

https://dspace.org/

Characterisation of bonded-repaired sandwich composite facing sheet using multiscale modelling and experimental testing

Makhate, Ikarabele

2020-01

https://hdl.handle.net/10539/30103

Downloaded from a DSpace repository. https://dspace.org/

CHARACTERISATION OF BONDED-REPAIRED SANDWICH COMPOSITE FACING SHEET USING MULTISCALE MODELLING AND EXPERIMENTAL TESTING

Ikarabele Makhate

A research dissertation submitted to the Faculty of Engineering and the Built Environment, University of the Witwatersrand, in fulfilment of the requirements for the Degree of Master of Science in Engineering.

Johannesburg, January 2020

DECLARATION

I declare that this dissertation is my own unaided work, except where otherwise acknowledged. It is being submitted for the Degree of Master of Science in Engineering to the University of the Witwatersrand, Johannesburg. It has not been submitted before for any degree or examination to any other University.

Signed this _____day of January 2020

_____

Ikarabele Makhate

ABSTRACT

Prediction of composite materials failure is a valuable resource for ensuring structural integrity of aerospace structures. A Multiscale Modelling (MsM) approach was developed with the objective to characterise the failure of a bonded-repaired Sandwich Composite Facing Sheet (SCFS). Failure characterisation of the carbon fibre SCFS was conducted for a four-point beam bending setup under static and fatigue loads.The MsM approach bridged three different finite element models across three length scales: from micro scale to meso scale and from meso scale to macro scale. The data processing and information transfer between the scales was facilited by Microsoft Excel Visual Basic for applications (VBA) scripts.For the experimental part of the study bonded-repaired sandwich composite facing sheet specimens were tested in a four-point bending setup. The specimens were manufactured and tested in a certified aircraft manufacturing facility to guarantee good quality. Comparisons of micro, meso and micro scale results were presented for both MsM and experimentation to validate the proposed MsM failure characterisation approach. Numerical and fractographical results from MsM and experimentation matched very well for the macro and meso scale but matched to a limited level for the micro scale. The limitation of obtainining progressive failure experiment data at the micro scale hampered this study. In general the MsM failure characterisation approach performed well against the experiment. It presents an opportunity to use an alternative composite materials failure characterisation method for future research.

Johannesburg, January 2020

ACKNOLEDGEMENTS

First of all I like to convey my gratitude to Prof. Jacob Muthu for his support and guidance throughout this research study.

Special thanks to the staff at Denel Aeronautics for their assistance with funding, manufacturing and testing activities.

Last but not least I am indebted to my family: Lieketso, Moeletsi and Thuto, for their understanding and support during the research study period.

Table of Contents

CHAPTER 1 - BACKGROUND 1
1.1 Research Background and Motivation 1
1.1.1 Types of Repairs 2
1.2 Research Question 5
1.3 Research Objectives 5
1.4 Study Layout 5
CHAPTER 2 - LITERATURE REVIEW 6
2.1 Applications of Composites 6
2.2 The Need for Composite Repairs 8
2.3 Types of Composite Materials Repairs 9
2.4 Types of Composite Materials Repair Failure Characterisation 11
2.4.1 Closed Form Solutions 11
2.4.2 Experimental Testing 11
2.4.3 Standard Finite Element Analysis and Damage Modelling 13
2.5 Multiscale Modelling and Progressive Failure Analysis 14
2.5.1 Homogenization 15
2.5.2 Static Fibre Failure Criterion 17
2.5.3 Static Matrix Failure Criterion 17
2.5.4 Fatigue Fibre Failure Criterion 18
2.5.5 Fatigue Matrix Failure Criterion 18
CHAPTER 3 - METHODOLOGY 19
3.1 Introduction 19
3.2 Experiments 19
3.2.1 Specimen Details 19
3.2.2 Tool Facing (TF) Sheet Fabrication 20
3.2.3 Core Layup and Bag Facing (BF) Sheet Fabrication 21
3.2.4 Bonded Repair Implementation 22
3.2.5 Static and Fatigue Test Procedures 24
3.2.6 Tool facing sheet stress calculation 30
3.2.7 Fractographical characterisation 30
3.3 Multiscale Modelling (MsM) and Analysis Procedure 31
3.3.1 Introduction 31

3.3.2 Input File Structure 33
3.3.3 Micro Model 36
3.3.4 Meso Model 43
3.3.5 Macro Model 46
3.3.6 Static and Fatigue Progressive Failure Analysis 50
CHAPTER 4 - RESULTS AND DISCUSSION 53
4.1 Static Experimental Results 53
4.2 Validation of Static Experimental Results using MsM Analysis 59
4.3 Static Failure Analysis 60
4.3.1 Experimental Micro Scale Failure 61
4.3.2 MsM Micro Scale Failure 64
4.3.3 Experimental Meso Scale Failure 67
4.3.4 MsM Meso Scale Failure 71
4.3.5 Experimental Macro Scale Failure 80
4.3.6 MsM Static Macro Scale Failure 82
4.4 Fatigue Experimental Results 85
4.5 Validation of Fatigue Experimental Results Using MsM Analysis 86
4.6 Fatigue Failure 86
4.6.1 Experimental Micro Scale Failure 86
4.6.2 MsM Micro Scale Failure 87
4.6.3 Experimental Meso Scale Failure 88
4.6.4 MsM Meso Scale Failure 89
4.6.5 Experimental Macro Scale Failure 91
4.6.6 MsM Macro Scale Failure 92
CHAPTER 5 - CONCLUSION 93
CHAPTER 6 - REFERENCES 95
A. APPENDIX A 99
B. APPENDIX B 101

List of figures

Figure 1.1 Composite weight saving and historical application 1
Figure 1.2 Composite application trends for Airbus aircraft [Roland] [2] 1
Figure 1.3 Repair scheme on the inner skin of a sandwich composite [Chp 7 FAA] Multiscale Modelling (MsM) 4
Figure 2.1 Carbon Fibre demand projections by weight for the four major applications analysed [16] 6
Figure 2.2 Auto light-weighting agenda of major global Original Equipment Manufacturers (OEMs) and their models from 2013 to 2020 [20] 7
Figure 2.3 Standard operational procedure for repair process. 10
Figure 2.4 Progressive failure characterisation of fibre reinforced composites [39] 12
Figure 2.5 RVE colour matching of vertices, edges and surfaces 15
Figure 3.1 Sandwich panel specimen 19
Figure 3.2 Manufacturing sequence of TF sheet. 20
Figure 3.3 Tool Facing sheet cure cycle for the tool side facing sheets 20
Figure 3.4 Manufacturing process of core layups and bag facing sheet 21
Figure 3.5 Bag Facing sheet cure cycle for the tool side facing sheets 22
Figure 3.6 Repair operation sequence 23
Figure 3.7 Cutting tools and specimen making 24
Figure 3.8 Strain gauge positions for E1,E2,E3 and E4 positions over the repair area 25
Figure 3.9 Static and fatigue testing machines and four-point bending setup 26
Figure 3.10 Sine wave loading applied to fatigue specimens 27
Figure 3.11 Roller positions on the specimen 28
Figure 3.12 Strain gauge positions and critical points for the repaired facing sheet 29
Figure 3.13 Multiscale modelling strategy framework 32
Figure 3.14 Patran\\Nastran input file structure 33
Figure 3.15 Micro model input file sample 34
Figure 3.16 Micro model input file sample Bulk Data Entries 35
Figure 3.17 Micro model input file sample bulk entries continued 36
Figure 3.18 Micro model 37
Figure 3.19 Generating the micro RVE input files 38
Figure 3.20 Colour coded node pairs for a Micro RVE model 39
Figure 3.21 Six different cases to be solved in order to calculate the homogenised elastic properties of the RVE 42
Figure 3.22 Meso model element groups for a plain weave 44
Figure 3.23 Six cases to be solved for the homogenised material properties of the meso RVE 46
Figure 3.24 Macro model geometry and materials 47
Figure 3.25 Macro model element groups 49
Figure 3.26 General MsM progressive failure analysis flow diagram 51
Figure 4.1 Displacement versus applied transverse load 53
Figure 4.2 Box and whiskers plot of repaired facing sheet failure loads 54
Figure 4.3 Strain gauge positions on the repaired facing sheet 55
Figure 4.4 Parent ply and repair ply strains of the sandwich beam experimental results 56

Figure 4.5 Experimental strains versus parent laminate stress at position E1 57
Figure 4.6 Tool facing sheet failure areas for experimental specimens.....58
Figure 4.7 Load versus strains from macro model (MsM) 59
Figure 4.8 Macro model stress versus strain at critical point B (macro element 7688).....59
Figure 4.9 Critical points A and B on the repaired facing sheet.....60
Figure 4.10 SEM pictures of static micro scale failure at critical points A (Warp Tows).....62
Figure 4.11 SEM pictures of static micro scale failure at critical points B (Warp Tows).....63
Figure 4.12 Von Mises stress plot of the micro model (GPa).....64
Figure 4.13 Tensile stress plot of the micro model matrix and fibres 65
Figure 4.14 Element centroid stress components along Z-dir arrow 66
Figure 4.15 Element centroid stress components along Y-dir arrow.....66
Figure 4.16 Static damage to micro RVE as a result of various loads (meso element 6630).....67
Figure 4.17 Ply delamination failure at critical point B.....69
Figure 4.18 Warp tow rupture and weft tow splitting failures at critical point B (handheld microscope pictures) 70
Figure 4.19 Element list positions on a stress plot of the meso model warp tows 71
Figure 4.20 Element centroid stress components along the warp length.....72
Figure 4.21 Element centroid stress components across the warp width.....72
Figure 4.22 Element list positions on a stress plot of the meso model weft tows 73
Figure 4.23 Element centroid stress components along weft length.....74
Figure 4.24 Element centroid stress components across the weft width 74
Figure 4.25 Element list positions on a stress plot of the meso model matrix.....75
Figure 4.26 Element centroid stress components along warp length.....76
Figure 4.27 Element centroid stress components along weft length.....76
Figure 4.28 Static damage to meso RVE as a result of various loads (macro element 7686) 77
Figure 4.29 Meso scale warp tow failure edge for MsM and experimental specimens.....78
Figure 4.30 Meso scale weft tow failure morphology for MsM and experimental specimens.....79
Figure 4.31 Matrix failure at the macro scale at critical point A (handheld microscope pictures) 80
Figure 4.32 Macroscale: Tool facing sheet (repair plies) rupture line **(Critical Point A)**.....81
Figure 4.33 Macro scale: Tool facing sheet (parent laminate) rupture line **(Critical Point B)** ...81
Figure 4.34 Tool facing sheet failure areas for MsM and experimental specimens 82
Figure 4.35 Element list positions on a stress plot of the macro model (critical point A).....83
Figure 4.36 Element list positions on a stress plot of the macro model (critical point B).....83
Figure 4.37 Element centroid stress components for critical point A.....84
Figure 4.38 Element centroid stress components for critical point B.....84
Figure 4.39 Experimental fatigue loading cycles against tool facing sheet stresses: S-N curve ..85
Figure 4.40 MsM fatigue loading cycles against tool facing sheet stresses: S-N curve.....86
Figure 4.41 Fatigue microstructural failure at CP B.....87
Figure 4.42 Micro RVE Fatigue at 7.0 kN with increase cycles (meso element 6666).....88
Figure 4.43 Meso scale failure observations at CP B 89
Figure 4.44 Fatigue damage to meso RVE as a result of 7.0 kN at various cycles (macro element 7677) 91
Figure 4.45 Macro scale failure cases of the repaired facing sheet 92

List of Tables

Table 2.1 Reinforcing fibers commonly use in aerospace applications. [22] 7
Table 2.2 Properties of polymeric matrices commonly used in aerospace sector [22] 8
Table 2.3 Fatigue properties for the carbon fibres [71] 18
Table 2.4 Fatigue properties for the epoxy resin [72] 18
Table 3.1 Loads for static specimens 26
Table 3.2 Static specimens for loading to failure 26
Table 3.3 Applied fatigue loads for specimens 27
Table 3.4 Micro RVE geometric properties 37
Table 3.5 Material properties for Carbon Fibre (HTS40 E13 3K) and Matrix (Epoxy 5276-1) 37
Table 3.6 Micro RVE MPC groups 39
Table 3.7 Meso model geometric properties 43
Table 3.8 Stacking sequence for the laminate before repair implementation 47
Table 3.9 Material properties for Aluminium and Nomex honeycombs 48
Table 4.1 Applied failure loads for specimens 54
Table 4.2 Tool side facing sheet laminate experimental material properties results 57
Table 4.3 Tool side facing sheet laminate MsM material properties results 60
Table 4.4 Applied fatigue loads for specimens 85

Nomenclature

$U_{_i(a1, x2,x3)}$	Displacement
$a_{_1i}$	RVE length
$E_{_ij}$	Average macro strain
E_{ii}^{j}	Elastic Modulus
$C_{_equiv}$	Stiffness matrix
$x_{_3}$	Variable for displacement
V_{Total}	RVE volume
V_i	Element volume
S_{ij}	Average macro stress
N_f	Cycles to fatigue failure
S_{max}	Maximum applied fatigue stress
S_{uts}	Ultimate tensile stress
D_m	Matrix damage indicator
D_f	Fibre damage indicator
σ_{ii}	Applied static stress
N_f	Loading cycles to failure

Abbreviations

CFRP	Carbon Fibre Reinforced Plastics
MRO	Maintenance Repair and Overhaul
AMC	Aviation Management Corporation
EASA	European Aviation Safety Agency
FAA	Federal Aviation Authority
CFA	Closed Form Analysis
FEA	Finite Element Analysis
MsM	Multiscale Modelling
SCFS	Sandwich Composite Facing Sheet
OEM	Original Equipment Manufacturer
NDT	Non-Destructive Testing
MRB	Material Review Board
SRM	Structural Repair Manual
ADL	Acceptable Defects Limits
CAIIAC	Consortium of Accelerated Innovation and Insertion of Advanced Composites
FPF	First Ply Failure
ASTM	American Society for Testing and Materials
NASA	National Aeronautics and Space Administration
SEM	Scanning Electron Microscope
MMF	Micromechanics of Failure
TTSP	Time-temperature Superposition Principle
MPC	Multi-point Constraints
RVE	Representative Volume Element

CHAPTER 1 - BACKGROUND

1.1 Research Background and Motivation

Composite materials are being increasingly used in various engineering applications such as aerospace and automobile structures [1]. In particular, carbon fibre reinforced plastics (CFRP) are used as an alternative material to aluminium for aerospace structures. The increasing applications of CFRP in aircraft structures are as a result of their weight saving potential, corrosion free properties as well as their higher fatigue life cycles when compared to aluminium. Moreover these materials offer lower maintenance cost since they are more robust than metals [2]. Based on weight saving potential of CFRP, it can be concluded that composite will dominate future aircraft structures as shown in Figure 1.1.



a) Future weight saving potential of CFRP in comparison with aluminium[2]



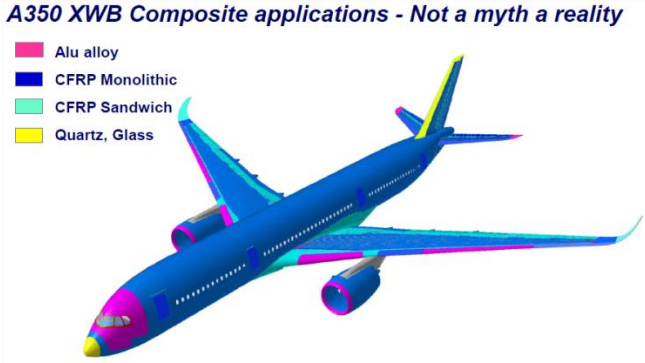
b) Application of composites to Airbus family of aircraft over the past 30 years[2]

Figure 1.1 Composite weight saving and historical application

Within the airbus family of aircraft, there has been an increasing trend of replacement of aluminium with CFRP. Back in 1985 the A310 started with 5% composite structural weight, currently 53 % of the A350 structural weight is in composites [1].Figure 1.2 shows typical composite aircraft parts in 1988 and 2013.



a) 1988 onwards



b) 2013 onwards

Figure 1.2 Composite application trends for Airbus aircraft [Roland] [2]

For over thirty years, Airbus has demonstrated that it can replace most of the aircraft aluminium parts with composite materials. Initially the target areas were the empennage, flaps, engine cowlings and the landing gear doors. Thereafter, the belly fairing and some parts of the rear

fuselage joined the group of composite parts. Of late the rest of the fuselage and wings are now made from composites as shown in Figure 1.2.

With the increasing application, comes the challenge for engineers to develop repair schemes that are efficient for non-conforming parts in manufacturing and damaged parts in-service. Repair rather than replacement of damaged parts is preferable when the replacement time is limited and the cost of replacement is higher than that of implementing a repair. For safety purposes manufacturing companies and Maintenance Repair and Overhaul (MRO) companies need to fully understand the composite repair design process. If the damage area is larger than the allowed size as explained in the structural repair manual then the repair has to go through extensive analysis and test substantiation as explained in the AMC-20 [3]. This process is expensive and time consuming. The repaired structure should meet the requirements of ultimate strength, structural stiffness, aerodynamic flatness, weight balancing and cyclic fatigue life [4] (durability).

1.1.1 Types of Repairs

There are two major types of repairs namely bonded and bolted repairs. Bolted repairs are used on thick monolithic composites that can withstand high bolt bearing loads. Bonded repairs are mainly used where the sandwich facing sheets are too thin to carry high bolt bearing loads and also for preventing damage to the sandwich core by water ingress. They also need to satisfy aerodynamic flatness requirements. Bonded repairs can be scarfed, overlapped or stepped as shown in Figure 1.3. A lot of repairs on composite parts are done during manufacturing as a result most repairs are of a bonded nature. Within the manufacturing facility, neat strong bonded repairs are easy to implement since all the required tools and facilities are accessibly. In the aerospace industry, repairs schemes are used to salvage damaged or non-conforming aircraft parts that are too expensive to replace or would take too long to replace with new parts. These repair schemes are validated by expensive experimental test procedures as required by airworthiness authorities (Federal Aviation Authority: FAR 23.305 and 23.307). Often these experimental tests need to be repeated before satisfactory results are obtained.

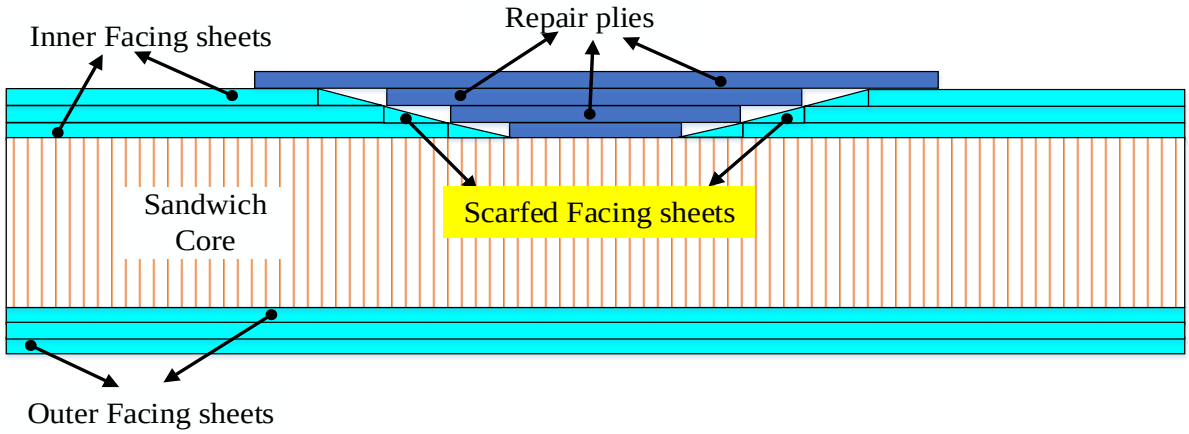
A complex mix of closed form analysis (CFA), finite element analysis (FEA) and experimental tests are used to minimise the risk of structural failure while meeting all technical, regulatory and customer requirements. CFA and FEA alone have been found to be inadequate to predict the failure behaviour under all possible loading scenarios. FEA models are usually too simple to provide engineers with detailed information on failure at micro and meso scales of the composite material. To acquire failure information at the micro scale (individual fibre diameter of 6 - 9 micrometres) engineers need to build highly detailed finite element models that would be difficult to build in order to account for microstructural features. The meso scales models (3 to 6 millimetre) would also just be as difficult when created to represent the fabric design repeated over the macro scale (0.5 metres). Experimental testing on its own can also be prohibitively expensive [5]. In some cases standard finite element models are used to gain an in depth understanding of the bond-line failure phenomena [6].

