaluating the crushing behaviour of unidirectional laminates. Journal of composite materials 2017;51(29):4041–51.

[70] Feraboli P. Development of a modified flat-plate test specimen and fixture for composite materials crush energy absorption. Journal of composite materials 2009;43(19):1967–90.

[71] Feraboli P. Development of a corrugated test specimen for composite materials energy absorption. Journal of Composite Materials 2008;42(3):229–56.

[72] Waimer M, Kohlgrüber D, Hachenberg D, Voggenreiter H. Experimental study of cfrp components subjected to dynamic crash loads. Composite Structures 2013;105:288–99.

[73] Vigna L, Babaei I, Garg R, Belingardi G, Paolino DS, Calzolari A, et al. An innovative fixture for testing the crashworthiness of composite materials. Frattura ed Integrità Strutturale 2021;15(55):76–87.

[74] Israr HA, Rivallant S, Barrau JJ. Experimental investigation on mean crushing stress characterization of carbon–epoxy plies under compressive crushing mode. Composite Structures 2013;96:357–64.

[75] Yan L, Chouw N, Jayaraman K. Effect of triggering and polyurethane foam-filler on axial crushing of natural flax/epoxy composite tubes. Materials & Design (1980-2015) 2014;56:528–41.

[76] Mamalis AG, Manolakos D, Ioannidis M, Papapostolou D. Crashworthy characteristics of axially statically compressed thin-walled square cfrp composite tubes: experimental. Composite Structures 2004;63(3-4):347–60.

[77] Mahdi E, Ochoa D, Vaziri A, Eltai E. Energy absorption capability of date palm leaf fiber reinforced epoxy composites rectangular tubes. Composite Structures 2019;224:111004.

[78] Liu Q, Mo Z, Wu Y, Ma J, Tsui GCP, Hui D. Crush response of cfrp square tube filled with aluminum honeycomb. Composites Part B: Engineering 2016;98:406–14.

References

[79] Boria S, Scattina A, Belingardi G. Axial energy absorption of cfrp truncated cones. Composite Structures 2015;130:18–28.

[80] Ochelski S, Gotowicki P. Experimental assessment of energy absorption capability of carbon-epoxy and glass-epoxy composites. Composite structures 2009;87(3):215–24.

[81] Alkateb M, Sapuan S, Leman Z, Ishak M, Jawaid M. Vertex angles effects in the energy absorption of axially crushed kenaf fibre-epoxy reinforced elliptical composite cones. Defence Technology 2018;14(4):327–35.

[82] Meredith J, Cozien-Cazuc S, Collings E, Carter S, Alsop S, Lever J, et al. Recycled carbon fibre for high performance energy absorption. Composites Science and Technology 2012;72(6):688–95.

[83] Farley GL. The effects of crushing speed on the energy-absorption capability of composite tubes. Journal of Composite Materials 1991;25(10):1314–29.

[84] Liu X, Belkassem B, Jonet A, Lecompte D, Van Hemelrijck D, Pintelon R, et al. Experimental investigation of energy absorption behaviour of circular carbon/epoxy composite tubes under quasi-static and dynamic crush loading. Composite Structures 2019;227:111266.

[85] Mamalis A, Manolakos D, Demosthenous G, Ioannidis M. The static and dynamic axial collapse of fibreglass composite automotive frame rails. Composite Structures 1996;34(1):77–90.

[86] Lavoie JA, Kellas S. Dynamic crush tests of energy-absorbing laminated composite plates. Composites Part A: Applied Science and Manufacturing 1996;27(6):467–75.

[87] Ramakrishna S. Energy absorption characteristics of knitted fabric reinforced epoxy composite tubes. Journal of reinforced plastics and composites 1995;14(10):1121–41.

[88] Kindervater C, Thomson R, Johnson A, David M, Joosten M, Mikulik Z, et al. Validation of crashworthiness simulation and design methods by testing of a scaled composite helicopter frame section 2011;.

[89] Liu Q, Ou Z, Mo Z, Li Q, Qu D. Experimental investigation into dynamic axial impact responses of double hat shaped cfrp tubes. Composites Part B: Engineering 2015;79:494–504.

[90] Hu D, Luo M, Yang J. Experimental study on crushing characteristics of brittle fibre/epoxy hybrid composite tubes. International journal of crashworthiness 2010;15(4):401–12.