A building-blocks approach is widely followed to validate aircraft structural designs including joints and repairs. The approach is implemented in phases such a way that the simple tests are completed first followed by complex tests. The phases are

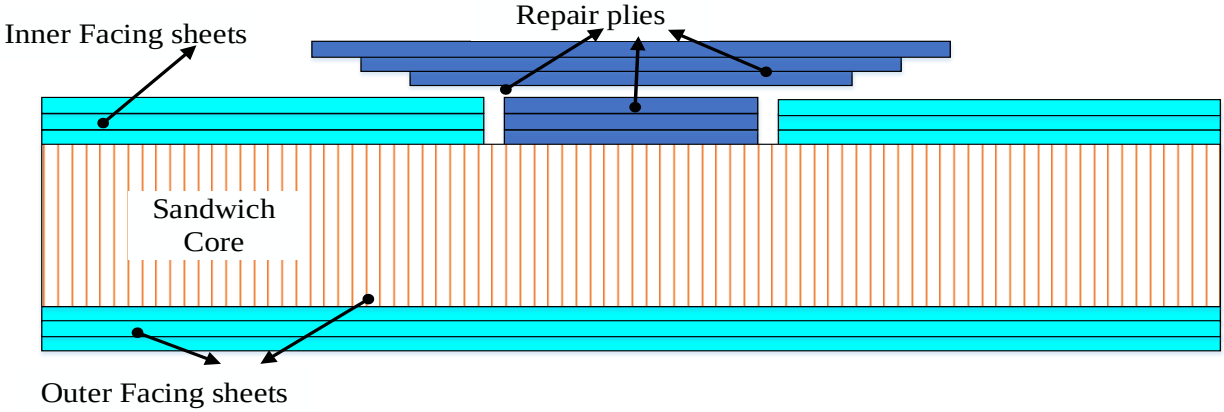
1- Coupon,
2- Detailed design,
3- Subassembly and
4- Assembly testing.

Coupon testing involves simple test to determine material properties used in the complex design. For example this phase for a Sandwich Composite Facing Sheet (SCFS) repair would be the determination of material properties for plies, foam adhesive and cores. Detailed design testing is the phase where the structural integrity of a specific design feature like a repair of a monolithic laminate is verified. This design feature uses the material properties determined in the coupon testing phase. After the detailed design testing phase comes the subassembly testing phase. In this phase the structural integrity of a combination of components and design details is validated. The final phase of the building-blocks approach is the validation of the complete aircraft, wing or fuselage. One can use either FEA or experimental testing methods in the building blocks approach. The dominant and most trusted method of repair failure characterisation is to use empirical data from experimental tests. This method is both expensive and limiting as it requires a test for every modification of the repair scheme and in some cases repetition of tests.

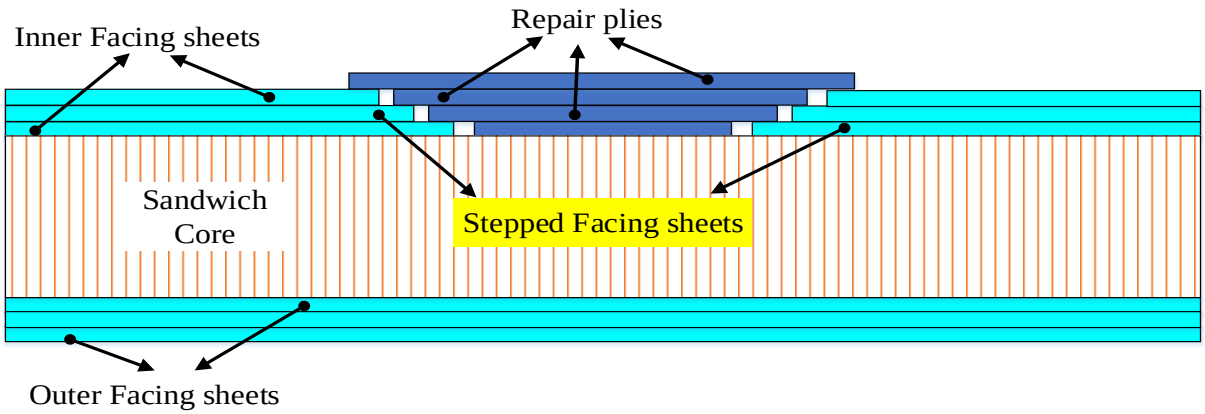
However, if a Multiscale Modelling technique (MsM) could be added in to the building-blocks approach, it could substantially eliminate the necessity of expensive structural testing and hence reduce the development costs and improve the accuracy of validation processes. The cost and time benefits could be realised by combining MsM, FEA and experimental testing in the building-blocks approach such that more FEA and MsM validation is used rather than the traditional experimental validation. This means more MsM than experimental testing would be used in the coupon testing phase to determine material properties. Different materials could be applied to concept designs without costly coupon testing with different materials. This structural justification approach would eliminate the current long lead time and expensive structural experimental validation processes.



(b) Overlapped Repair



(b) Overlapped Repair



(c) Stepped Repair

Figure 1.3 Repair scheme on the inner skin of a sandwich composite [Chp 7 FAA] Multiscale Modelling (MsM)

As explained before, the past research activities have focused on using the experimental tests based building-blocks approach to characterise bonded repair failures. The major setback of this approach is its failure to provide researchers with failure details at the micro and meso scales. The failure of composites generally starts at the micro scale of the reinforcing fibres [8]. The cumulative effect of micro failure then progresses to the ply level before causing ultimate failure at the laminate level. To reach micro and meso scale failure characterisation, there is a need for analysing at three length scales namely: micro, meso and macro scales [9]. This will allow damage analysis of the composite at the matrix and fibre scales as well as inter-ply failure analysis. Hence sophisticate models and advanced computational techniques are necessary to adequately simulate the interrelated behavioural characteristics of CFRP [10]–[13]. A more practical approach to study the details of the bonded repair structures at the required scales is the Multiscale Modelling (MsM) approach [14] that is currently gaining popularity in various engineering applications.

MsM is a structural analysis tool that researcher are using to study complex composite materials. Its main advantage over other analysis tools is that one can analyse different length scales of a material at the same time. Composite materials are inherently multiscale from the reinforcing fibres to large sandwich panels that are used in aircraft fairings and floors. Hence MsM is naturally the most suitable tool to comprehensively study the material behaviour of a composite material through its constituents.

The MsM approach should serve as a guideline to material selection for the design of bonded repairs for SCFS. Development of such an approach will eliminate costly trial and error practical tests usually used in industry [15]. For safety purposes, it is important to demonstrate the structural integrity of a bonded repair since the stress state around the repair scheme will be higher than the baseline design stress state. The increase in the stresses of the material around the bond line is as a result of the removal of material. This study proposes to characterise failure of a bonded repaired SCFS using multiscale modelling.

Hence the purpose of this research is to characterise the failure of bonded repaired sandwich composite facing sheets (SCFS) using MsM and validate the results with experimental testing. Once validated, MsM could be used by aircraft components manufacturers and MRO companies to design the most efficient repairs quickly. This research study is the first step in making a transition from experimental testing to MsM in the bonded repair analysis process.

1.2 Research Question

Based on the literature review there is a need to improve the failure characterisation of bonded repaired SCFS at the various length (micro and meso) scales. Furthermore MsM can be used to improve the failure characterisation of bonded repairs by going beyond the current limits set by experimental testing, closed form solutions and standard FEA. The question on whether MsM can characterise progressive failure of a SCFS bonded repair is the subject of this research study.

The failure characterisation exercise will cover both the static and fatigue failure of the area around a bonded repaired SCFS. Information collected from this characterisation can be used to demonstrate that MsM can improve the failure characterisation and reduce expensive experimental validation currently practised in industry.

1.3 Research Objectives

This research study will be conducted with the following objectives;

- Multiscale Modelling (MsM) failure characterisation of bonded repaired SCFS at micro, meso and macro scales.
- Experimental and Morphological characterisation of bonded repaired SCFS at micro, meso and macro scales.
- Validation of MsM results with experimental and morphological results

1.4 Study Layout

This research report structure starts with an introduction (Chapter 1) to fibre reinforced composites bonded repairs and the multiscale modelling (MsM) analysis approach. The literature review (Chapter 2) starts with the general applications of fibre reinforced composites across various industries. Thereafter the needs of composite repairs for economic and safety purposes in the aerospace industry are discussed. Followed by a discussion of the shortcomings of repair validation and verification methods. The last section of the literature review outlines recent hierarchical multiscale modelling techniques used for composite material constitutive behavioural and progressive failure predictions.

The methodology is provided in chapter 3 where the multiscale modelling method study is explained across three length scales. The micro, meso and the macro length scale models are presented as well as the application of periodic boundary conditions on their surfaces. Execution of the progressive failure characterisation process by means of visual basic subroutines is also covered. This chapter is concluded by reporting the experimental verification and MsM validation procedure for static and fatigues loading cases. Chapter 4 provides the results of static and fatigue failure results of SCFS repair scheme. The validation of the MsM model is discussed and confirmed by means of experimental results. Chapter 5 concludes with research findings and recommendations. References and appendices are included at the end of the report.

CHAPTER 2 - LITERATURE REVIEW

2.1 Applications of Composites

There are four major application sectors for CFRP worldwide namely; wind energy, aerospace, automotive and pressure vessels [16]. Demand is growing at a rate of 12 – 15 % annually and it is based on the pursuit of more fuel efficient transport vehicles and strong and light weight structures. Prior to 2008 the aerospace sector was the primary consumer of CFRP but there has been a strong demand from the wind energy sector that is poised to lead the demand. Close to half of the global demand is for aerospace parts where most of these parts are for commercial airliners and regional jets. At the current CFRP usage growth rate the cumulative demand will increase to more than 160,000 tonnes in the next ten years. Refer to Figure 2.1 below which shows an increasing demand for the future. Forecasts from [17] predict that CFRP revenues in aerospace and defence applications will exceed US$1.4 billion in 2020.

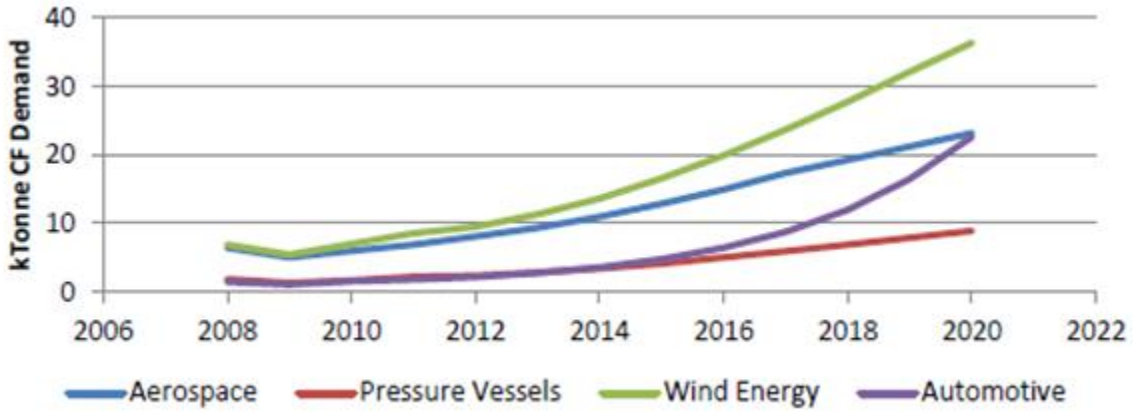


Figure 2.1 Carbon Fibre demand projections by weight for the four major applications analysed [16]

There is more demand for CFRP coming from sports and leisure equipment but to a lesser extent than the major four contributors mentioned in the paragraph above. In sporting equipment sector CFRP are offering high strength, lightweight, higher degrees of freedom for design, easy processing and formability [18], [19]. In sports CFRP are used to make skis, golf clubs, bicycles, bows and tennis rackets. Developed countries are taking the lead in creating research bodies to study in detail the material characteristics of composites for sports applications.

Increasing pressure from environmental groups is forcing Original Engineering Manufactures (OEM) from the aerospace and automotive sectors to improve the efficiency of their transport vehicles. Environmental groups believe that more than a quarter of greenhouse gases originate from transport vehicles [20]. The targets for weight reduction by the major automotive manufacturers are summarised in Figure 2.2 below. Current legislature is also prompting OEM to review their transport vehicles with the objective to improve fuel efficiency. If the structure of the vehicle is redesigned to be lighter then CFRP is the material of choice for replacement of heavier materials. This means there will be an increasing application of CFRP and inherent increase for a better understanding of the behaviour of CFRP. For instance, CFRP have been found to be more durable than metals to fatigue from repeated take-off and landing cycles [21] but the progressive micromechanics failure of CFRP still needs to be comprehensively understood. The maximum

benefits of CFRP can only be fully utilised when the material is completely understood to the micro level.

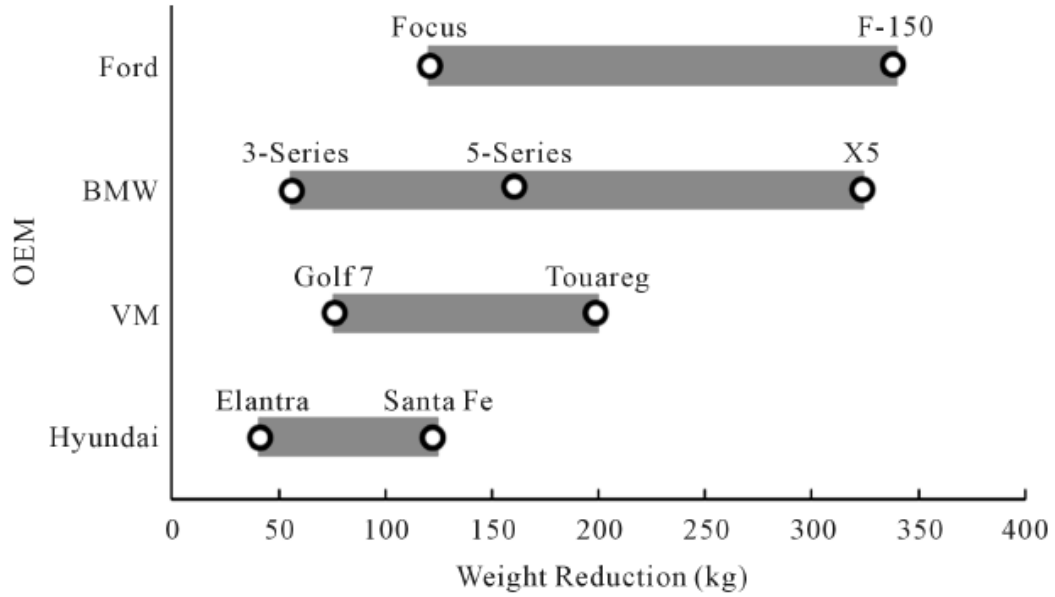


Figure 2.2 Auto light-weighting agenda of major global Original Equipment Manufacturers (OEMs) and their models from 2013 to 2020 [20]

In Table 2.1 carbon is the leading fibre material in terms of lightness and high strength. Its properties are suitable for applications in the most demanding structures. In Table 2.2 the properties of common matrices are compared. These tables are useful for selection of composite constituents for different operating conditions. [22]

Table 2.1 Reinforcing fibers commonly use in aerospace applications. [22]

Fibre	**Density (g/cc)**	**Modulus (Gpa)**	**Strength (Gpa)**	**Applications**
GLASS				
E-glass	2.55	65 - 75	2.2-2.6	Small passenger a/c parts, aircraft interiors, secondary parts
S-glass	2.47	85-95	4.4-4.8	Highly loaded parts in small passenger a/c
ARAMID				
Low modulus	1.44	80-85	2.7-2.8	Fairings; non-loaded bearing parts
Inter. Modulus	1.44	120-128	2.7-2.8	Radar domes, some structural parts; rocket motor castings
High Modulus	1.48	160-170	2.3-2.4	Highly loaded parts
CARBON				
Standard modulus (high strength)	1.77-1.80	220-240	3.0-3.5	Widely used for almost all types of part in a/c, satellite, antenna dishes, missiles, etc.
Inter. Modulus	1.77-1.81	270-300	5.4-5.7	Primary structural parts in high performance fighters.
High modulus	1.77-1.80	390-450	2.8-3.0 4.0-4.5	Space structures, control surfaces in a/c
Ultra-high strength	1.80-1.82	290-310	7.0-7.5	Primary structural parts in high performance fighters and spacecraft.

Table 2.2 Properties of polymeric matrices commonly used in aerospace sector [22]

Thermosets				**Thermoplastic**
Forms cross-linked networks in polymerization curing by heating				**No chemical change**
Epoxies	**Phenolics**	**Polyester**	**Polyimides**	**PPS, PEEK**
Most popular	Cheaper	Cheap	High temp. applications 300 Degrees C	Good damage tolerance
Low shrinkage (2-3 %)	More shrinkage	High shrinkage (7-8%)		
	Brittle than epoxies	Brittle	Brittle	
Good storage stability to make prepregs	Less storage stability - difficult to prepreg	Difficult to prepreg		Infinite storage life but difficult to prepreg
Absolute moisture (5-6 %) causing swelling and degredation of high temp. properties	Absorbs moisture but no significant effect of moisture in working service range	Less sensitive to moisture than epoxies		No moisture absorption

CFRP can be manufactured by a variety of techniques depending on the desired degree of quality. Low quality CFRP may be manufactured by introducing the resin to dry fibres by a brush or roller and curing the resin at room temperature. For high quality products prepreg materials are usually required and the curing process occurs in an autoclave. Aerospace products are classified as high quality products so the latter manufacturing method is used.

2.2 The Need for Composite Repairs

Like all other engineering materials composites need to be repaired when defective in manufacturing or damaged in service. From manufacturing operations composite structures are likely to incur defects during fabrication and assembly phases. The vulnerability to damage of the composite structure extends to the service life where crew operations and environmental factors can inflict damages. Comprehensive repair technologies are required for composite structures that are increasingly being used across all industries.

Defects and damages do not just manifest themselves as a reduction in the compressive and bending strength of the CFRP. They can also hide inside the composite part where they are not visible to the naked eye hence non-destructive testing (NDT) techniques need to be employed to characterise them. When the defects and damages are large enough to cause serious strength reduction a repair needs to be done as quickly as practically possible. The repair types and associated damage size limits are defined in the structural repair manual (SRM) of the aircraft. Presently extensive testing is used to develop repairs but this is expensive and prohibitively time consuming. Current research is focused on determining size limits, inspection intervals and types of repairs and damages [22]. This research study is focused on characterising damage within a repair as an area of high stress concentration using MsM.

Common manufacturing defects include but are not limited to the following; fibre waviness, void formation, multiple porosity, delamination, inclusion of impurities, disbonds, core crushing, fibre misalignment, wear and scratches and tool impact damage. The following are defects that could lead to repair implementation. In sandwich part fabrication the core that supports the thin facing sheets could get crushed and fail to provide the necessary support to the facing sheets. There are also cases where the disbonds may form between the facing sheets and core layer and prevent load

transfer between the two layers. Inclusions can also be found within the laminate as a result of carelessness during the layup operation. Depending on the size of the defect as set by the material review board (MRB), structural repair manual (SRM) and the acceptable defect limits (ADL). The defect could be repaired by a cosmetic repair where its visual appearance is removed without significantly altering the load paths of the part. When material has to be removed and replaced the load paths of the part are altered and the repair is implemented through the SRM or a new repair has to be implemented.

During the long service life of an aircraft there are numerous opportunities that could lead to damaged parts in operation. In flight an aircraft could be hit by a lightning strike, a bird or hail causing delamination and structural degradation. Take-off and landing phases are also prone to hits from runway debris on the fuselage belly and bird strikes. Airport ramp impact damage caused by ramp vehicles and equipment can cause scratches, dents, punctures, fibre breakage and delamination. Surface punctures if left unattended could lead to further damage from moisture ingress into sandwich parts. CFRP parts exposed to excessive temperatures and ultraviolet radiation can also incur matrix material degradation. Based on the ADL specified in the SRM the in service damage can cause part replacement or grounding of the aircraft. Or the damaged part can be removed immediately to be fixed later in an MRO facility. When the aircraft is grounded due to the severity of the damage a repair needs to be developed if one does not exist already.

The Georgia Manufacturing Technology Institute through its Consortium of Accelerated Innovation and Insertion of Advanced Composites (CAIIAC) has identified a number of fundamental research needs [23]. Amongst them are the requirements of damage growth assessment and prediction approach for composite structures used in aerospace. Generally cracking is noted in composite structures even when analysis proves otherwise. Repairs are usually developed for specific types and sizes of defects and damages through tests.

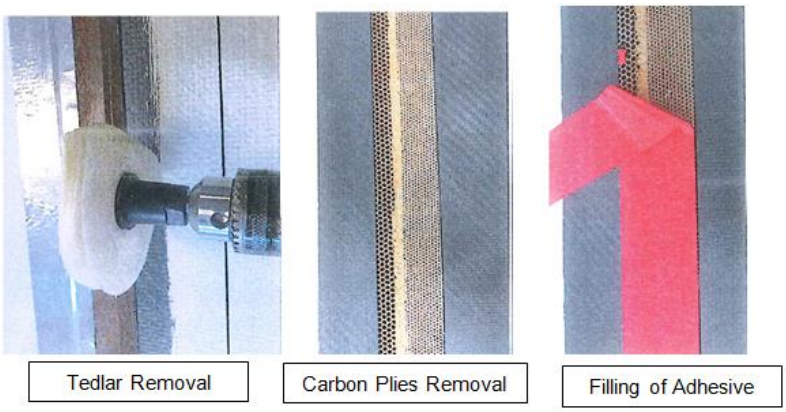
2.3 Types of Composite Materials Repairs

There are basically two types of repairs for fibre reinforced composites; one can use a bonded or bolted repair. In aerospace bonded repairs are common since they are aesthetically appealing but the preference is also technical since they impart no aerodynamic penalties. But for CFRP fuselages the repair development challenge is new since fuselage skins are thick and the fuselage itself is pressurized in contrast to control surfaces. CFRP Fuselages are not common in the airline transportation industry. They were only introduced by the major airliner manufacturers through the B787 from Boeing and A350 from Airbus [24]. Since these airliners have not been in service for a long time there is very little standardisation of repair training and techniques. Development of both bolted and bonded repairs for pressurised structures is still in its infancy. While lot of confidence goes to bolted repairs for their structural integrity in primary load paths bonded repairs have the advantages of quick implementation and no aerodynamic penalties. A lot of research needs to be done to increase the confidence of aircraft end users in fuselage repairs particularly multiple repairs.

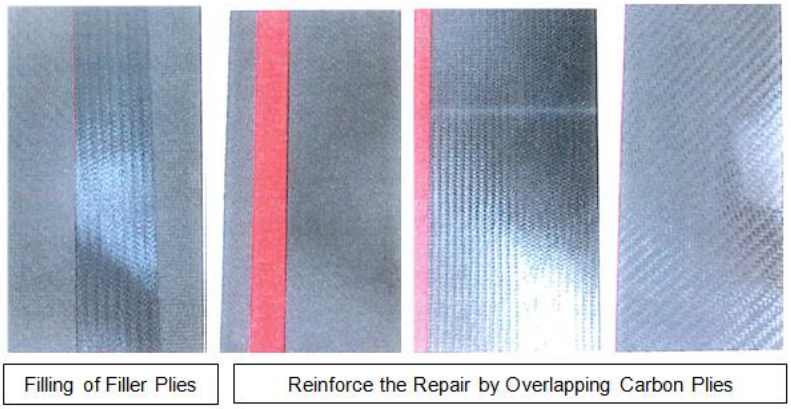
The reworking and repair of CFRP structures is extremely time consuming when the repair is done manually as is the common practice [10], [13], [25], [26]. For sandwich laminates the process can be simplified by using a step or overlap repair since the facing sheets are very thin. The facing sheet can be removed in one step hence saving time and simplifying the operation. The resulting stress concentration effect can be reduced by overlapping the patching layers to introduce the load into the patch gradually. Simple repairs are preferred since they reduce aircraft-on-ground times.

The manufacture of CFRP parts for aerospace applications follows a very strict process to ensure that only the highest quality parts are used for aircraft structures. In the case of a sandwich panel with CRFP facing sheets the process starts with the preparation of the laminate tool. Followed by removal of the prepreg material from the freezer storage and cutting it. Thereafter the plies are laid up and bagged. The facing sheet can then be cured to form the tool side of the panel. After curing the tool side it is debagged for the second lay-up of the sandwich core and the bag-side of the facing sheet. When the lay-up of the second bagging is complete then the complete panel is cured for the final stage of curing.

The completely cured panel progresses to the trimming and machining of the panel according to design specifications. At this point the panel needs to go through inspection to make sure that its fabrication and machining was done according to design specifications. When nonconformity is discovered it is removed and a repair is implemented if recommended. The repair follows the same fabrication process as the tool side and bag side facing sheets and the whole part is cured in an autoclave to consolidate the repair layers. If the repair is done outside the manufacturing facility a pre-cured patch or fresh prepreg layer stack cured by heat blankets is used. The final stage of the CFRP part manufacturing is painting and finishing. Figure 2.3 below summarises the standard operational procedure for a bonded repair.



The repair process starts with the removal of the tedlar layer and facing sheets. Thereafter the adhesive film is applied.



Once the damaged area is removed the filler layers are applied and a layer of film adhesive. And finally the reinforcing layers are applied.

Figure 2.3 Standard operational procedure for repair process.

2.4 Types of Composite Materials Repair Failure Characterisation

The literature review was conducted to explain the various research works associated with bonded repairs of sandwich composites used in aircraft structures. Generally there are three methods used to characterise failure of bonded repairs namely, experimental testing, closed form solutions and standard finite element analysis (FEA).

2.4.1 Closed Form Solutions

When extensive test data is reduced to a simple set of equations to deliver quick solutions the result is what is usually called closed form solutions. These types of solutions have found wide spread applications in engineering design due to their simplicity and repeatability. Soutis et al [27] employs linear elastic fracture mechanics to derive analytical expressions for predicting bonded-repair strengths. Analytical solutions are conservative but acceptable for design purposes. Brian Falzon [28] from Garteur presented a closed form solution for bonded repair strength prediction recommended by various aircraft manufactures. In most cases the aircraft manufactures will often have in-house methods used for design of patch repairs that have been developed from their detailed experimental tests. A review of bonded composite joints of fibre reinforced, monolithic and sandwich, composites was conducted by Banea and Silva [29]. They investigated effects of fabrication parameters and environmental factors on joint behaviour. The review also covered published analytical and numerical (standard FEA) methods of failure prediction. No failure method was chosen to yield accurate predictions but progressive failure analysis was recommended for its promising performance. They also highlighted the requirement of an accurate stress and strength prediction method to decrease the reliance on expensive experimental testing.

Airworthiness authorities also publish guidelines and analysis methods to be used for bonded repairs as detailed in [30], [7]. Closed form solutions are usually faster than experimental tests in provision of results but they suffer from lack of flexibility. In sandwich composites analysis closed form solutions are applicable only up to first ply failure (FPF) but they say nothing about the behaviour of the laminate beyond FPF. Another disadvantage of closed form solutions is their heavy reliance on empirically determined parameters. While closed form solutions lack flexibility MsM can provide the required flexibility at more than one length scale. A combination of FEA and composite failure theories can be used to characterise bonded repaired SCFS. In this analysis method composite failure theories like Hill, Tsai-Wu, Hoffman, Hashin and Puck are used to determine when to degrade a material's Young modulus or shear modulus as it is progressively loaded. The degradation can be abrupt or gradual depending on the composite under consideration. This combination of FEA and composite failure theories is call progressive failure analysis and has been used for sensitivity analysis of bonded repairs as presented in the following paragraph.

2.4.2 Experimental Testing

As the name implies testing methods rely on capturing the full history of the material degradation history from first ply failure to complete rupture. This failure characterisation approach relies on lots of testing using generally accepted testing methods like the American Society for Testing and Materials (ASTM) or British Standards. Testing yields accurate results but at a very high cost. Design and testing of sandwich and monolithic composites is covered widely in two NASA reports [8, 9]. These reports offer guidelines on fabrication and testing of sandwich laminate composites repairs.

Bonded repairs were implemented on composite sandwich panels that were subsequently loaded in edgewise compression by Guan et al [33] and [34] Their experimental results indicated that most of the panels' compressive strength was restored by the repairs. Included in their research study

was the influence of manufacturing and design parameters on repair strength. The results of the study also provided a lot of insight into the mechanical performance of repaired sandwich panels under compressive loads that could be used in development of repair designs. Four point bending loads were applied to repaired sandwich by Mahdi and Kinloch et al [35]. The four point bending loads allow easy application of tension and compression loads using the same experimental setup. They found that the applied repair scheme restored a high fraction of the original strength just like Guan et al. Their other important finding was that the loading mode and repair scheme moderately affected the strength restoration.

Zenkert and Burman [36] conducted experiments for determination of the fatigue failure of sandwich composites at constant amplitude cyclic loading. Their experimental results indicated that the facing sheets and core exhibit different stress life (S-N) relationships based on the amplitude of the loading. The face sheets failed earlier than the core at small loading amplitudes and the core fails earlier for large loading amplitudes. Failure results from these experiments are only reported at macro scale levels but no failure is reported at the micro scale. Some of the disadvantages associated with testing are that it is expensive and takes a long time to yield results. It also only validates repair designs that were tested since test results are sensitive to geometric and stiffness deviations to what was tested. Strain and stress profiles in sandwich beams for failure characterisation were determined experimentally in a study by Daniel and Abot [37].

Guan et al [34] studied the compressive behaviour of bonded repaired sandwich laminates that had undergone different levels of damage. Experimental results from their test showed that a single side patch repair can recover 83% of the component's original strength. Keeping in mind that fabrication parameters affect the final mechanical properties of a composite material. They also looked at the influence of fabrication parameters on the performance of the bonded repair. After extensive comparison of test results they concluded that a double sided patch repair, high curing temperature and pressure improved the mechanical properties of a bonded repair. The benefit to the practising engineer is that their result can be used to support bonded repair designs.

Scarf and overlap repair investigations were conducted by Campilho et al [38] to determine stress states, crack propagation and crack growth within the adhesive. The investigation was based on FEA and special interface elements and a cohesive mixed-mode damage model. Geometric parameters of the bonded repairs were varied to determine their influence bond area stress distribution and strength. Their conclusions were that scarf angle patch overlap, patch stiffness and patch thickness all have an influence on bonded repairs.

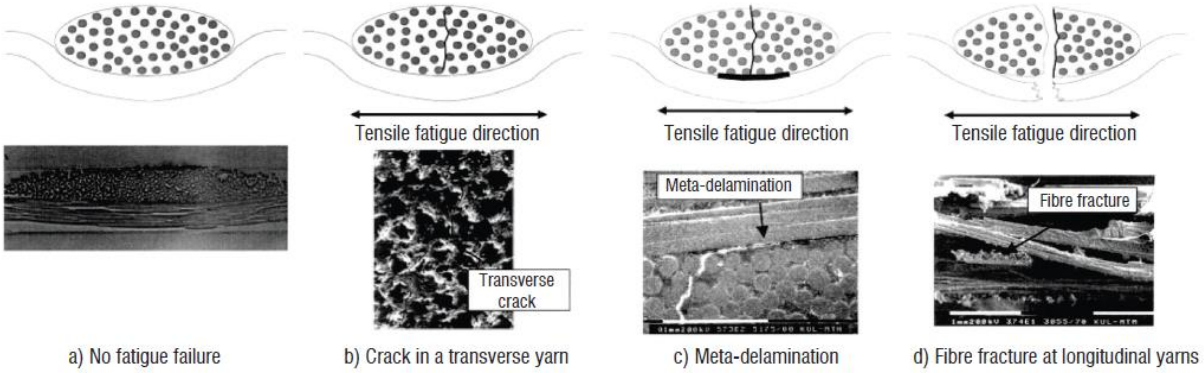


Figure 2.4 Progressive failure characterisation of fibre reinforced composites [39]

An electric microscope and Scanning Electron Microscope (SEM) can be used for detailed fractographic analysis. According to research review by Onera [39] failure progresses through intra-tow cracking, inter-tow cracking and finally tow-rupturing. As shown in Figure 2.4 above.

Typical indicators of resin and carbon fibres failure could be fibre pull-out, fibre rupture chop marks [40] and matrix rolling and flaking [41].

2.4.3 Standard Finite Element Analysis and Damage Modelling

A combination of FEA and cohesive damage modelling was used by Campilho et al [42] to study the behaviour of patch repaired sandwich composite panels. From the study geometric design guidelines were drawn for the overlay and scarf repair configurations. In a different study Kinloch and Mahdi et al [43] used a three dimensional FEA model to predict stiffness and failure of sandwich panels under four point bending. While stiffness prediction was successful failure prediction was not achieved. Their results confirm that a lot still needs to be done in the field of failure prediction.

The National Physical Laboratories manual and other researchers [44]–[48] used FEA to investigate parameters that influence strength, stiffness and life expectancy of bonded composite joints. The investigations focused on geometric, material and fabrication factors in characterising mechanical properties of the joints. While the FEA presented above shows simplified matrix and fibre failures it still does not show details of the state variables at the micro and meso scale, for example, the influence of the fibre-matrix interface strength. The only other alternative is to use MsM to study the details at a micro and meso scale of the composite, hence the idea to conduct this research. The FEA mentioned above relies heavily on empirical parameters for material characterisation which are usually obtained through costly experiments. With minimal dependence on experimental tests, MsM could extend the failure characterisation to the micro scale

In his review of FEA of adhesively bonded composite joints He [49] summarised recent research in terms of their static loading analysis, environmental behaviour, fatigue loading and dynamic characteristics. The review forecasts that bonded joints see increasing applications in the future over bolted joints as a result of their short implementation periods. These diverse research studies are important in developing designs for wider applications. It is also mentioned in the review how FEA helps in lowering development costs and shorting development periods of bonded joints.

The usage of CFRP in highly loaded primary structures of aerospace components has directed a lot of focus on their durability after damage and post repair as reported by Pinto et al [6]. In their evaluation they used three-dimensional FEA and cohesive zone modelling to study the tensile behaviour of bonded joints with a ductile adhesive layer. Repair with, scarf angle and overlapping plies were varied to study their influence on the joint strength. With the results from the study they made a proposal for a design of a bonded joint.

Katnam et al [11] conducted a review that looks at the current scientific challenges and opportunities within the area of structural bonded composite joints. In cases where damage is not extensive repairs are viable since replacement of the entire part is not cost effective in most cases. The review highlights the critical need to develop procedures for bonded repairs to deliver robust, reliable and repeatable bonded repairs in the aerospace industry. This need is motivated by the increasing application of composites in critical aerospace parts. Amongst the advantages of bonded composite joints, over other joining methods, are their load transfer efficiency and better aerodynamic performance.

Wahab [50] conducted a review of 222 journal articles specifically focused on fatigue of bonded joints and their hybrids. In the review the cited articles are divided into four groups according to fatigue analysis, fatigue loading, hybrid joints or nano-adhesive joints. Analysis types included in the review are as follows fatigue strength and lifetime prediction, fatigue crack initiation, fatigue crack propagation and fatigue durability while the loading types are as follows variable amplitude fatigue load, impact fatigue load, thermal fatigue load and torsional fatigue. The reason for conducting a review of this magnitude results from the growing popularity of bonded joints

mainly in the aerospace and transportation industries. Bonded joints offer a number of advantages over classical mechanical fastened joints. They offer high fatigue resistance and longer fatigue lives according to the review. Other advantages are their light weight, ability to join thin and dissimilar parts, good sealing properties and low manufacturing costs and good dynamic properties. As results of their popularity bonded joints fatigue analysis and fatigue strength prediction are critical requirements for design purposes. This requirement is difficult to fulfil because of the complicated nature of fatigue crack propagation and growth, geometry of bonded joints and complex material behaviour of composites.

2.5 Multiscale Modelling and Progressive Failure Analysis

Another new method that is quickly gaining popularity is MsM. MsM has been successfully used for determination of constitutive composite properties at the macro scale by Babero[51]. Current research on carbon fibre reinforced plastics (CFRP) has focused on the application of MsM to study the complex failure characterisation of CFRP based on [15] and [47]. Investigations by Yue [52] have indicated that periodic boundary conditions show better accuracy when compared to displacement and traction boundary conditions. A hexagonal or square fibre arrangement has been validated by Tsai in Strength and Life of Composites [53] as a simplest model that delivers accurate results. With the inclusion of the interface layer between the matrix and the fibre a pull-out failure component can be modelled for some loading cases. For the plain weave architecture MsM has been conducted as covered in several research studies [54]–[56].

A static loading based MsM approach was presented in a recent paper by Couegnat et al [57]. It bridges a micro scale and meso scale models for investigations of progressive failure of ceramic composite materials. Among the damage mechanisms considered are intra-yarns matrix micro-cracking, fibre breaking at the micro scale, inter-yarns matrix cracking, transverse yarns cracking at the mesoscale. This approach has been found to be quite accurate when compared to small industrial structural tests. Another two level MsM approach is used to investigate the stiffness and strength of quadriaxial non-crimp fabric by Cao et al [58]. The model also included the influence of the distortions caused by stitching. Results from this model were in agreement with experimental results. McLendon and Whitcomb [59] developed a meso model of a plain weave textile composite for prediction of progressive damage evolution under inplane static loads. The model used cohesive zone modelling to account for cracks in the matrix, tows and interfaces. Observations from experiments qualitatively matched results from the study and thermally induced stresses were found to cause premature failure in the composite fabric. Various locations of the fabric were highlighted as failure initiation sites for various loading conditions. Nilakantan et al [60], [61] used MsM for an investigation of impact loads on a textile fabric. The approach used a hybrid element analysis of solids and shells and validation was through baseline solid element models. Other MsM studies under static loads were conducted by researchers as reported in [62]–[64].

Various fatigue loading based MsM studies were conducted by Gu et al [65]–[69]. The focus of the studies was the development of a unit cell for a four-step three-dimensional braided composite material. Three-point bending fatigue loads were applied to the unit cell to study its damage accumulation behaviour with increasing fatigue load cycles. Results from the unit cell model closely matched experimental results. The studies also included detailed solid element models of the four-point three-dimensional braided composite at micro, meso and macro scales. Damage morphologies from experimental and simulation observation bear close resemblance.