[91] Mamalis A, Manolakos D, Demosthenous G, Ioannidis M. Energy absorption capability of fibreglass composite square frusta subjected to static and dynamic axial collapse. Thin-Walled Structures 1996;25(4):269–95.

[92] Mamalis A, Manolakos D, Demosthenous G, Ioannidis M. Axial collapse of thin-walled fibreglass composite tubular components at elevated strain rates. Composites Engineering 1994;4(6):653–77.

[93] Chiu LN, Falzon BG, Ruan D, Xu S, Thomson RS, Chen B, et al. Crush responses of composite cylinder under quasi-static and dynamic loading. Composite Structures 2015;131:90–8.

[94] Mamalis AG, Yuan YB, Viegelahn G. Collapse of thin–wall composite sections subjected to high speed axial loading. International journal of vehicle design 1992;13(5-6):564–79.

[95] Chiu LN, Falzon BG, Boman R, Chen B, Yan W. Finite element modelling of composite structures under crushing load. Composite Structures 2015;131:215–28.

[96] Chiu LN, Falzon BG, Chen B, Yan W. Validation of a 3d damage model for predicting the response of composite structures under crushing loads. Composite Structures 2016;147:65–73.

[97] Tan W, Falzon BG, Price M, Liu H. The role of material characterisation in the crush modelling of thermoplastic composite structures. Composite Structures 2016;153:914–27.

[98] Feraboli P, Deleo F, Wade B, Rassaian M, Higgins M, Byar A, et al. Predictive modeling of an energy-absorbing sandwich structural concept using the building block approach. Composites Part A: Applied Science and Manufacturing 2010;41(6):774–86.

[99] Feraboli P, Wade B, Deleo F, Rassaian M, Higgins M, Byar A. Ls-dyna mat54 modeling of the axial crushing of a composite tape sinusoidal specimen. Composites Part A: applied science and manufacturing 2011;42(11):1809–25.

[100] Boria S, Belingardi G, et al. Composite impact attenuator with shell and solid modelling. In: 11th World congress on computational mechanics (WCC M XI). 2014, p. 2237–45.

[101] ABAQUS. 22.1.1 Solid (continuum) elements. 2006. `https://classes.engineering.wustl.edu/2009/spring/mase5513/abaqus/docs/v6.6/books/usb/default.htm?startat=pt06ch22s01alm01.html` [Accessed: 26-05-2024].

[102] ABAQUS. 3.2.1 Solid element overview. 2006. `https://classes.engineering.wustl.edu/2009/spring/mase5513/abaqus/docs/v6.6/books/stm/default.htm?startat=ch03s02ath59.html` [Accessed: 26-05-2024].

[103] ABAQUS. 23.6.2 Choosing a shell element. 2006. `https://classes.engineering.wustl.edu/2009/spring/mase5513/abaqus/docs/v6.6/books/usb/default.htm?startat=pt06ch21s01abo21.html#usb-elm-egeneral` [Accessed: 26-05-2024].

[104] Reiner J, Feser T, Waimer M, Poursartip A, Voggenreiter H, Vaziri R. Axial crush simulation of composites using continuum damage mechanics: Fe software and material model independent considerations. Composites Part B: Engineering 2021;225:109284.

[105] Wade B, Deleo F, Feraboli P, Rassaian M. Crushing of composite structures: experiment and simulation. In: 50th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference 17th AIAA/ASME/AHS Adaptive Structures Conference 11th AIAA No. 2009, p. 2532.

[106] Siromani D, Awerbuch J, Tan TM. Finite element modeling of the crushing behavior of thin-walled cfrp tubes under axial compression. Composites Part B: Engineering 2014;64:50–8.

[107] Sommer DE, Thomson D, Falcó O, Quino G, Cui H, Petrinic N. Damage modelling of carbon fibre composite crush tubes: Numerical simulation and experimental validation of drop weight impact. Composites Part A: Applied Science and Manufacturing 2022;160:107033.

[108] ASTM International. ASTM D3039/D3039M-08 Standard Test Method for Tensile Properties of Polymer Matrix Composite Materials. 2014. `https://www.astm.org/d3039_d3039m-08.html` [Accessed: 26-06-2024].