An MsM based long-term fatigue life prediction study was executed by Zheng et al [70] for a CFRP pressure vessel. In the investigation a combination of micromechanics of failure (MMF) and time-temperature superposition principle (TTSP) was used to predict a CFRP vessel. The fatigue life prediction approach captures the pressure and thermal loads into consideration. The

MMF and TTSP predictions correlate well with experimental results. Although MsM has been used for failure characterisation on composite laminates it has not been used to characterise failure of bonded repaired SCFS.

2.5.1 Homogenization

In the mechanics of composite materials homogenization is the process of averaging mechanical properties of several materials such that they can be treated as one virtual material. Homogenization can be implemented at two levels. The first implementation is applied to fibre and matrix at a micro scale. Homogenized micro scale properties are used in a meso scale model. The second implementation is applied to the meso scale model tows and matrix where the homogenized properties are used in a macro scale model.

Some basic assumptions need to be considered about the composite material before applying homogenization principles. Firstly, the fibre distribution within the tows is assumed to be regular. A square matrix with a fibre in its centre is used for some micro scale models as shown in Figure 2.5. A hexagonal arrangement of fibres is used in detailed homogenisation cases but the square arrangement is mostly used because it simplifies explanation.

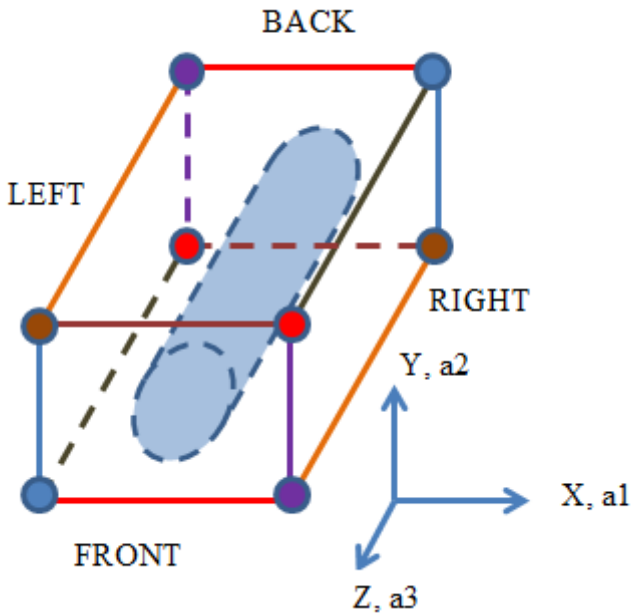


Figure 2.5 RVE colour matching of vertices, edges and surfaces

Periodic boundary conditions on the RVE are applied such that the vertices and edges of similar colours in Figure 2.5 are connected by multi-point constraints (MPCs). The interior nodes of the faces are attached to nodes of opposite faces; front-to-back, right-to-left and top-to-bottom. In addition to providing a rigid connection to the master and slave nodes, the MPCs can also impose initial forced displacements for application of constant strains to the RVE. Refer to Figure 2.5 for schematic diagram of the RVE boundary classification and matching philosophy.

In order to evaluate the overall elastic constitutive matrix ($C_{_equivalent}$) average strain ($E_{_ij}$) based displacements are applied to the RVE. The average strain based displacements are imposed on the boundaries of the meso RVE by means of multi-point-constraints (MPCs) that use the following displacements range equations (Eq. 2.1) from reference [14]. Where the coordinate axes origin is in the centre of the square and $a_{_1}$ $a_{_2}$ and $a_{_3}$ are distances from the centre to the faces in the X, Y and Z direction respectively.

$U_verti_{a_1,x_2,x_3}$ - $U_opp_verti_{a_1,x_2,x_3}$ = 2×a_1×E_xx for $a_2 \leq x_2 \leq a_2$ and $-a_3 \leq x_3 \leq a_3$

$U_edge_{a_1,x_2,x_3}$ - $U_opp_edge_{a_1,x_2,x_3}$ = 2×a_1×E_xx for $-a_2 \leq x_2 \leq a_2$ and $-a_3 \leq x_3 \leq a_3$ (2.1)

$U_face_{a_1,x_2,x_3}$ - $U_opp_face_{a_1,x_2,x_3}$ = 2×a_1×E_xx for $-a_2 \leq x_2 \leq a_2$ and $-a_3 \leq x_3 \leq a_3$

Where U is the contrained displacement, a_1 is the fixed distance to the RVE faces in the Y-Z plane, x_2 and x_3 are variable points between the faces.

A complex state of strain in the RVE results from the applied average strain (E_xx) due to differences in stiffness from the fibre and matrix. The relationship between the average strain and the strain state within the RVE is as follows;

$$E_{xx} = \frac{1}{V}\int E_{ij}dV \tag{2.2}$$

Similarly the relationship between the average stress (S_xx) and the complex stress state within the RVE is also as follows;

$$S_{xx} = \frac{1}{V}\int S_{ij}dV \tag{2.3}$$

If the relationship between the average stress (S_xx) and average strain (E_xx) is as shown in the equation below.

$$S_{xx} = C_{equivalent_xx} \times E_{xx} \tag{2.4}$$

Then a unit strain (E_xx) applied to the RVE should result in a complete column of the constitutive tensor $C_equivalent$.

$$C_{equivalent_xx} = S_{xx} = \frac{1}{V}\int S_{ij}dV \qquad E_{xx} = 1 \tag{2.5}$$

Six average macro strains need to be applied on the RVE to determine the complete homogeneous constitutive tensor ($C_equivalent$). Strain energy within the RVE relates the strains and stresses between the heterogeneous RVE and the equivalent homogeneous RVE. This is true provided the RVEs are under the same uniformly distributed loading and the homogeneous RVE and heterogeneous RVE strain energies are the same.

$$Strain\ Energy = \frac{1}{2}\sum S_{ij} \times E_{ij} \times V_i = \frac{1}{2}S_{xx} \times E_{xx} \times V_{Total} \tag{2.6}$$

Numerically the average stress (S_xx) and equivalent strain (E_xx) are as follows

$$S_{xx} = \frac{1}{V}\sum S_{ij} \times V_i \tag{2.7}$$

$$E_{xx} = \frac{1}{V}\sum E_{ij} \times V_i \tag{2.8}$$

The homogenization process can be implemented in any commercial FEA package of choice. The requirements on the user are a good understanding of Excel macros (Microsoft Software) and the anatomy of the FEA input file. For this study the FEA package of choice is Nastran from the Msc Software suite.

Initially a Patran (.bdf) file is required for manipulation using Excel macros (subroutines). This input file will hold all the geometric, material properties and displacement loads for the RVE. The Excel macros are used to manipulate the displacement loads and generate six different load cases for the RVE. Using the MPCY Nastran card rigid constraints and initial displacements can be imposed on the faces, edges and vertices of the RVE. As explained above the stress results from

the load cases are used to calculate the constitutive matrix ($C_{_equivalent}$). The next step is to calculate the engineering material properties from the components of the constitutive matrix ($C_{_equivalent}$). Engineering material properties that can be determined are Young's moduli, shear moduli, and Poisson's ratios.

The non-linear material property of the multiscale model results from the repeating the homogenisation procedure after the commencement of element failure within the RVE model. This repetition of the homogenization proces results in a slightly different ($C_{_equivalent_i}$) and engineering material properties that are used in the next load increment.

2.5.2 Static Fibre Failure Criterion

The damage within the fibre elements assumes two values 1 and 0.001 since the fibres exhibit brittle failure. Failure initiates in the most loaded elements and spread out to adjacent elements as the loading increases according to the following model:

$$\sigma_{ii} < S_{ii} \text{ then } D_{f-j} = 1 \tag{2.9}$$

But

$$\sigma_{ii} \leq S_{ii} \text{ then } D_{f-j} = 0.001 \tag{2.10}$$

When the applied stress exceeds the fibre strength then the cumulative effect of the failed elements is a reduction in stiffness of the RVE effective mechanical properties as shown in the model below [56].

$$E_{ii}^{j} = E_{ii}^{0}\left(D_{f-j}\right) \tag{2.11}$$

Where: D_{f-j} is the damage indicator for the carbon fibres at the j th load increment.

The resolution of the stiffness changes depends on the mesh density of the RVE. Meaning that coarse RVE meshes exhibit large changes in effective mechanical properties and fine RVE meshes result in small and gradual changes.

2.5.3 Static Matrix Failure Criterion

The matrix failure model is similar to the fibre failure model presented above.

$$\sigma_{ii} < S_{ii} \text{ then } D_{f-j} = 1 \tag{2.12}$$

But

$$\sigma_{ii} \leq S_{ii} \text{ then } D_{f-j} = 0.001 \tag{2.13}$$

When the applied stress exceeds the matrix strength then the cumulative effect of the failed elements is a reduction in stiffness of the RVE effective mechanical properties as shown in the model below:

$$E_{ii}^{j} = E_{ii}^{0}\left(D_{m-j}\right) \tag{2.14}$$

Where: D_{m-j} is the damage indicator for the matrix at the j th load increment.

2.5.4 Fatigue Fibre Failure Criterion

The relationships in Table 2.3 were used to normalised stress and fatigue life in an MsM study in reference [71]. Three different stress - life curves were used since the fibre exhibits different behaviours as the load level changes [71].

Table 2.3 Fatigue properties for the carbon fibres [71]

	Stress Ranges	**S N Curve**
1	$0.85 \leq \frac{S_{max}}{S_{uts}} \leq 0.94$	$\frac{S_{max}}{S_{UTS}} = -0.0506 log N_f + 1.059$
2	$0.75 \leq \frac{S_{max}}{S_{uts}} \leq 0.85$	$\frac{S_{max}}{S_{UTS}} = 1 - 0.310 \left(\frac{log N_f - 4.014}{log N_f - 1.469} \right)^{0.812}$
3	$\frac{S_{max}}{S_{uts}} \leq 0.75$	$log N_f > 6$

2.5.5 Fatigue Matrix Failure Criterion

Tensile and compressive fatigue failure stresses and lives for epoxy resin are presented in Table 2.4. For epoxy both the compressive and tensile fatigue properties are presented since it is easy to make test specimens with pure epoxy compared to pure carbon fibres. Published literature also contains more information on epoxy resin than carbon fibres [72]. In the composite material the epoxy has a lower strength value compared to the carbon fibres hence it requires more characterisation.

Table 2.4 Fatigue properties for the epoxy resin [72]

Cycles to failure	**1**	$\mathbf{10^2}$	$\mathbf{10^3}$	$\mathbf{10^4}$	$\mathbf{10^5}$	$\mathbf{10^6}$
S_{tens}^{matrix} [MPa]	45	35	32	26.5	21	16
S_{comp}^{matrix} [MPa]	70	70	59	47	34	22

In this research study characterisation of a bonded repaired SCFS failure will be implemented within an MsM framework for typical aircraft design load cases. The main benefit of using MsM is that the stress state of the composite constituents can be determined and analysed separately using common continuum mechanics.

CHAPTER 3 - METHODOLOGY

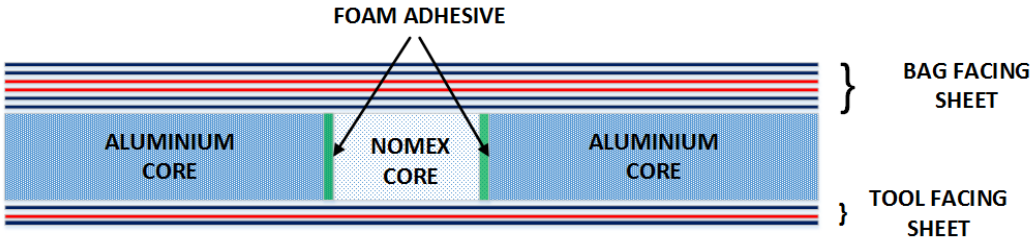
3.1 Introduction

The proposed SCFS repairs were characterized by using both experimental and multiscale numerical analysis. This chapter starts with the experimental procedure followed by the multiscale analysis. Both the static and fatigue tests were conducted and the experimental test results were used to validate the multiscale numerical analysis outcome.

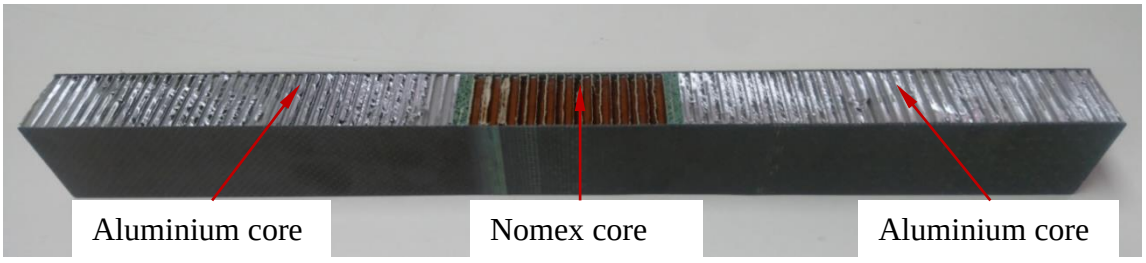
3.2 Experiments

3.2.1 Specimen Details

Figure 3.1 (a) shows a schematic of an aircraft sandwich panel lay-up sequence along with the inner core details. The dimensions of the experimental specimen are 500 mm long, 50 mm wide and 37.4 – 38.2 mm thick. A single specimen consists of aluminium and nomex honeycomb core sandwiched between (BF) bag face and (TF) tool face sheets. The nomex honeycomb was made of hexagonal core (Hexcell N636; 37 kg/m^3) with a cell size 4.8 mm and the aluminium core was made of hexagonal core (Hexcell CRIII-1/8-5052; 130 kg/m^3) with a cell size 3.0 mm. The bag face and tool face sheets were of pre-preg made of Cytec 5276 -1 - toughened epoxy and standard modulus woven carbon fibre tows. The carbon fibres tows were 3K70PW plain woven 42 Grade A with a nominal fibre count of 3000 per tow. The carbon woven fibre consists of warp and weft tows. The warp tows were woven at 90 degrees to the weft tows thus resulting in a mesh of 0/90 degrees. The bag face and tool face sheets were made from the pre-preg and the pre-preg sequence of the bag face sheets was $0^o/0^o/45^o/45^o/0^o/0^o$ and the tool face sheets sequence was of $0^o/45^o/0^o$, respectively. The aluminium and nomex cores were spliced by foam adhesive and the facing sheets were bonded to the cores by film adhesive FM300**.** Figure 3.1 shows the sandwich panel specimen used for the current study.



a) Schematic of the sandwich panel

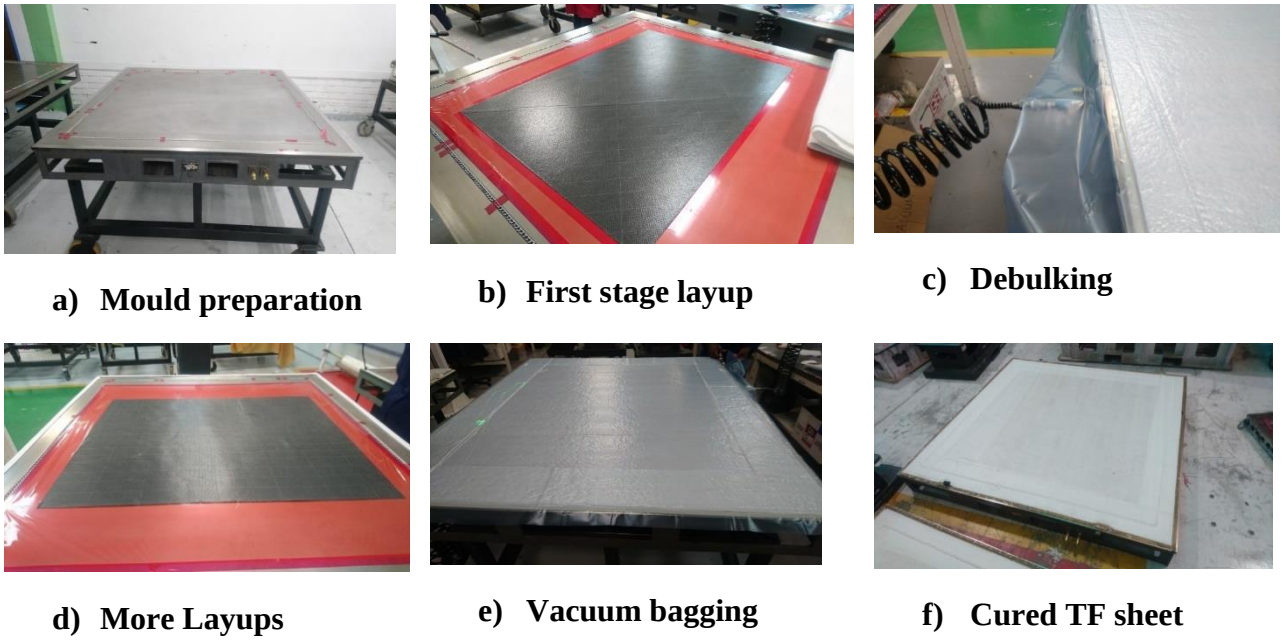


b) Manufactured sandwich specimen

Figure 3.1 Sandwich panel specimen

3.2.2 Tool Facing (TF) Sheet Fabrication

The manufacturing process for the sandwich panels started with the fabrication of the tool side facing sheet. At first, the laminate tool was cleaned to remove grease and foreign objects from its surface and then three pre-pregs of carbon pre-preg were laid on it. The first pre-preg was laid in aligning warp and the mould direction together and the second pre-preg warp direction was rotated to 45^{o} with respect to the mould direction. These two pre-preg were then debulked at a vacuum pressure of 0.2 bar for 5 minutes to remove voids and compact the laminate. After debulking, the last pre-preg was laid in 0^{o} direction and then the laminate was placed in a vacuum bag before moving it from the clean room to the autoclaves for curing. The individual thickness of the pre-preg was about 0.2±0.015 mm which summed to an average thickness of the tool facing sheet of 0.6 mm . Figure 3.2 shows the sequence of the manufacturing process.



a) Mould preparation **b) First stage layup** **c) Debulking**

d) More Layups **e) Vacuum bagging** **f) Cured TF sheet**

Figure 3.2 Manufacturing sequence of TF sheet.

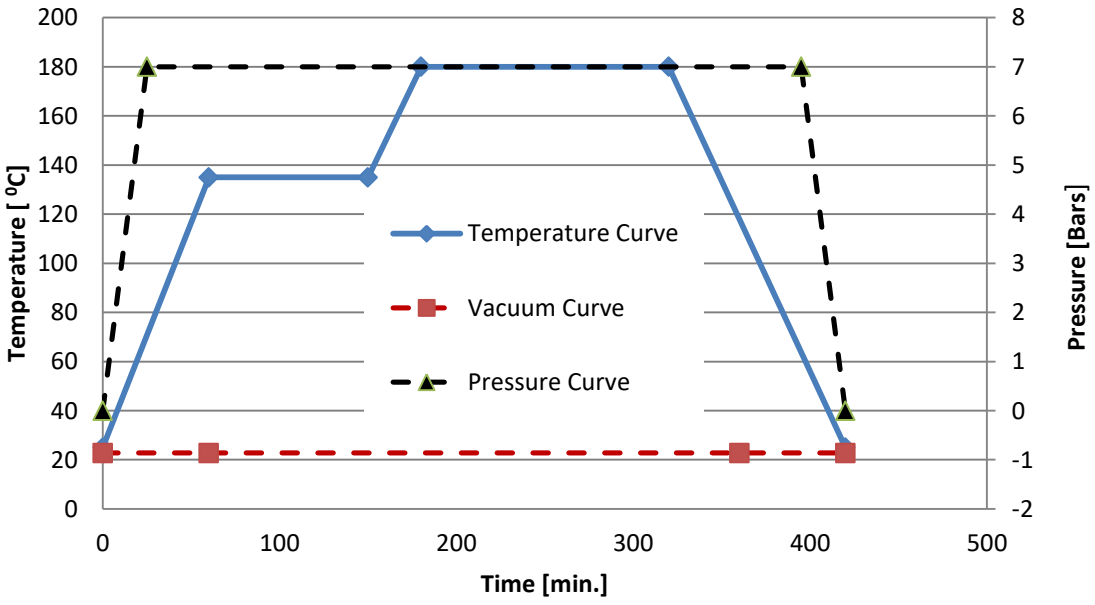
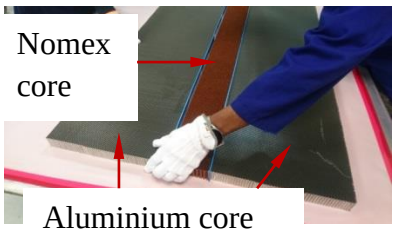


Figure 3.3 Tool Facing sheet cure cycle for the tool side facing sheets

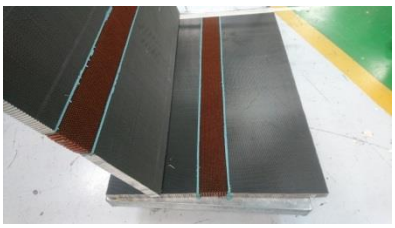
Specific temperature and pressure profiles were applied to consolidate the laminate to strength and durability requirements during the autoclave curing. Two types of pressure were applied in the autoclave. One was a positive pressure applied within the autoclave and the other was a vacuum pressure (Vacuum Pressure) applied to the laminate mould. The temperature profile was applied in two stages; first a dwell temperature of 135 ^{o}C was applied for about 90 minutes and subsequently increased to curing temperature of 180 ^{o}C for about 120 minutes. A graphical representation of the cure cycle is shown in Figure 3.3. After completion of the cure cycle the laminate tool was allowed to cool down before removing it from the autoclave.