[109] Schwartz MM. Composite materials handbook. (No Title) 1984;.

[110] Catalanotti G. Uncertainty quantification for advanced progressive damage models for composites by means of efficient emulators and bootstrapping. Composites Part A: Applied Science and Manufacturing 2022;163:107238. URL: `https://www.`

sciencedirect.com/science/article/pii/S1359835X22004195. doi:https://doi.org/10.1016/j.compositesa.2022.107238.

[111] Sudret B, Marelli S, Wiart J. Surrogate models for uncertainty quantification: An overview. In: 2017 11th European conference on antennas and propagation (EUCAP). IEEE; 2017, p. 793–7.

[112] Metalios Z. What is autodesk inventor? 2020. https://solidprofessor.com/blog/what-is-autodesk-inventor/ [Accessed: 27-05-2024].

[113] Autodesk Inventor. VBA in Autodesk Inventor. 2022. https://help.autodesk.com/view/INVNTOR/2022/ENU/?guid=GUID-BDDB44A3-3415-44EB-B10E-216BB08BD614 [Accessed: 27-05-2024].

[114] Library of Congress. (STereoLithography) File Format Family. 2022. https://www.loc.gov/preservation/digital/formats/fdd/fdd000504.shtml [Accessed: 27-05-2024].

Appendix A

Experimental Material Data and Results From the Material Characterization Script

A.1 Carbon Fibre Laminate

A.1.1 Experimental Data

The data from the quasi-statically tested specimens was used to obtain Table A.1. The mid-surface radius was calculated from each specimen's internal diameter and thickness. Then, a table was made by matching the average crush stress of each specimen with their mid-surface radius. The file containing this table can be directly inputted into the **MaterialCharacterization.m** script.

Table A.1: CFRP RoC experimental data (Obtained from [8])

Mid-surface Radius [mm]	Average Crush Stress (MPa)
11,29	173,96
17,36	141,84
23,89	129,69
29,87	126,50
39,14	120,87
11,31	168,54
17,38	140,93
23,91	126,98
29,95	119,84
39,205	111,05
10,91	162,13
17,36	143,26
23,89	124,46
29,915	126,74
39,155	119,46
10,84	168,85
17,34	146,79

Table A.1 continued from previous page

Mid-surface Radius [mm]	Average Crush Stress (Mpa)
23,91	127,81
29,92	117,14
39,195	109,27
10,89	172,17
17,39	146,51
23,9	121,86
29,94	131,12
39,21	115,99

The obtain Table A.2, the data for the dynamically tested R5 specimen was used. Two distinct dynamic tests were conducted by Moreira [8]: one at a low impact velocity resulting in an average crushing velocity of around 4.3 m/s and another at high velocity resulting in an average crushing velocity of around 6.7 m/s. Each tested R5 specimen is presented in Table A.2 with their average velocity and corresponding average crush stress. The data for the 0 m/s R5 specimens came from the quasi-static experimental tests.

The tube size R5 was chosen because its higher diameter might isolate the effect of crushing velocity in the crush stress. There was a concern that in smaller radii of tube specimens, unforeseen effects might occur between the fronds due to the lack of space to expand or another unknown factor causing an issue, and the resulting experimental data might represent another influence in addition to the effect of crushing velocity.

Table A.2: CFRP Velocity experimental data (Obtained from [8])

Average Velocity [m/s]	Average Crush Stress [MPa]
0	120,87
0	111,05
0	119,46
0	109,27
0	115,99
4,52	105,43
4,52	101,54
4,21	105,17
4,04	110,18
3,84	109,18
6,80	100,13
6,74	96,65
6,69	100,68
6,48	108,99
6,55	101,93

A.1.2 Bootstrapping Coefficients

Table A.3 shows the curve fitting coefficients for a *power2* fit that were obtained when the experimental data from Table A.1 was bootstrapped 100 times by the script **MaterialCharacterization.m**. The values shown are ordered by the fitting coefficient c.