3.2.3 Core Layup and Bag Facing (BF) Sheet Fabrication

Before fabricating the bag facing sheet, the sandwich cores needed to be assembled with the tool facing sheet. Two wide aluminium core blocks and a narrow nomex core block were spliced with foam adhesives and then cured for two hours before assembling on top of the tool facing sheet with a film adhesive. This assembly was then debulked and compacted at 0.2 Bar for five minutes. A film adhesive was again placed on top of the assembled core before laying up the first and second pre-preg of the bag facing sheets along 0^{o} direction. The subsequent pre-pregs were laid up along the 45^{o} direction. Each laying up process was followed with debulking to compact the laminate. The final fifth and sixth pre-preg were laid in 0^{o} direction. The finished assembly was then vacuum bagged to complete the pre-curing fabrication process. Figure 3.4 shows the sequence of the assembly process.



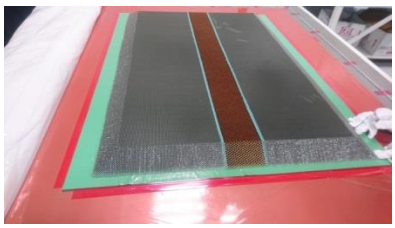
(a) Core Splicing



(b) Spliced cores



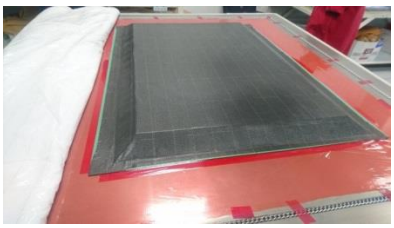
(c) TF facing sheet



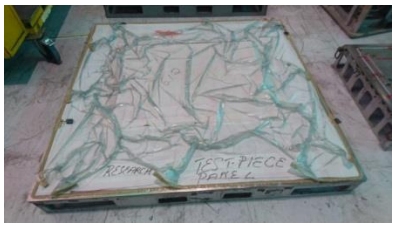
(d) Core layup T sheet



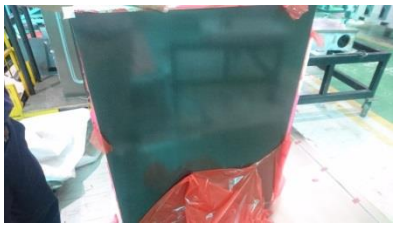
(e) Debulking



(f) BF facing sheet layup



(g) Vacuum bagging



(h) Cured sandwich panel

Figure 3.4 Manufacturing process of core layups and bag facing sheet

The assembly was cured with the maximum applied autoglave pressure of 2.2 bars for 8 hours. Thermocouples were used to measure the overall temperature distribution during the curing process which also helped to avoid laminate distortion due to warping. Figure 3.5 shows the curing temperature and pressure profiles.

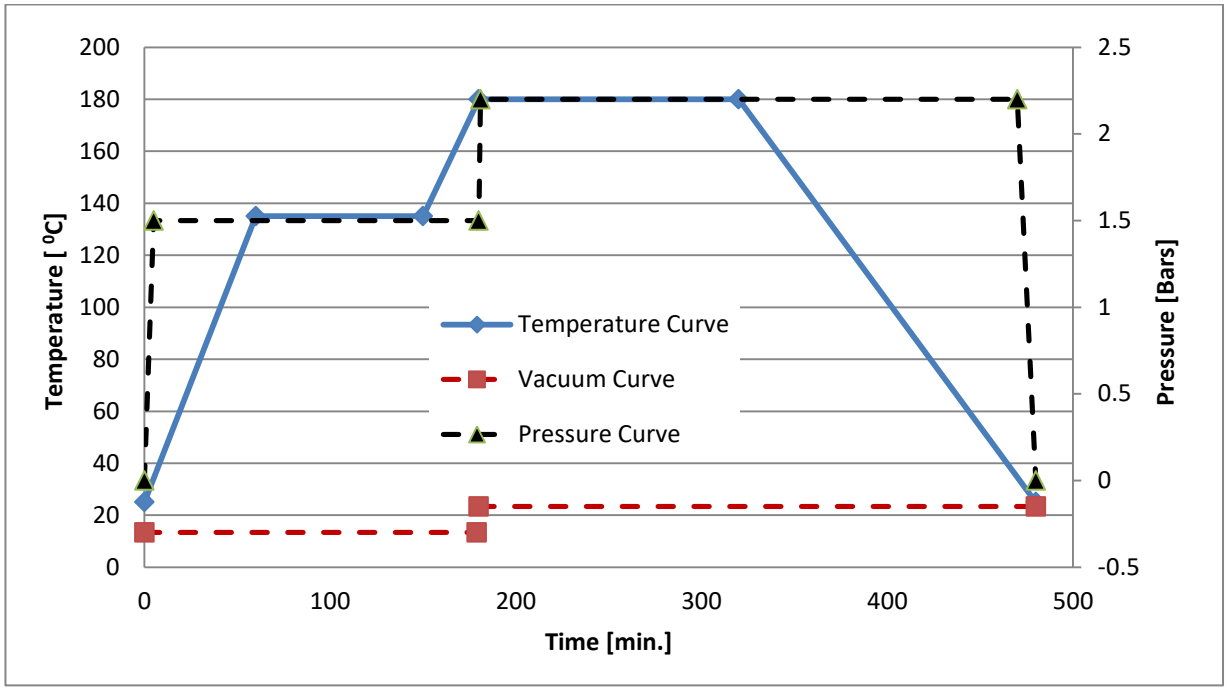
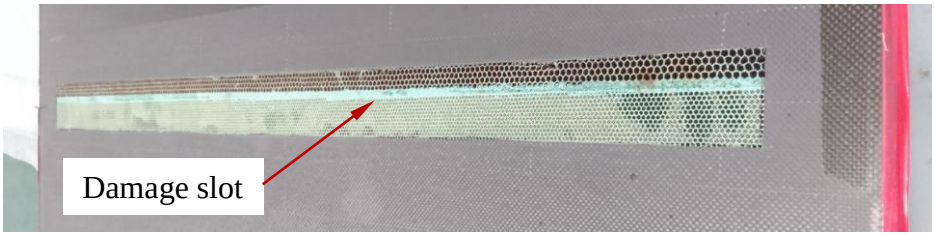


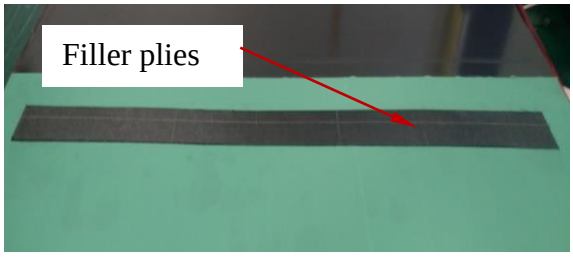
Figure 3.5 Bag Facing sheet cure cycle for the tool side facing sheets

3.2.4 Bonded Repair Implementation

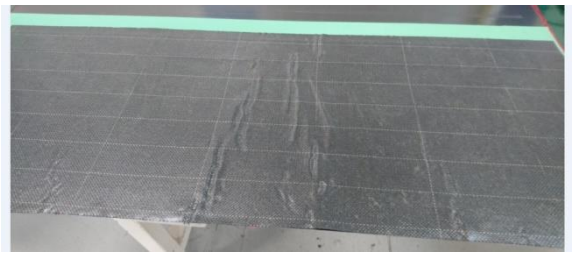
Before implementing the bonded repair, the damge slot was formed on the tool facing sheet. The size of the damage slot was 60 mm wide to the depth of the tool facing sheet without damaging the cores underneath, as indicated in Figure 3.6 (a). The edges of the damaged slot as well as the surface of the tool facing sheet were cleaned with an abrasive scotch bright pad. All dust and debris were removed from the bonding surfaces and were cleaned thoroughly using an isopropyl alcohol cleaning solvent. The next step was to implement the bonded repair scheme. Before starting the repair, a layer of film adhesive (FM 300) was applied inside the damage slot as well as on the tool facing sheet where the repair pre-pregs were applied. The repair scheme consisted of three layers of pre-preg with a laying sequence of $0^0/45^0/0^0$ similar to tool facing sheet layup as filler and repair plies. Thereafter the repair scheme started by placing filler plies inside the damage slot and then the repair plies were laid up ontop of the filler plies covering the cleaned surface of the tool facing sheet, as shown in Figure 3.6 (c). The overlapping of the repair plies reduces the stress concentration by gradually reducing the thickness at the edge of the damage slot, which reduce the likely hood of the premature failure of the repair scheme. The assembly was enclosed in a vacuum bag and cured in accordance with the bag facing sheet cure cycle. Temperature, vessel (autoclave) pressure and vacuum pressure were recorded for the duration of the curing cycle in accordance with normal curing requirements. The cured sandwich panel will be kept ready for cutting specimens. Figure 3.6 shows the sequence of the repair implementation.



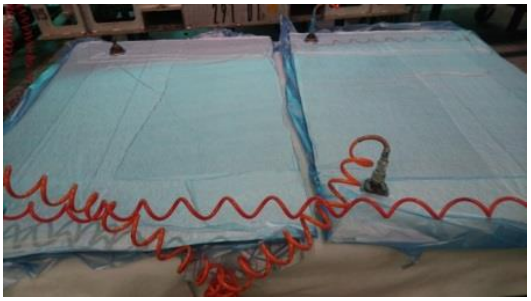
(a) Create damage slot and clean surface



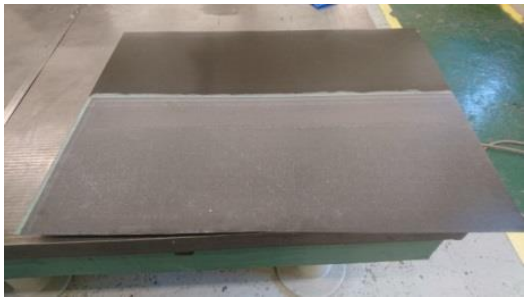
(b) Apply film adhesive and filler plies



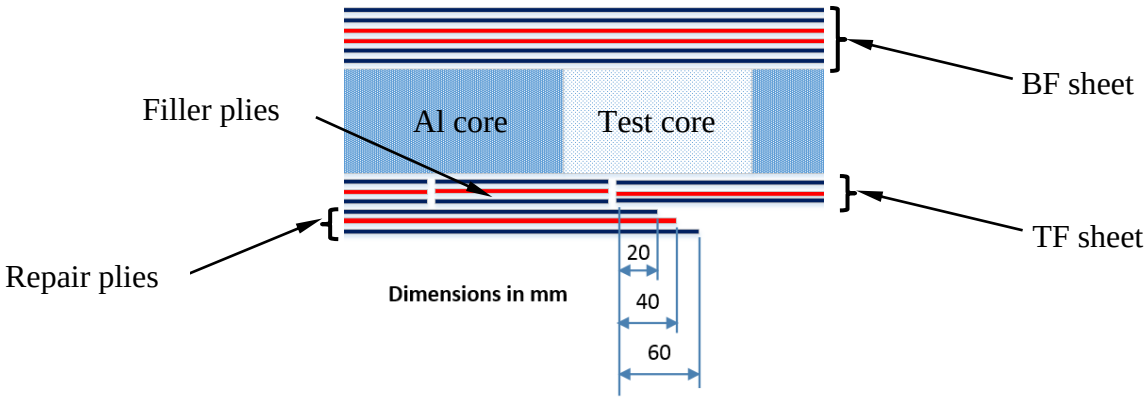
(c) Repair plies application



(d) Vacuum bagging



(e) Cured repair



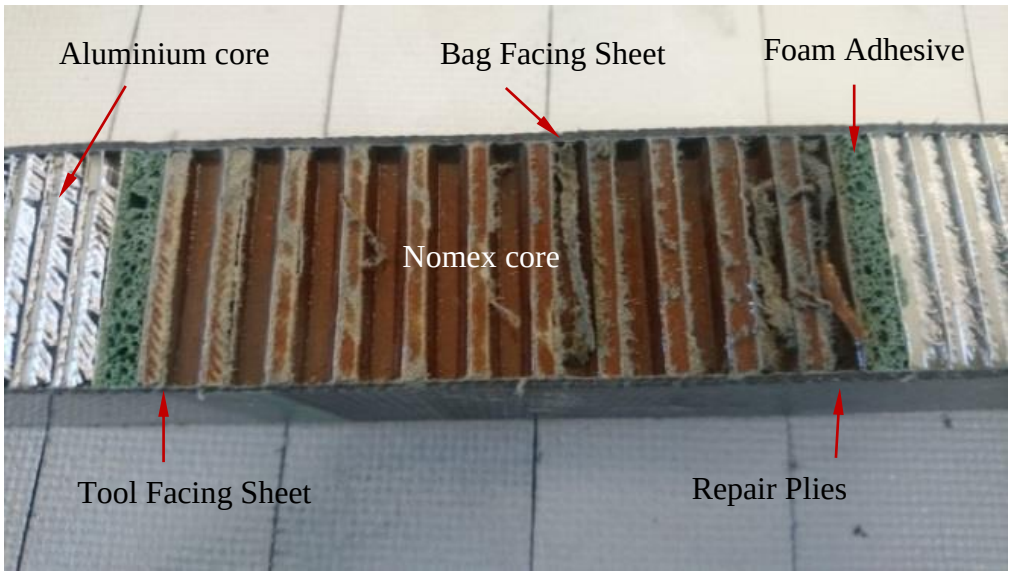
(f) Tool facing sheet filler and repair plies

Figure 3.6 Repair operation sequence

Before the cutting operation, lines were drawn on the white tedlar surface to mark the positions the specimen on the panels as show in Figure 3.7 (a). A fixed table cutting machine was used to cut 50 mm by 500 mm sandwich specimens from the cured repaired panels. In total 32 specimens were produced and these specimens were numbered 1 to 16. The naming convention for the specimens is W93_# and W94_# where # varies from 1 to 16.



(a) Labelling specimens



(b) Specimen test section

Figure 3.7 Cutting tools and specimen making

3.2.5 Static and Fatigue Test Procedures

Before testing the specimens four strain gauges (E1, E2, E3 and E4) were pasted on the surface of the repair plies of three specimens. Three strain gauges were pasted on the repair plies and one was pasted on the tool facing sheet as show in Figure 3.8. Theses strain gauges were used to measure the strains which were then used to validate the elastic modulus and monitor the strain variation over the tapered area of the repair.

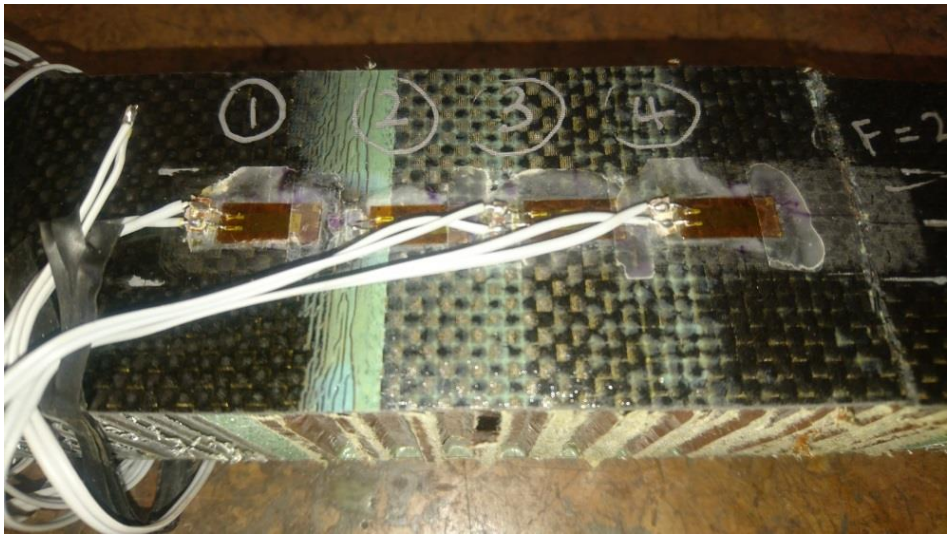
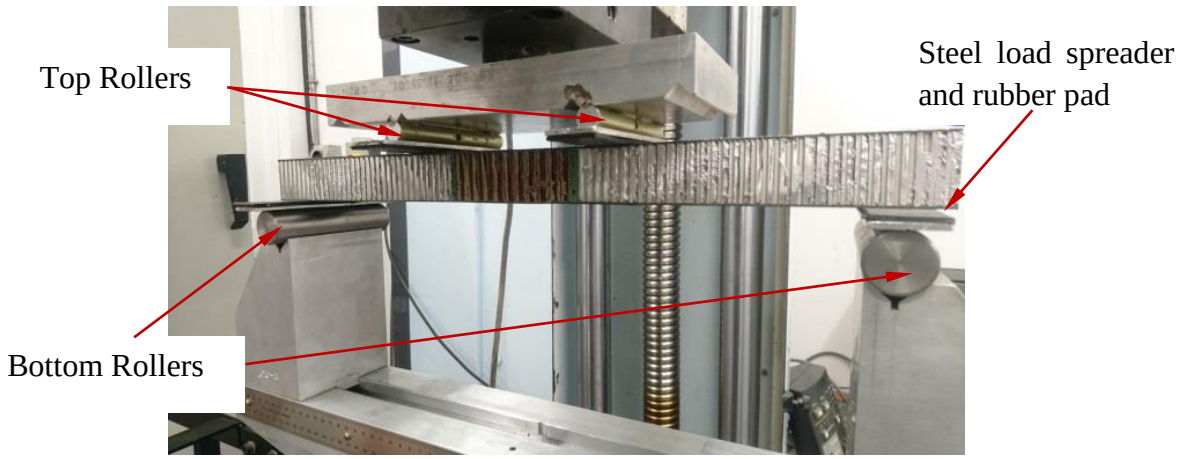
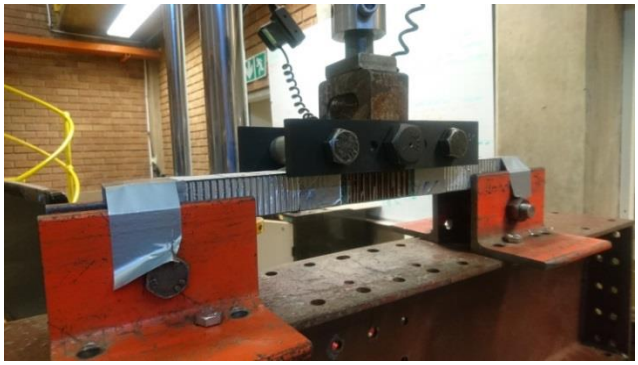


Figure 3.8 Strain gauge positions for E1,E2,E3 and E4 positions over the repair area

The static tests in the form of four point bending test were conducted based on the Airbus AITM 1-0018 testing procedure using an Instron 5500R testing machine fitted with a four point bending setup. All the test parameters and the results were acquired by a 100 kN IX Series automatic materials testing system software. A slotted support beam was fitted to the bottom actuator and it allowed the bottom supporting rollers to slide to any desirable span. A grooved magnetic supporting beam was fitted to the top actuator and the loading rollers fitted into the grooves at the desirable span. Steel plates with rubber pads were placed between the rollers and the specimen as shown in Figure 3.9 (a). These plates were placed to avoid crushing of the core at the loading and support points. Loading was applied according the test specification at a rate of 12mm/min at a room temperature range of 20 – 25 ^{0}C. For the static tests, the specimens were separated into two groups. One group of specimens was loaded with a load in the range of 4.1 to 8.6 kN. The test was stopped before the specimens failed. Results from these tests were used to characterise the progressive failure of repaired facing sheet and to obtain the elastic modulus of the tool facing sheet. This test detail is given in Table 3.1. The other group of specimens was loaded untill failure and the failure loads recorded after the experiment. Failed specimens were used to characterise the tool facing sheet failure at the fracture points and the test details are as presented in Table 3.2. The surfaces of the tested group one specimens and fractured surfaces of the group two specimens were characterised using hand held microscope and scanning electron microscope.