Table A.3: RoC curve fitting coefficients ordered by parameter c resulting from 100 bootstrapped curves

	Fitting Parameters		
ID	a	b	c
1	417,35	-0,5433	55,52
2	369,11	-0,5057	55,56
3	421,74	-0,5743	61,58
4	440,10	-0,6139	66,95
5	479,62	-0,6660	71,08
6	437,58	-0,6278	71,75
7	462,50	-0,6607	73,53
8	468,62	-0,6736	74,25
9	621,89	-0,7761	74,61
10	525,81	-0,7158	75,80
11	472,48	-0,6858	76,05
12	478,84	-0,6959	76,23
13	572,56	-0,7562	76,98
14	477,15	-0,7135	78,06
15	510,95	-0,7171	78,72
16	484,04	-0,7269	80,55
17	589,22	-0,7898	80,79
18	594,39	-0,7961	81,30
19	604,42	-0,8096	82,32
20	599,98	-0,8001	82,63
21	647,34	-0,8389	82,89
22	544,51	-0,7754	83,51
23	676,190	-0,8567	83,52
24	660,77	-0,8479	84,00
25	709,45	-0,8695	84,18
26	571,30	-0,8018	85,09
27	598,98	-0,8191	85,28
28	796,72	-0,9242	85,41
29	652,84	-0,8553	86,36
30	641,29	-0,8530	86,60
31	701,12	-0,8932	87,18
32	664,25	-0,8721	87,28
33	661,49	-0,8796	88,45
34	602,91	-0,8502	88,94
35	666,30	-0,8742	89,10
36	561,45	-0,8230	89,24
37	739,84	-0,9187	89,29
38	670,28	-0,8825	89,66
39	788,13	-0,9454	89,83
40	650,90	-0,8822	89,85
41	820,50	-0,9695	90,27
42	581,71	-0,8471	90,45
43	777,33	-0,9415	90,64
44	569,32	-0,8465	90,69
45	736,05	-0,9384	90,79

Table A.3 continued from previous page

	Fitting Parameters		
46	700,71	-0,9219	90,85
47	726,80	-0,9243	90,92
48	748,34	-0,9404	91,15
49	745,36	-0,9296	91,20
50	746,87	-0,9392	91,23
51	653,81	-0,8908	91,35
52	575,02	-0,8585	92,84
53	702,10	-0,9404	93,24
54	787,40	-0,9692	93,46
55	688,13	-0,9331	93,46
56	738,02	-0,9441	93,76
57	778,09	-0,9740	94,04
58	737,16	-0,9601	94,08
59	805,68	-0,9838	94,17
60	904,53	-1,0310	94,60
61	721,28	-0,9655	95,31
62	778,14	-0,9879	95,37
63	800,15	-0,9866	95,48
64	857,57	-1,0174	95,84
65	881,16	-1,0320	96,15
66	849,31	-1,0342	96,76
67	888,66	-1,0448	96,82
68	1082,56	-1,1238	97,15
69	999,02	-1,0968	97,60
70	847,40	-1,0436	97,77
71	946,94	-1,0766	98,47
72	904,07	-1,0625	98,98
73	1366,17	-1,2171	99,27
74	982,03	-1,0992	99,41
75	951,71	-1,0906	99,82
76	1100,82	-1,1454	99,89
77	1436,74	-1,2460	100,66
78	1261,09	-1,1992	101,17
79	1148,98	-1,1738	101,59
80	1022,79	-1,1414	101,86
81	1005,65	-1,1354	102,15
82	985,12	-1,1174	102,71
83	1028,65	-1,1503	103,08
84	1266,45	-1,2281	103,21
85	1312,62	-1,2595	103,80
86	1543,28	-1,2959	103,88
87	1707,74	-1,3422	104,09
88	1538,58	-1,3214	104,69
89	1467,08	-1,3056	104,71
90	1193,81	-1,2164	104,72
91	996,24	-1,1611	105,17

Table A.3 continued from previous page

	Fitting Parameters		
92	1229,93	-1,2438	105,21
93	1590,61	-1,3274	105,30
94	1961,46	-1,4165	106,63
95	1738,44	-1,3842	107,10
96	2052,50	-1,4410	107,60
97	1709,66	-1,3799	107,69
98	1715,66	-1,3875	108,13
99	1939,14	-1,4422	108,48
100	1844,90	-1,4273	109,87

A.1.3 Normalized Crush Stress Parameter Curves

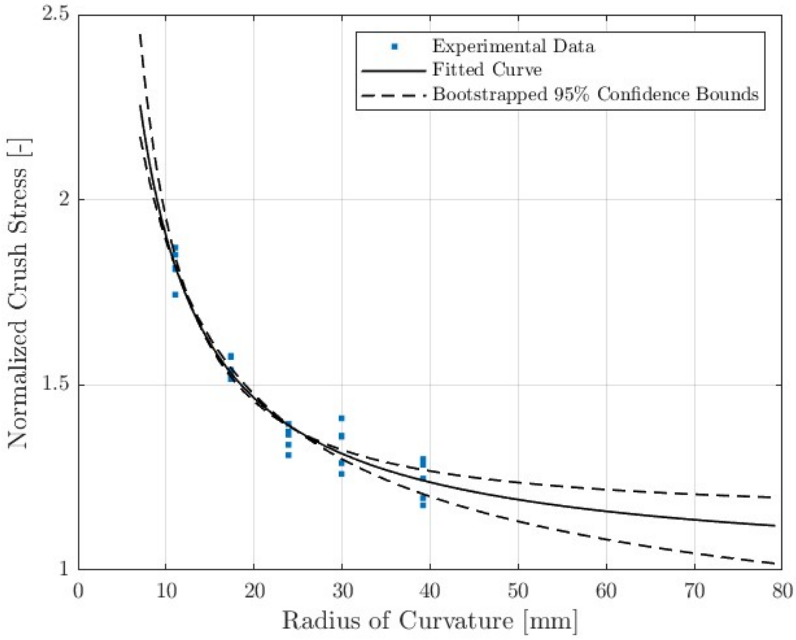


Figure A.1: Normalised variation of crush stress with radius of curvature for a CRFP laminate

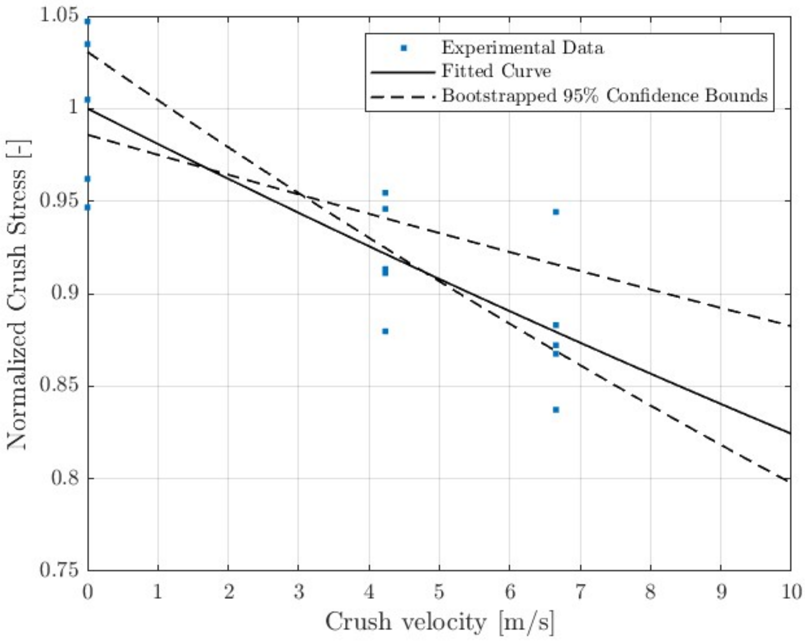


Figure A.2: Normalised variation of crush stress with crushing velocity for a CRFP laminate

Table A.4: Data contained in the DataCurvature file for the CFRP laminate

Flat Transition curvature [mm]	Flat Crush Stress [MPa]	Type of Fit	Fit coefficients	Lower confidence bound coefficients (95%)	Upper confidence bound coefficients (95%)
1221	92,97	power2	7,95	4,78	19,56
			-0,94	-0,588	-1,39
			0,99	0,652	1,15

Table A.5: Data contained in the DataVelocity file for the CFRP laminate

Type of Fit	Fit coefficients	Lower bound coefficients (95%)	Upper bound coefficients (95%)
exp1	1	1,030	0,9858
-	-0,019	-0,02564	-0,0110