(a) Specimen positioning with pads at loading points for the static test



(b) Specimen positioning with pads at loading points for the fatigue test

Figure 3.9 Static and fatigue testing machines and four-point bending setup

Table 3.1 Loads for static specimens

Specimen ID	Average Specimen Width [mm]	Average Specimen Hieght [mm]	Applied Load [KN]
W94_4	50.1	36.87	4.1
W93_11	50.2	36.7	5.1
W94_5	50.0	36.87	6.1
W93_6	50.0	36.72	7.1
W94_3	50.0	36.5	7.6
W93_14	50.3	36.72	8.1
W94_9	50.3	36.83	8.6

Table 3.2 Static specimens for loading to failure

Specimen ID	Average Specimen Width [mm]	Average Specimen Hieght [mm]
W93_10	49.8	36.8
W94_2	49.8	36.8
W94_6	49.9	36.9
W93_16	49.1	36.6
W94_11	49.8	36.92
W94_14	49.8	36.7
W93_15	49.7	37.1

The Fatigue testing was in the form of bending fatigue, the setup followed Airbus AITM 1-0018 and the testing was carried out using Instron servo-hydraulic testing machine Model 1342. The specimen was mounted on a fixed bottom actuator on an I-beam. Fatigue loads were applied through the top actuator and the tests were run under load control. The fatigue loads were in the form of a sine wave (Figure 3.10) with a maximum load (P_max) in the range of 60% to 90% of 8.5 KN, static load. Moreover, as explained in the literature, carbon fibre reinforced composite exhibit finite fatigue lifes at the maximum loads above 60% of the static failure loads. Thus the fatigue mean loads we arbitrarily selected within the range of 60% to 90% of the static failure loads that is i.e of 3.41 KN to 4.72 KN. The fatigue experiment started with a mean fatigue load of 4.7 KN and the subsequent fatigue loads were reduced until 3.41 KN. A total of thirteen

specimens were tested. The loading was a tensile unidirectional fatigue load with a stress ratio, R of 0.1 and was cycled at 2 to 3 Hz. Figure 3.9 (b) shows the fatigue test setups. Table 3.3 shows the applied fatigue loads and the specimen geometric properties. The relationship between the mean load and amplitude is as follows.

$$P_{max} = P_{amp} + P_{mean} \tag{3.1}$$

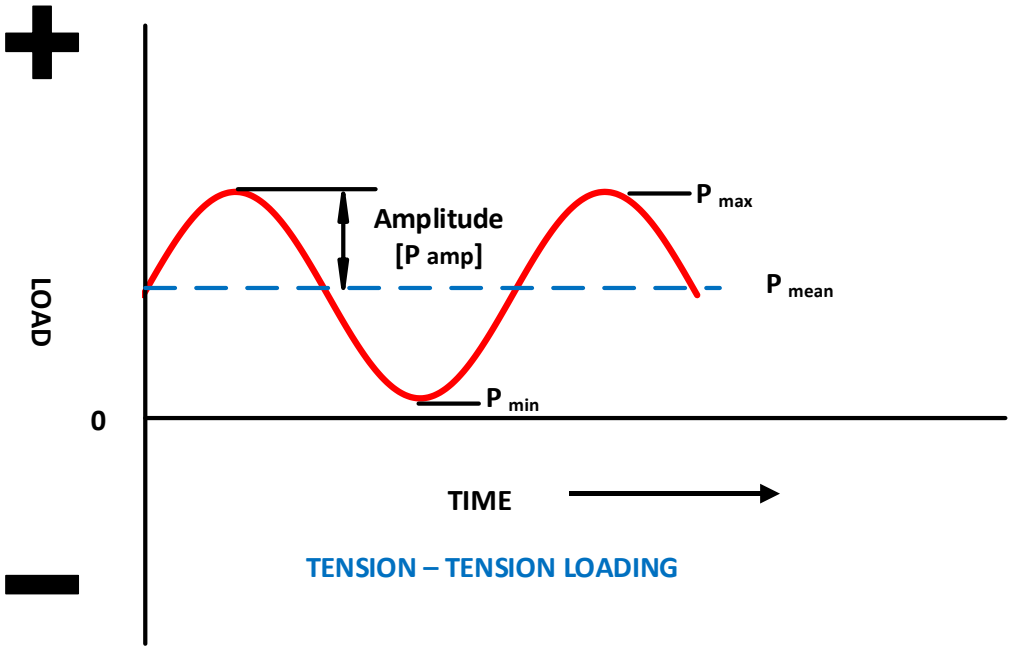


Figure 3.10 Sine wave loading applied to fatigue specimens

Table 3.3 Applied fatigue loads for specimens

SPECIMEN ID	Average Specimen Width [mm]	Average Specimen Shickness [mm]	Load Amplitude [KN]	Mean Load [KN]	Loading Frequency [Hz]
W93_13	50.4	36.6	3.60	4.39	3
W93_5	49.9	36.8	2.92	3.57	2
W93_7	49.8	36.8	3.78	4.61	3
W93_2	50.2	36.6	3.23	3.95	3
W93_3	50.2	36.8	3.14	3.84	2
W94_15	49.7	36.9	3.56	4.35	3
W94_16	50.1	37.0	3.33	4.06	2
W94_8	49.8	37.0	4.05	4.94	3
W94_7	50.2	36.9	3.74	4.57	

The fatigue bending moment load was applied to the sandwich specimens as the sandwich panel was designed to carry bending moments created by distributed loads like air pressure. Applying the maximum bending moment to the test section of the four point bending setup depends largely on the relative position of the top and bottom rollers presented schematically in Figure 3.11. For

maximising the applied pure bending moment, the distance between the bottom and top roller had to be maximised. Since the bottom rollers were positioned at their extreme locations, i.e., 25 mm from the beam ends according to AITM 1-0018, the top rollers are placed as close to each other as possible without loading the softer nomex core directly. Loading the soft nomex core with concentrated loads from the rollers could dent the facing sheets. Denting the facing sheets could causes premature failure especially when the facing sheets are loaded in compression. This problem was also partially avoided by spreading the load with the rubber pads under the load spreaders. Another step taken to avoid denting the tool facing sheet and crushing the core was using stronger aluminium hexagonal core in areas where the rollers make contact with the specimen.

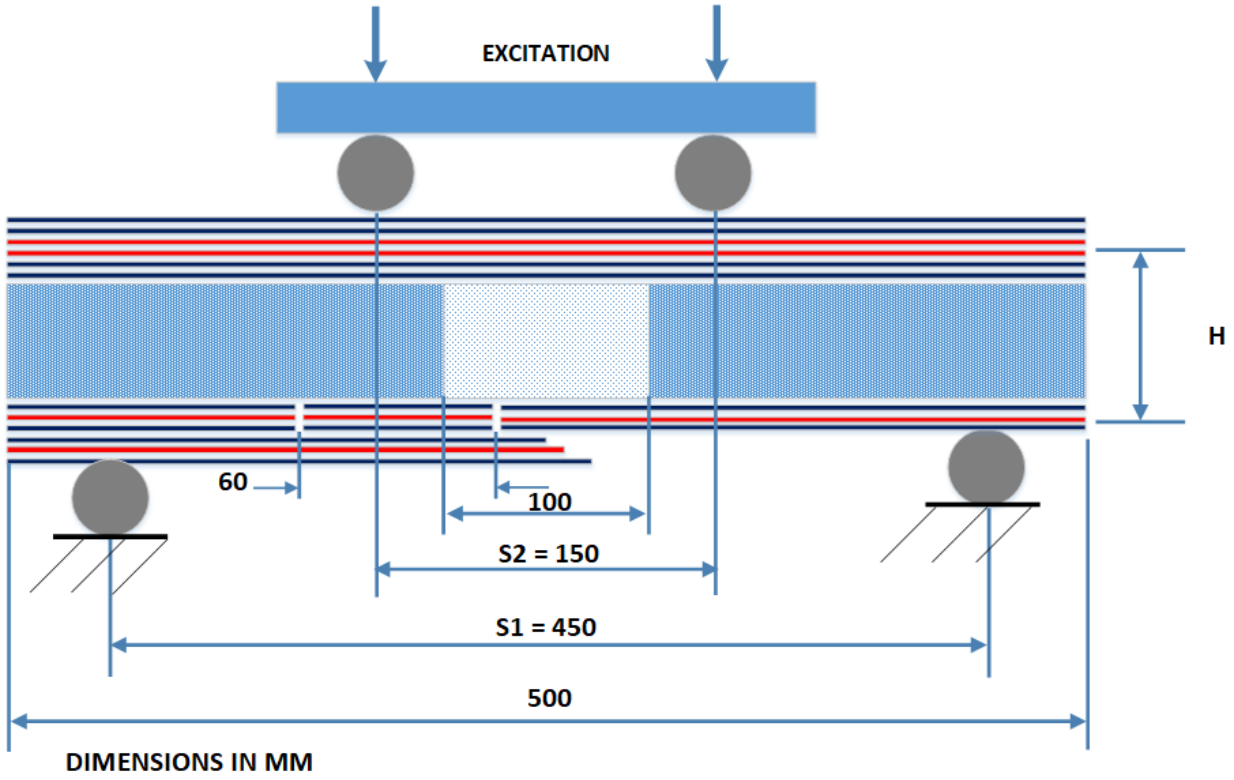
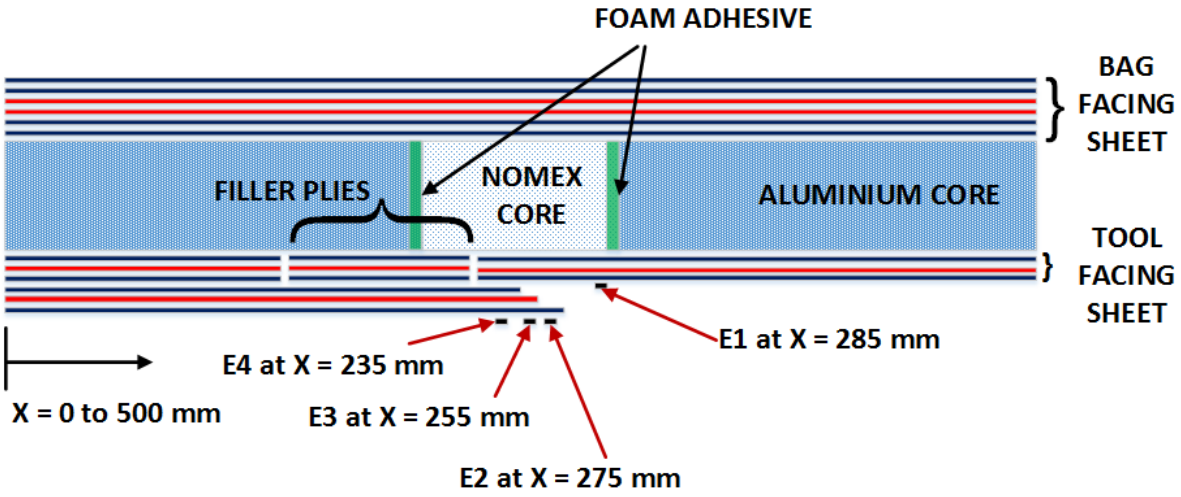
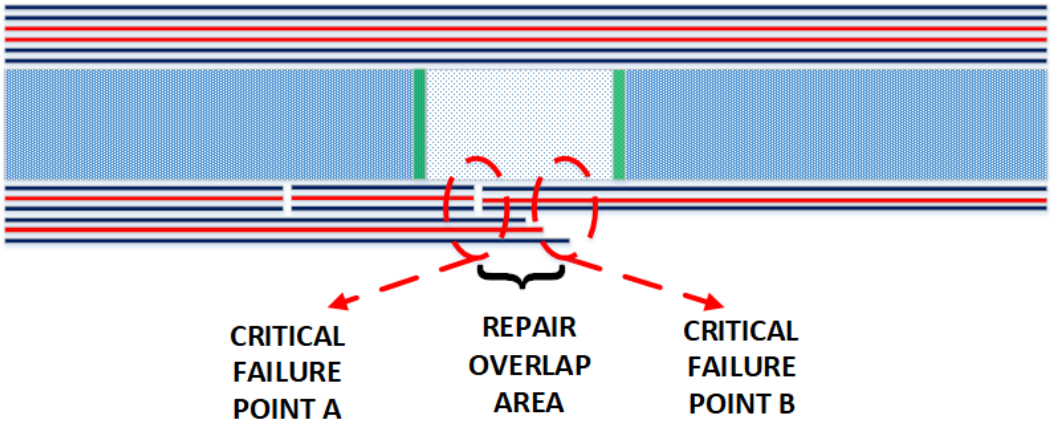


Figure 3.11 Roller positions on the specimen

There were two critical points of failure for both static and fatigue loadings conditions. Both points were critical since the repaired facing sheet thickness was at its lowest at three plies thick and the bending moment was at its maximum at these points. The combination of lowest thickness and highest bending moment promoted failure at these points. In order to analyse the failure behaviour at these failure points, four strain gauges were used to measure the strain on the repaired facing sheet as explained before. The position of the four strain gauges were in such a way that E1 was bonded to the critical point B, E2 to E3 were bonded to the tapered area of the repaired facing sheet and E4 was bonded close to critical point A as shown in Figure 3.12 (a). The results of the strain reading analysis will be used to characterise the two critical failure points indicated in Figure 3.12 (b). The strain readings were used to calculate the bending tensile stress on the tool facing sheet. These stresses were calculated using the following eqautions.



(a) Strain gauge positions on the tapered area of the repaired tool facing sheet



(b) Critical failure points on the repaired tool facing sheet

Figure 3.12 Strain gauge positions and critical points for the repaired facing sheet

3.2.6 Tool facing sheet stress calculation

The tool side repaired facing sheet [0/90, ±45, 0/90] modulus was obtained from strains and stresses at position E1. A linear regression line fitted to the strain-stress graph can also give the laminate modulus as its gradient. Strains are obtained from experimental measurements and the stresses were calculated as follows. The stress at positions E1:

$$\sigma_{facing\ stress} = \frac{F \times (S2 - S1)}{4 \times H} \times \frac{1}{w \times t} \tag{3.2}$$

For F = 1003 N the stress [MPa] is:

Where:

S1 = support span [mm]

S2 = loading span [mm]

F = Total applied load [N]

t = tool side laminate thickness [mm]

H = couple loads offset [mm]

w = specimen width

σ = facing sheet stress

3.2.7 Fractographical characterisation

The fractographic analysis were used to capture the failure behaviour exhibited by the matrix and the fibres at the meso and micro scale level using scanning Electron Microscope (SEM) and an optical hand held microscope. Observations were made with a CE ROHS digital microscope with a magnification range of 50 to 500. The digital microscope's size made it easy to point it at the hard to reach areas of the specimen hence its preference over conventional microscopes. The tested specimens were analysed to identifiy the fracture and failure behaviour. The results were compared with the MsM numerical results.

3.3 Multiscale Modelling (MsM) and Analysis Procedure

3.3.1 Introduction

The schematic of the MsM for the three length scales is shown in Figure 3.13. This diagram summarises the procedure that was involved in a multiscale modelling framework. The modelling process started at the micro scale level by defining micro RVE models which were then loaded with fictitious small strains within the elastic limit. The results from this modelling process were used to calculate micro RVE homogenised material properties which were then forwarded to the meso scale level to be used in the meso RVE model. The meso scale model consisted of weft and warp tows impregnated in a matrix and the homogenised micro RVE material properties were used as the material properties of the weft and warp tows. Once more fictitious strains within the elastic limit were applied to the meso RVE models to obtain the homogenised meso RVE material properties. The homogenised meso RVE material properties were used in the macro model. The macro scale model consisted of composite plies (carbon fibre and epoxy) and the properties of the homogenised meso RVE model were used for the composite plies.

At this stage, the macro model was ready for application of the first load increment. The first load increment load was 10% of the experimental failure load. This load was applied to the macro model such that the locations of the critical strain points were obtained. The macro model with the first load increment was solved using Nastran solver and critical ply element strains were extracted from critical points A and B. These macro element strains were applied to the meso scale model as multipoint constraints (MPC) displacements. If there was no failure in the meso scale model, the critical strains were extracted from the warp tows at critical points A and B and were applied to the micro RVE model. Now the micro RVE was checked for any failed elements. If there was no failed elements in the micro RVE model then the second load increment of 20 % was applied to the macro model to obtain the updated critical strains. This cyclic process is called the progressive failure analysis and the process continues until failure occurs in elements within the models.

If suppose the micro RVE model elements failed due to the applied strains then the same fictitious strains were applied to the micro RVE model with failed elements to obtain the update homogenised material properties. These properties were used in the meso RVE model warp and weft tow elements.

If suppose the applied strains initiated elements failure in the meso RVE model then the new meso RVE material properties were recalculated. This was achived by appliying the same fictitious strain to the meso RVE model with failed elements to obtain the update homogenised material properties. These properties were used in the macro model before moving to the next load increment. Subsequent macro load increment analysis followed the same analysis procedure and VBA scripts were used to automate the procedure.

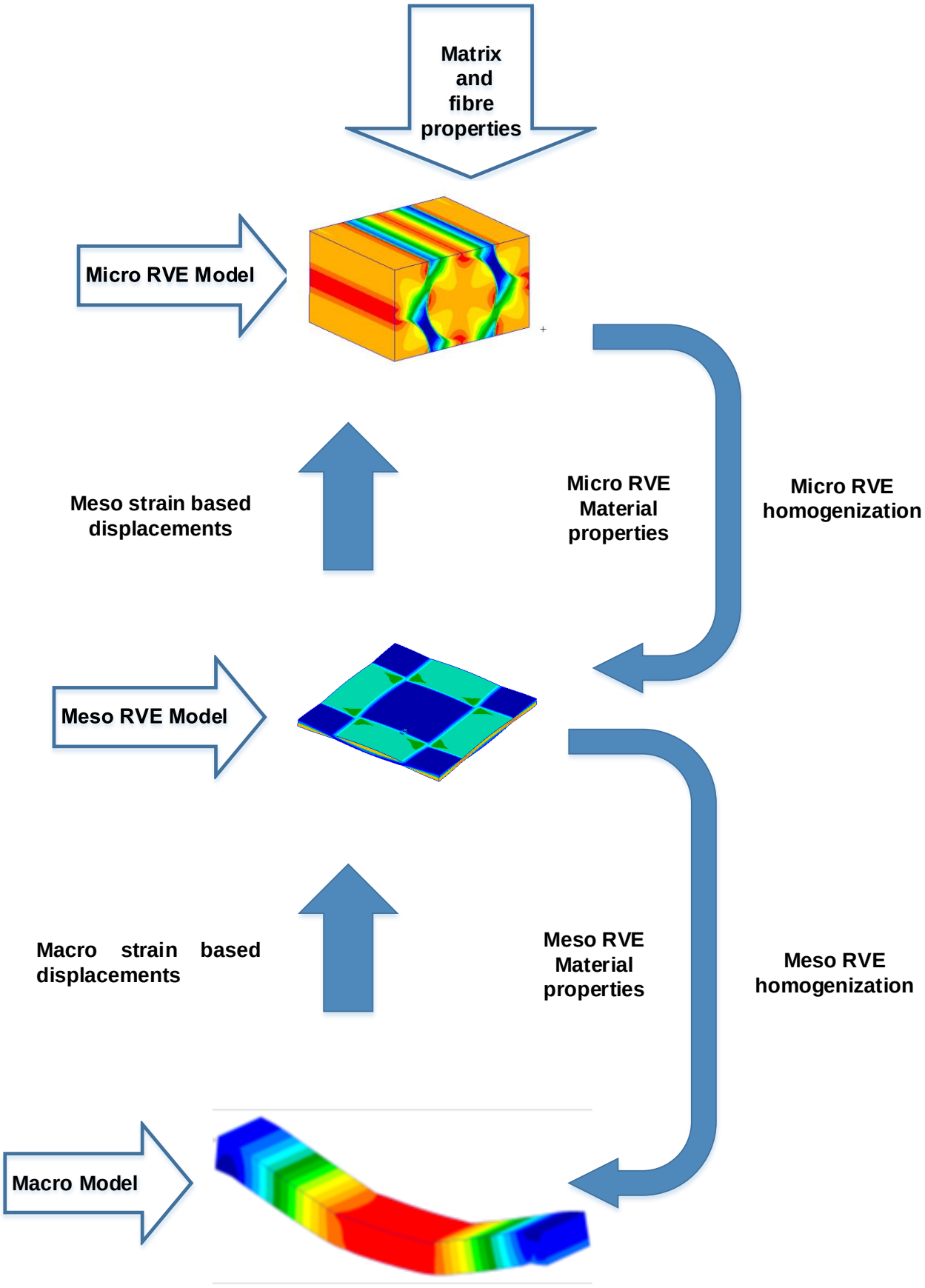


Figure 3.13 Multiscale modelling strategy framework

The MsM analysis was carried out in a Toshiba laptop with an Intel (R) Core (TM) 7i-2640M central processing unit,with a processing speed of 2.8 GHz and 8.0 gigabytes of RAM. The following subsections explain the modelling process of the micro RVE model, meso RVE model and macro model. The original input text files for the three models were created in MSc-Patran 2014 preprocessor. Microsoft Excel Visual Basic for Applications (VBA) was used to edit and update the input text files prior to solving through MSc Nastran 2014 solver. The VBA scripts were used to execute the following tasks during the analysis.

1. Adding increment loads at all levels of the MsM analysis.
2. Implementing the element material properties degradation during progressive failure analysis by referencing the appropriate material property identification numbers as the MsM elements failed.
3. Implementing the failure criteria based on the constituents' material tensile strengths and the resulting stresses.

3.3.2 Input File Structure

It is necessary to understand the Nastran input file structure before developing the models. Figure 3.14 shows the structure of the Patran\Nastran input file. The main sections of the input file are Nastran statement, File Management Statements, Executive Control Statements, Case Control Commands and Bulk Data Entries. Nastran Statement section is used to control system default settings and is not needed in most input files. Another section that is not compulsory is the File Management Statement which is used to allocate files, control restarts and database operations. Case Control Commands request outputs, select bulk data items such as loadings and constraints. Bulk Data Entries define the model geometry, materials, loadings and boundary constraints. The input file also includes the CEND, BEGIN BULK and ENDDATA delimiters. CEND marks the end of the Executive Control Statements and beginning of the Case Control Commands. BEGIN BULK indicates the end of the Case Control Commands and the beginning of the beginning of the Bulk Data Entries. ENDDATA is the last statement in the input file. The "Nastran Statement" and "File Management Statement" were not included in the input file since they are optional.

Input File Sections/Delimiters	Importance
Nastran Statement	Optional
File Management Statements	Optional
Executive Control Statements	Required Section
CEND	Required Delimiter
Case Control Commands	Required Section
BEGIN BULK	Reqiured
Bulk Data Entries	Required Section
ENDDATA	Required

Figure 3.14 Patran\Nastran input file structure

Figure 3.15 shows the first part of the micro RVE model input file. The Executive Control Statement "SOL 101" defines a linear solution for the solution type. The Case Control statements "MPC=4" and "SPC=2" were used to select multipoint constraints used in the analysis. The other statements were used to request forces and element stresses after the analysis.

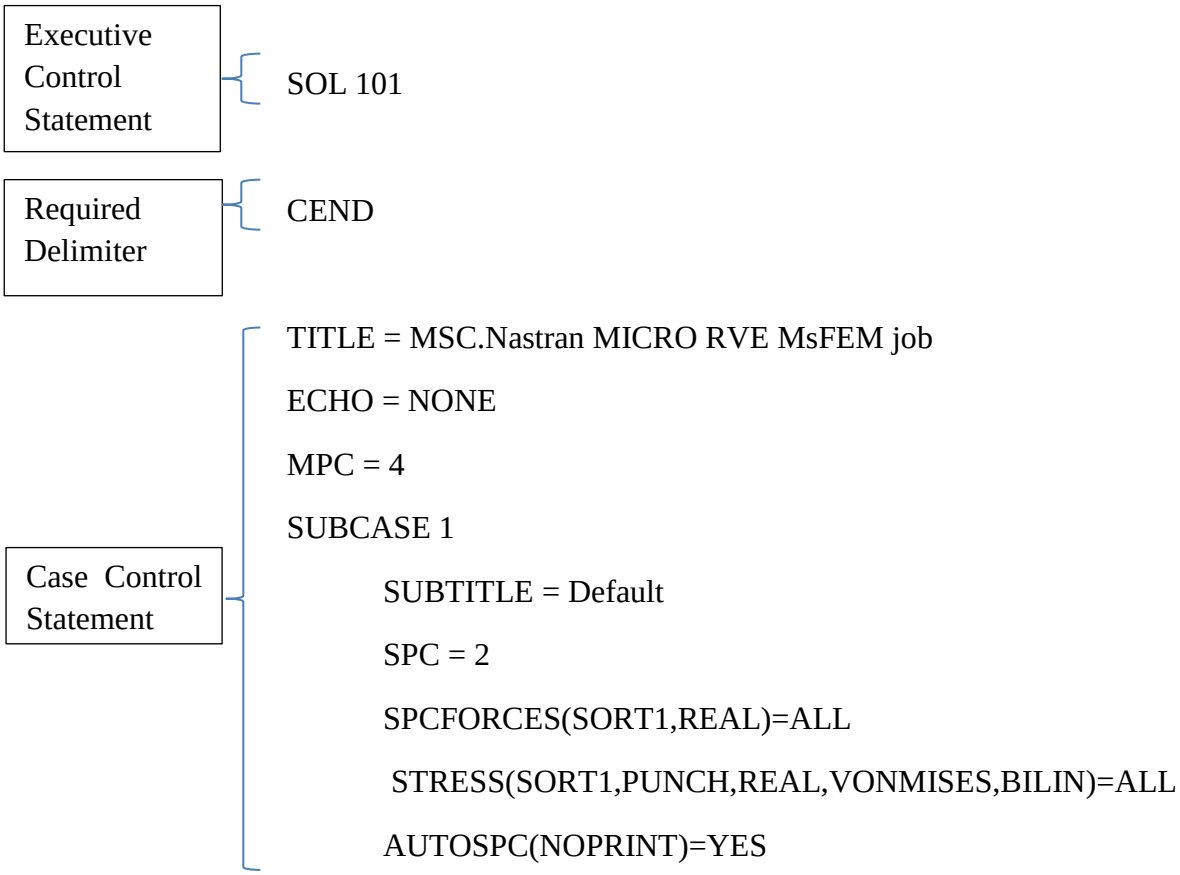


Figure 3.15 Micro model input file sample

Some sample bulk data entries of the middile of the input file are shown in Figure 3.16. It starts with an entry of the carbon fibre solid element material property card "PSOLID,1,1,0". Where the first numerical is the propertiy ID number and the second numerical is the material ID number. The material properties used card "MAT1,1,250,,.3". MAT1 is an isotropic material with ID number 1, elastic modulus 250 GPa and poisson's ratio 0.3. The carbon fibres were modelled using solid hexagonal elements in card "CHEXA, 10501,1,… ". Where the first numerical is the element ID number which selected material 1 in the second numerical. The rest of the numbers in the card are the eight nodes at the vertices of the solid element. The matrix element entry is not included in the sample but it uses the same cards as the solid carbon fibre elements. The MPC group was defined using card "MPCADD,4,1,2,3,4,5,6,7" where the first numerical is the ID number also referenced in the Case Control section. Next was the list of all the MPC node pairs and only the seven are show in the Figure 3.16. A sample "MPCY,1,23769,1,1.0,2.4 ,,17542,1,-1.0" card is included which has ID number 1 in the first numerical. Numericals 23769 and 17542 are the node numbers on which the MPC boundary conditions were applied.

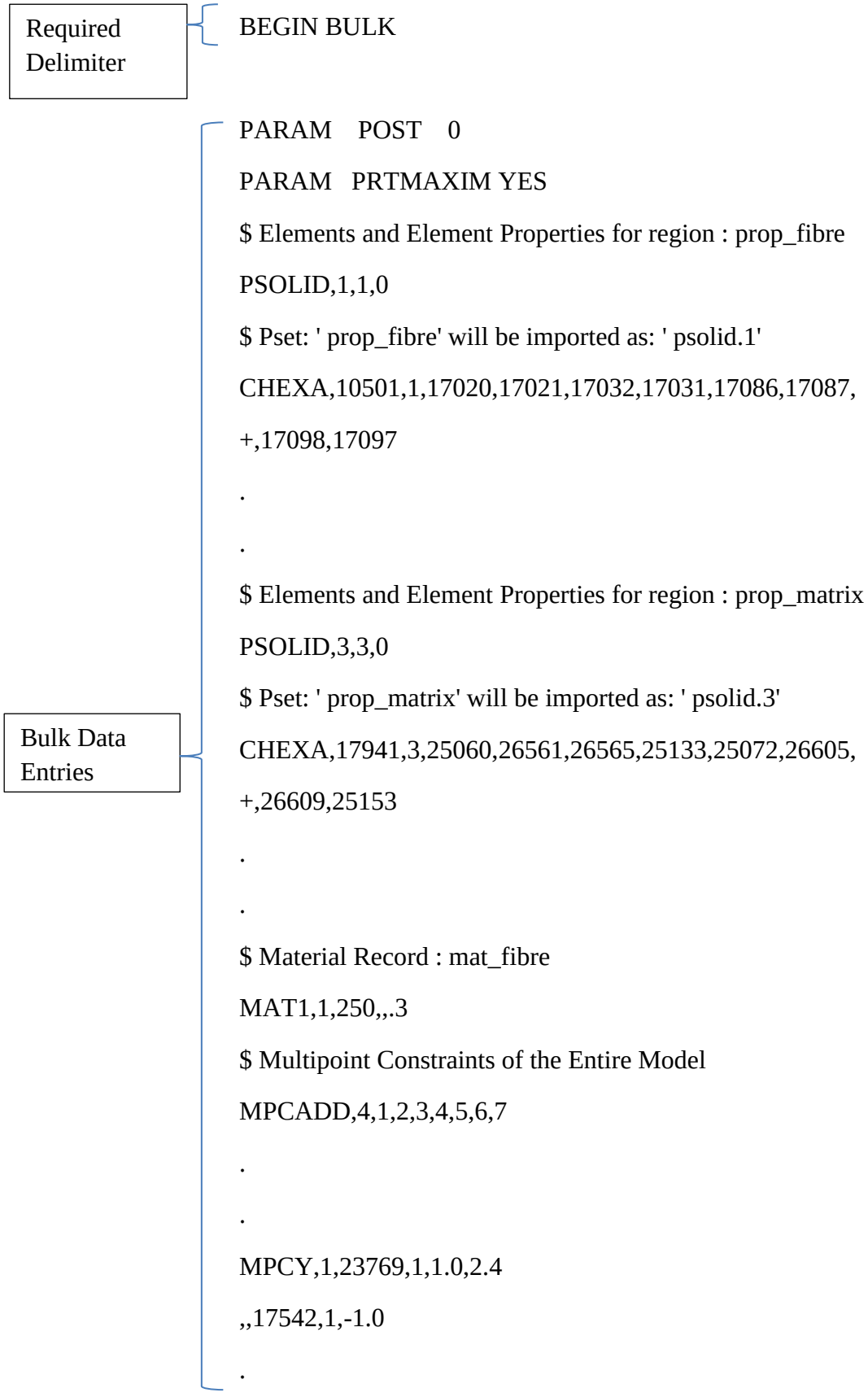


Figure 3.16 Micro model input file sample Bulk Data Entries

The last part of the input file is shown in Figure 3.17. It defined the grid points using card "GRID,17020,,5,22.4,38.8 where the first numerical is the ID number and the rest of the numbers are the nodal coordinates. The constraints group was defined in card "SPCADD,2,1" and referenced in the Case Control section. The first numerical is the ID number and the list of contraints is 1. Lastly the SPC boundary conditions were defined using card "SPC1,1,123,17218" where all the translations of node 17218 were fixed.

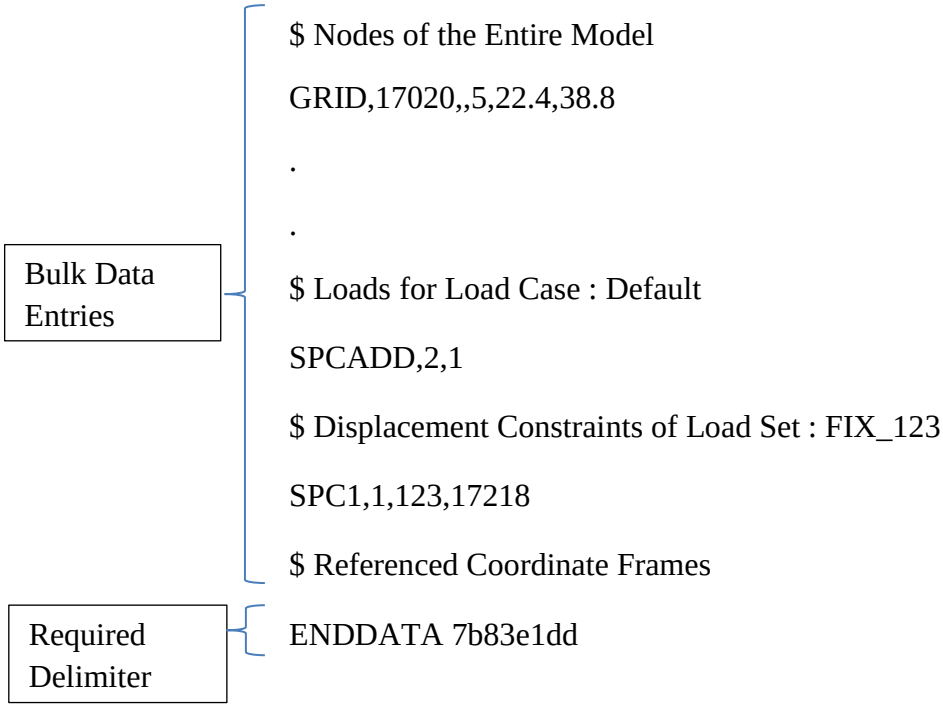
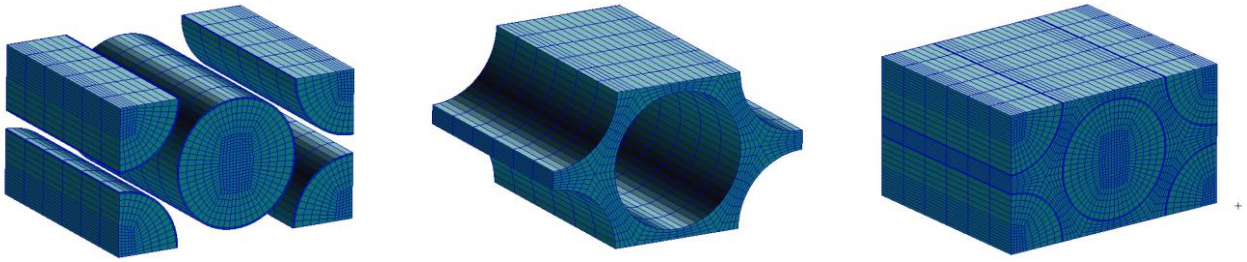


Figure 3.17 Micro model input file sample bulk entries continued

3.3.3 Micro Model

The micro RVE model was defined using the input file discussed above. The micro RVE model contains fibre and matrix phases which were used to determine the material properties of the homogenised micro RVE model. The homogenised micro RVE material properties were used as the weft and warp tows element properties in the meso model. The fibre diameter of the micro model was fixed at 7.2 µm as per the manufacturer specifications. Based on the volume fraction (55 % fibre volume in a cured ply) and the fibre diameter, the RVE dimensions were calculated for a hexagonal arrangement of fibres. Four quarter fibre segments and one complete fibre were joined together to form a hexagonal arrangement of fibres. The matrix was inserted between the fibres and glued together the micro RVE composite model. Figure 3.18 (a) shows the fibre FE model i.e four quarters and one fibre model and Figure 3.18 (b) shows the matrix elements that filled the gaps between the fibre elements to make up the complete micro RVE model, which is shown in Figure 3.18 (c). The materials properties for the model were taken from epoxy (5276-1) and carbon fibre (HTS40 E13 3K) respectively as given in Table 3.4. Both the fibre and matrix were modelled using CHEXA solid elements. Only a quarter of the model was created and the rest of the model was completed by mirror images in two planes parallel to the fibre axial direction. The matrix and fibre materials properties were isotropic as given in Table 3.5.



a) Fibre Model b) Matrix model c) Micro RVE

Figure 3.18 Micro model

Table 3.4 Micro RVE geometric properties

Micro RVE Details	Values (μm)
RVE Width (w)	24
RVE Thickness (t)	16.4
RVE Length (l)	28.4
Fibre Diameter (df)	7.2

Table 3.5 Material properties for Carbon Fibre (HTS40 E13 3K) and Matrix (Epoxy 5276-1)

	Elastic Modulus (GPa)	Shear Modulus (GPa)	Tensile Strength (GPa)
	E11, E22, E33	E12, E13, E23	S11, S22, S33
Carbon fibre	250	---	3.85
Matrix	4	1.54	0.045

The analysis sequence of the micro RVE is schematically shown in Figure 3.19. The first step was to create a 3-D micro RVE model and to generate the Patran input text file. The Patran input text file was generated without the boundary conditions.