A.1.4 Discretized RoC Material Curve

Table A.6: Discretised curvature values for CFRP laminates

RoC Bounds [mm]	Assigned Crush Stress [MPa]
5-6	241.8
6-7	219.8
7-8	203.6
8-9	191.1
9-10	181.2
10-11	173.1
11-12	166.4
12-13	160.8
13-14	156.0
14-15	151.8
15-16	148.2

Table A.6 continued from previous page

RoC Bounds [mm]	Assigned Crush Stress [MPa]
16-18	143.6
18-20	138.5
20-23	133.5
23-26	128.6
26-31	123.9
31-37	119.0
37-47	114.3
47-63	109.4
63-94	104.7
94-180	100.0
180-1221	95.3
1221-1000000000	93.0

A.2 Hybrid Laminate

A.2.1 Experimental Data

Table A.7: Hybrid RoC experimental data (Obtained from [8]).

Mid-surface Radius [mm]	Average Crush Stress [MPa]
11,05	131,73
17,54	112,23
24,05	97,06
30,13	89,02
39,18	87,65
11,06	130,72
17,56	111,23
24,06	95,64
30,07	86,10
39,20	86,40
11,06	128,29
17,55	109,19
24,06	93,64
30,11	87,87
39,35	86,31

Table A.8: Hybrid Velocity experimental data (Obtained from [8]).

Average Velocity [m/s]	Average Crush Stress [MPa]
0	87,65
0	86,40
0	86,31
4,15	71,01
3,96	72,59
4,39	71,35
6,36	69,38

6,26	69,65
6,48	67,69

A.2.2 Normalized Crush Stress Parameter Curves

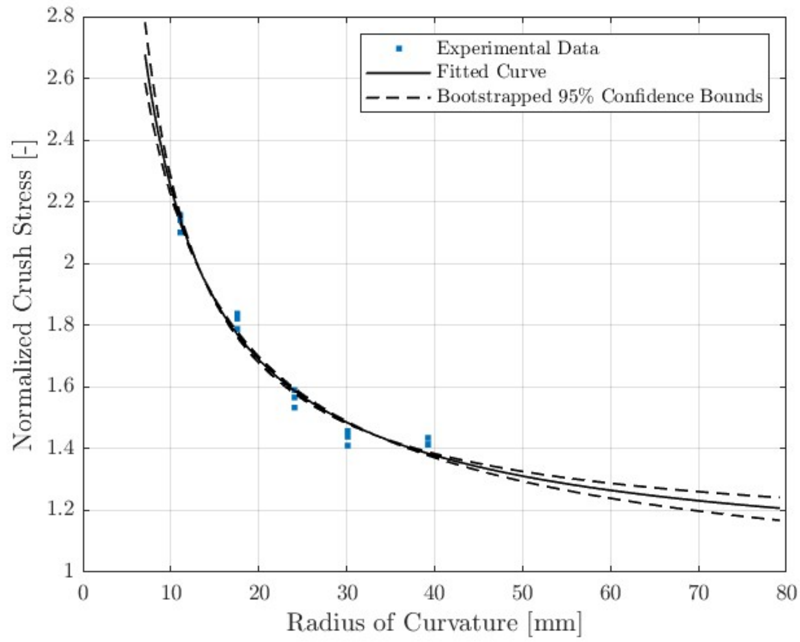


Figure A.3: Normalised variation of crush stress with radius of curvature for a CRFP laminate

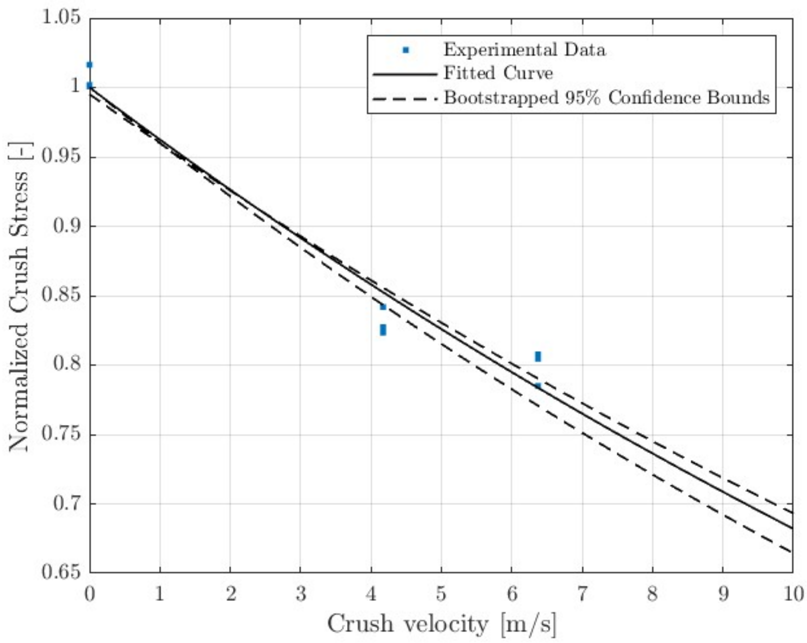


Figure A.4: Normalised variation of crush stress with crushing velocity for a CRFP laminate

Table A.9: Data contained in the DataCurvature file for the Hybrid laminate.

Flat Transition curvature [mm]	Flat Crush Stress [MPa]	Type of Fit	Fit coefficients	Lower confidence bound coefficients (95%)	Upper confidence bound coefficients (95%)
3031	61,06	power2	8,84	6,56	12,54
			-0,847	-0,671	-1,028
			0,989	0,818	1,102

Table A.10: Data contained in the DataVelocity file for the Hybrid laminate.

Type of Fit	Fit coefficients	Lower bound coefficients (95%)	Upper bound coefficients (95%)
exp1	1,00	1,00	0,995
-	-0,038	-0,0409	-0,036

A.2.3 Duplicated R5 RoC Curve

Table A.11: Hybrid RoC experimental data with duplicated R5 data (Obtained from [8]).

Mid-surface Radius [mm]	Average Crush Stress [MPa]
11,05	131,73
17,54	112,23
24,05	97,06
30,13	89,02
39,18	87,65
11,06	130,72
17,56	111,23
24,06	95,64
30,07	86,10
39,20	86,40
11,06	128,29
17,55	109,19
24,06	93,64
30,11	87,87
39,35	86,31
80	87,65
80	86,40
80	86,31

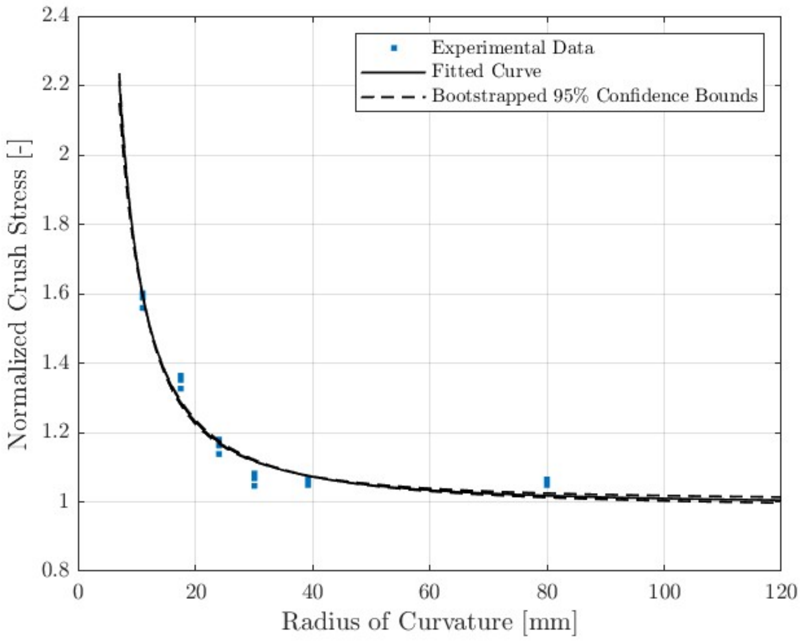


Figure A.5: Normalized RoC curve with duplicated R5 values.

Table A.12: Data contained in the DataCurvature file for the Hybrid laminate with duplicated R5 values

Flat Transition curvature [mm]	Flat Crush Stress [MPa]	Type of Fit	Fit coefficients	Lower confidence bound coefficients (95%)	Upper confidence bound coefficients (95%)
155,43	82,29	power2	25,59	19,90	30,62
			-1,556	-1,448	-1,643
			0,989	0,9777	1,002

Appendix B

`nodes` and `triangles` Structures for a Cone Simulation

partNodes.nodes

Fields	coords	neighbours	curvature	normals	inclination	ply	distances	CrushStress
1	[-20;0;200]	3x4 double	0	[0.9795,0.9922,0.9795;0.1065,0,-0.1065;-0.1709,-0.1247,-0.1709]	0	0	0	0
2	[-19.5324;-4.2994;200]	3x4 double	0	[0.9795,0.9337,0.9690;0.1065,0.3146,0.2133;-0.1709,-0.1709,-0.1247]	0	0	0	0
3	[-19.5324;4.2994;200]	3x4 double	0	[0.9795,0.9690,0.9337;-0.1065,-0.2133,-0.3146;-0.1709,-0.1247,-0.1709]	0	0	0	0
4	[-18.1515;-8.3978;200]	3x4 double	0	[0.8443,0.9005,0.9337;0.5080,0.4166,0.3146;-0.1709,-0.1247,-0.1709]	0	0	0	0
5	[-18.1515;8.3978;200]	3x4 double	0	[0.9337,0.9005,0.8443;-0.3146,-0.4166,-0.5080;-0.1709,-0.1247,-0.1709]	0	0	0	0
6	[-15.9219;-12.1035;200]	3x4 double	0	[0.7153,0.7899,0.8443;0.6776,0.6005,0.5080;-0.1709,-0.1247,-0.1709]	0	0	0	0
7	[-15.9219;12.1035;200]	3x4 double	0	[0.8443,0.7899,0.7153;-0.5080,-0.6005,-0.6776;-0.1709,-0.1247,-0.1709]	0	0	0	0
8	[-12.9477;-15.2432;200]	3x4 double	0	[0.5529,0.6423,0.7153;0.8155,0.7562,0.6776;-0.1709,-0.1247,-0.1709]	0	0	0	0
9	[-12.9477;15.2432;200]	3x4 double	0	[0.7153,0.6423,0.5529;-0.6776,-0.7562,-0.8155;-0.1709,-0.1247,-0.1709]	0	0	0	0
10	[-9.3682;-17.6702;200]	3x4 double	0	[0.3647,0.4648,0.5529;0.9153,0.8766,0.8155;-0.1709,-0.1247,-0.1709]	0	0	0	0

Figure B.1: Excerpt of the `nodes` structure after executing the `Treatment_v2.m` function partTriangles.triangles

Fields	coords	normal	indexes
1	[-19.5324,-20.6284,-20;-4.2994,-2.2435,0;200,195,200]	[0.9795;0.1065;-0.1709]	[2,30,1]
2	[-20,-20.6284,-20.6284;0,-2.2435,2.2435;200,195,195]	[0.9922;0;-0.1247]	[1,30,31]
3	[-20,-20.6284,-19.5324;0,2.2435,4.2994;200,195,200]	[0.9795;-0.1065;-0.1709]	[1,31,3]
4	[-19.5324,-20.6284,-19.6638;4.2994,2.2435,6.6255;200,195,195]	[0.9690;-0.2133;-0.1247]	[3,31,33]
5	[-19.5324,-19.6638,-18.1515;4.2994,6.6255,8.3978;200,195,200]	[0.9337;-0.3146;-0.1709]	[3,33,5]
6	[-18.1515,-19.6638,-17.7798;8.3978,6.6255,10.6977;200,195,195]	[0.9005;-0.4166;-0.1247]	[5,33,35]
7	[-18.1515,-17.7798,-15.9219;8.3978,10.6977,12.1035;200,195,200]	[0.8443;-0.5080;-0.1709]	[5,35,7]
8	[-15.9219,-17.7798,-15.0644;12.1035,10.6977,14.2698;200,195,195]	[0.7899;-0.6005;-0.1247]	[7,35,37]
9	[-15.9219,-15.0644,-12.9477;12.1035,14.2698,15.2432;200,195,200]	[0.7153;-0.6776;-0.1709]	[7,37,9]
10	[-12.9477,-15.0644,-11.6446;15.2432,14.2698,17.1745;200,195,195]	[0.6423;-0.7562;-0.1247]	[9,37,39]

Figure B.2: Excerpt of the `triangles` structure after executing the `Treatment_v2.m` function