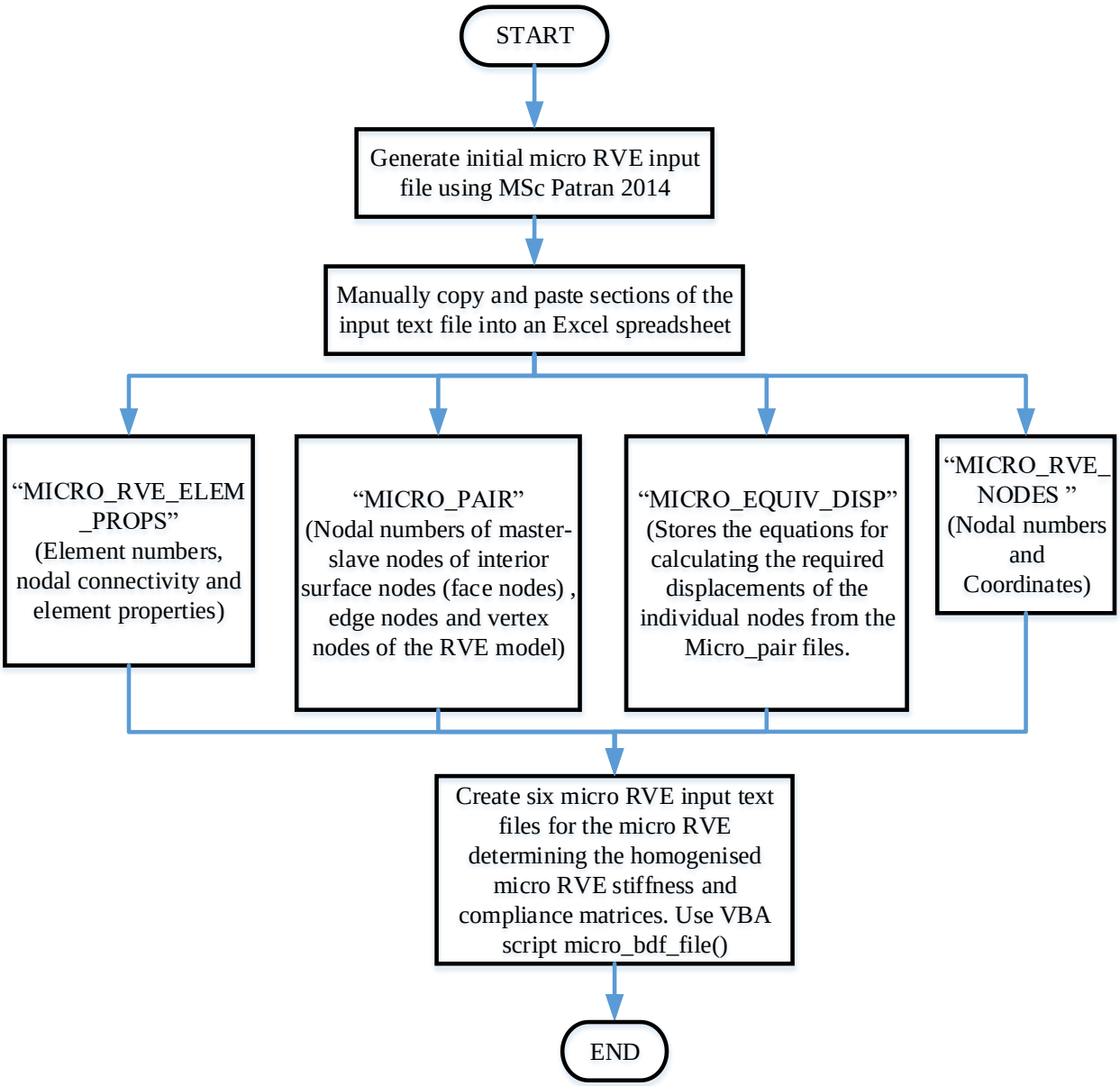


Figure 3.19 Generating the micro RVE input files

The information in the Patran input text file was thereafter split and stored into two excel worksheets as listed below,

1. Worksheet "MICRO_RVE_NODES" contained the micro RVE nodal numbers and their coordinates
2. Worksheet "MICRO_RVE_ELEM_PROPS" contained element numbers, nodal connectivity and material properties as given in Table 3.5.

The next step was to create master and slave nodes for applying periodic boundary conditions and paring them.

3. The faces, edges and vertices nodes were stored in excel worksheet MICRO_PAIR. The purpose of creating paired nodes was that the slave nodes follow the behaviour of the master nodes such that, matching displacements could be applied to opposite micro RVE faces, edges and vertices. This ensured that the displacement boundary conditions were assigned to the master nodes and would be automatically assigned to the slave nodes.
4. The micro RVE faces, edges and vertices master nodes were assigned with the displacement boundary conditions which were followed by the corresponding slave nodes.

5. Before applying displacement boundary conditions, the nodes from the MICRO_RVE_NODES worksheet were separated and grouped as face, edges and vertices nodes by the VBA scripts pair_surf().
6. Thereafter the master-slave nodes for the face nodes groups were paired by the VBA script pair_surf(), and the pairing for the edges and vertices nodes groups were done by the VBA script pair_edge() respectively.

The matching displacements are a basic requirement of periodic boundary conditions as stated in the literature review section. For example, the face nodes from BAEF were assumed as master nodes and the face nodes CDHG was assumed as slave face nodes (refer Figure 3.20). The edges and vertices master-slave pairs are also shown in the same figure. Table 3.6 gives the list of groupings created by the VBA script pair_surf.

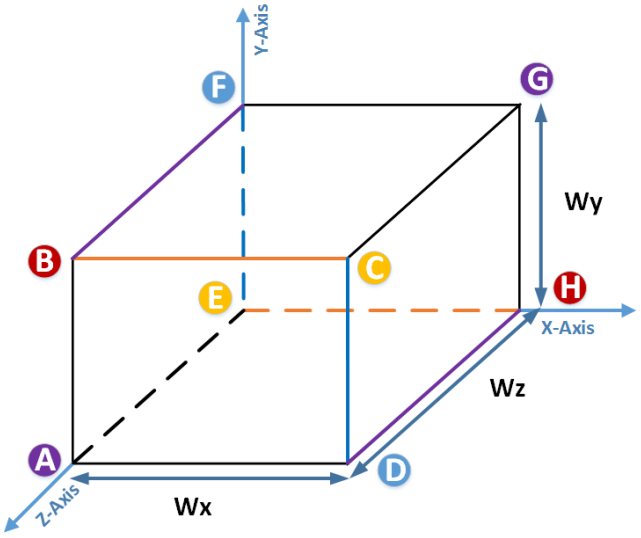


Figure 3.20 Colour coded node pairs for a Micro RVE model

Table 3.6 Micro RVE MPC groups

Micro RVE Vertices master-slave groups	These groups use all corner nodes
	D – F
	B – H
	A – G
	C – E
Micro RVE Edge master-slave groups	These groups use all node on edges excluding corner nodes
	CG – EA
	BF – DH
	CD – EF
	AB – GH
	BC – EH
	AD – FG
Micro RVE Face master-slave groups	These groups use all nodes on the micro RVE model faces excluding corner and edge nodes
	ABEF – CDGH
	ADEH – BCFG
	ABCD – EFGH

The next step was to apply six distinct periodic boundary conditions (three axial and three shear deformations) using multipoint constraints (MPCs) to the master-slave node groups. The values of the applied MPC displacements were calculated using equations 3.3, 3.4 and 3.5. These equations were based on axial strains in the x, y and z directions and shear strains, in the xy, yz and xz planes. The right hand side column vectors in the equations are the micro RVE's original length (W_x), width (W_y) and height (W_z). The MPC displacements were calculated using 10 % of the required strains. The displacements resulted in normal extensions in x, y and z directions and three shear deformations in xy, yz and xz planes as shown in Figure 3.21. The calculated MPC displacements and the equations were stored in worksheet "MICRO_EQUIV_DISP".

Vertices MPC displacements:

Matrix equation for C – E pair:

$$\begin{bmatrix}\Delta x\\ \Delta y\\ \Delta z\end{bmatrix} = \begin{bmatrix}\varepsilon_{xx} & \varepsilon_{xy} & \varepsilon_{xz}\\ \varepsilon_{xy} & \varepsilon_{yy} & \varepsilon_{yz}\\ \varepsilon_{xz} & \varepsilon_{yz} & \varepsilon_{zz}\end{bmatrix}\begin{bmatrix}W_x\\ W_y\\ W_z\end{bmatrix}$$

Matrix equation for A – G pair:

$$\begin{bmatrix}\Delta x\\ \Delta y\\ \Delta z\end{bmatrix} = \begin{bmatrix}\varepsilon_{xx} & \varepsilon_{xy} & \varepsilon_{xz}\\ \varepsilon_{xy} & \varepsilon_{yy} & \varepsilon_{yz}\\ \varepsilon_{xz} & \varepsilon_{yz} & \varepsilon_{zz}\end{bmatrix}\begin{bmatrix}-W_x\\ -W_y\\ W_z\end{bmatrix} \qquad (3.3)$$

Matrix equation for B – H pair:

$$\begin{bmatrix}\Delta x\\ \Delta y\\ \Delta z\end{bmatrix} = \begin{bmatrix}\varepsilon_{xx} & \varepsilon_{xy} & \varepsilon_{xz}\\ \varepsilon_{xy} & \varepsilon_{yy} & \varepsilon_{yz}\\ \varepsilon_{xz} & \varepsilon_{yz} & \varepsilon_{zz}\end{bmatrix}\begin{bmatrix}-W_x\\ W_y\\ W_z\end{bmatrix}$$

Matrix equation for D – F pair:

$$\begin{bmatrix}\Delta x\\ \Delta y\\ \Delta z\end{bmatrix} = \begin{bmatrix}\varepsilon_{xx} & \varepsilon_{xy} & \varepsilon_{xz}\\ \varepsilon_{xy} & \varepsilon_{yy} & \varepsilon_{yz}\\ \varepsilon_{xz} & \varepsilon_{yz} & \varepsilon_{zz}\end{bmatrix}\begin{bmatrix}W_x\\ -W_y\\ W_z\end{bmatrix}$$

Edges MPC displacements:

Matrix equation for CG – EA pair:

$$\begin{bmatrix}\Delta x\\ \Delta y\\ \Delta z\end{bmatrix} = \begin{bmatrix}\varepsilon_{xx} & \varepsilon_{xy}\\ \varepsilon_{xy} & \varepsilon_{yy}\\ \varepsilon_{xz} & \varepsilon_{yz}\end{bmatrix}\begin{bmatrix}W_x\\ W_y\end{bmatrix}$$

Matrix equation for BF – DH pair:

$$\begin{bmatrix}\Delta x\\ \Delta y\\ \Delta z\end{bmatrix} = \begin{bmatrix}\varepsilon_{xx} & \varepsilon_{xy}\\ \varepsilon_{xy} & \varepsilon_{yy}\\ \varepsilon_{xz} & \varepsilon_{yz}\end{bmatrix}\begin{bmatrix}-W_x\\ W_y\end{bmatrix} \qquad (3.4)$$

Matrix equation for CD – EF pair:

$$\begin{bmatrix}\Delta x\\ \Delta y\\ \Delta z\end{bmatrix} = \begin{bmatrix}\varepsilon_{xx} & \varepsilon_{xz}\\ \varepsilon_{xy} & \varepsilon_{yz}\\ \varepsilon_{xz} & \varepsilon_{zz}\end{bmatrix}\begin{bmatrix}W_x\\ W_z\end{bmatrix}$$

Matrix equation for AB – GH pair:

$$\begin{bmatrix}\Delta x\\ \Delta y\\ \Delta z\end{bmatrix} = \begin{bmatrix}\varepsilon_{xx} & \varepsilon_{xz}\\ \varepsilon_{xy} & \varepsilon_{yz}\\ \varepsilon_{xz} & \varepsilon_{zz}\end{bmatrix}\begin{bmatrix}-W_x\\ W_z\end{bmatrix}$$

Matrix equation for BC – EH pair:

$$\begin{bmatrix}\Delta x\\ \Delta y\\ \Delta z\end{bmatrix} = \begin{bmatrix}\varepsilon_{xy} & \varepsilon_{xz}\\ \varepsilon_{yy} & \varepsilon_{yz}\\ \varepsilon_{yz} & \varepsilon_{zz}\end{bmatrix}\begin{bmatrix}W_y\\ W_z\end{bmatrix}$$

Matrix equation for AD – FG pair:

$$\begin{bmatrix}\Delta x\\ \Delta y\\ \Delta z\end{bmatrix} = \begin{bmatrix}\varepsilon_{xy} & \varepsilon_{xz}\\ \varepsilon_{yy} & \varepsilon_{yz}\\ \varepsilon_{yz} & \varepsilon_{zz}\end{bmatrix}\begin{bmatrix}-W_y\\ W_z\end{bmatrix}$$

Face MPC displacements:

Matrix equation for ABEF – CDGH pair:

$$\begin{bmatrix}\Delta x\\ \Delta y\\ \Delta z\end{bmatrix} = \begin{bmatrix}\varepsilon_{xy} & 0 & 0\\ 0 & \varepsilon_{yy} & 0\\ 0 & 0 & \varepsilon_{yz}\end{bmatrix}\begin{bmatrix}-W_y\\ W_y\\ -W_y\end{bmatrix}$$

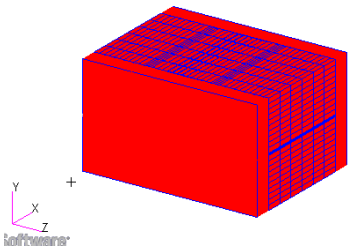
Matrix equation for ADEH – BCFG pair:

$$\begin{bmatrix}\Delta x\\ \Delta y\\ \Delta z\end{bmatrix} = \begin{bmatrix}\varepsilon_{xx} & 0 & 0\\ 0 & \varepsilon_{xy} & 0\\ 0 & 0 & \varepsilon_{xz}\end{bmatrix}\begin{bmatrix}W_x\\ -W_x\\ -W_x\end{bmatrix} \qquad (3.5)$$

Matrix equation for ABCD – EFGH pair:

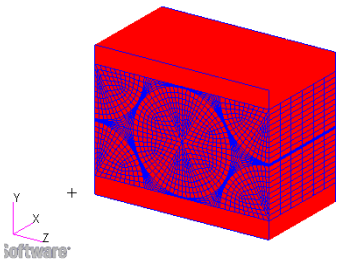
$$\begin{bmatrix}\Delta x\\ \Delta y\\ \Delta z\end{bmatrix} = \begin{bmatrix}\varepsilon_{xz} & 0 & 0\\ 0 & \varepsilon_{yz} & 0\\ 0 & 0 & \varepsilon_{zz}\end{bmatrix}\begin{bmatrix}-W_z\\ -W_z\\ W_z\end{bmatrix}$$

Case 1: extension in *xx*



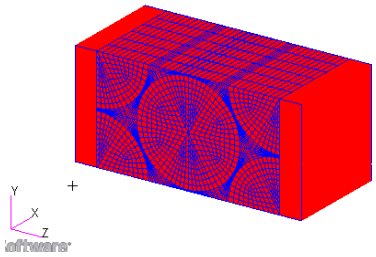
$$\begin{bmatrix}0.1 & 0 & 0\\ 0 & 0 & 0\\ 0 & 0 & 0\end{bmatrix}$$

Case 2: extension in *yy*



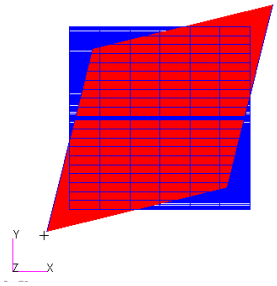
$$\begin{bmatrix}0 & 0 & 0\\ 0 & 0.1 & 0\\ 0 & 0 & 0\end{bmatrix}$$

Case 3: extension in *zz*



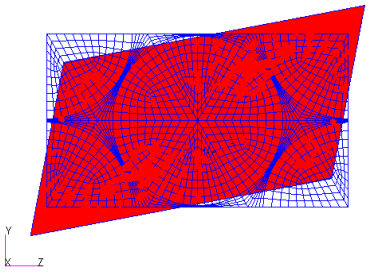
$$\begin{bmatrix}0 & 0 & 0\\ 0 & 0 & 0\\ 0 & 0 & 0.1\end{bmatrix}$$

Case 4: shear in *xy*



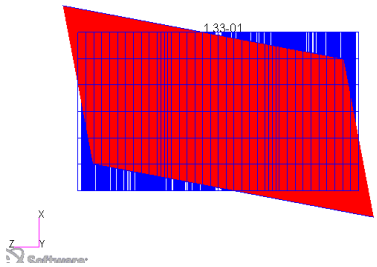
$$\begin{bmatrix} 0 & 0.1 & 0 \\ 0.1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

Case 5: shear in *yz*



$$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0.1 \\ 0 & 0.1 & 0 \end{bmatrix}$$

Case 6: shear in *xz*



$$\begin{bmatrix} 0 & 0 & 0.1 \\ 0 & 0 & 0 \\ 0.1 & 0 & 0 \end{bmatrix}$$

Figure 3.21 Six different cases to be solved in order to calculate the homogenised elastic properties of the RVE

Thereafter the new input files were generated using VBA script "micro_bdf_file ()", which transferred the nodal coordinates from worksheet "MICRO_RVE_NODES" and nodal connectivity and element properties from worksheet "MICRO_RVE_ELEM_PROPS" respectively. Then the VBA script "micro_mpc_pair ()" were used to write the MPC displacements for the master-slave node pairs from worksheet "MICRO_PAIR" and the zero and non-zero MPC displacements from "MICRO_EQUIV_DISP" to the input files. Six distinct micro RVE input text files were generated for each case shown in Figure 3.21. For example, in case one, $\varepsilon_{xx} = 0.1$ was substituted into the equations to calculate the MPC displacements. Both zero and non-zero MPC displacements were obtained from the equations. The zero MPC displacements (Δ_y and Δ_z) defined the fixed boundary conditions along y and z directions and the non-zero displacements (Δ_x) defined the displacements along the x direction applied to the master-slave node groups. The MPC displacements were applied to both master and slave nodes simultaneously. In addition to the MPC boundary conditions, one node at the centre of the micro RVE was fixed in x, y and z direction to prevent rigid body displacements. A Similar procedure was followed for other five input files by substituting$\varepsilon_{yy} = 0.1$, $\varepsilon_{zz} = 0.1$, $\varepsilon_{xy} = 0.1$, $\varepsilon_{zy} = 0.1$ and $\varepsilon_{xz} = 0.1$ in the MPC equations and their respective input text files were generated.

A linear static solver in MSc Nastran 2014 (SOL 101) was used to solve the six text input files and displacement and stress results were obtained from the finite element analysis. The displacement results were used for sanity checks to validate the modelling process. The micro RVE displacement results for these six input text files are shown in Figure 3.21. The blue meshed blocks are the original micro RVE and the red blocks represent the deformed micro RVE. For case 1 to case 3 the results are shown in an isometric view and for case 4 to case 6 the results are shown in the plane of the shear deformations. Thereafter, the micro RVE elements stresses output from Nastran 2014 solver was written to a text file and was then converted into a table and stored in worksheet "MICRO_RVE_ELEM_PROPS". This table was again transferred to an Excel virtual array to simplify subsequent matrix operations. The array's number of rows was the same as the number of micro RVE elements and it had six columns for storing xx, yy, zz, xy, yz and xz components of the micro RVE elements stress output (i.e. the array size in terms of rows by columns was; number of micro RVE elements by six). The structure of the micro elements stress array is shown in equation (3.6) where m stands for element number.

$$\sigma_{micro_elm_stress_array} = \begin{bmatrix} \sigma_{xx_1} & \sigma_{yy_1} & \sigma_{zz_1} & \sigma_{xy_1} & \sigma_{yz_1} & \sigma_{zx_1} \\ \sigma_{xx_2} & \sigma_{yy_2} & \sigma_{zz_2} & \sigma_{xy_2} & \sigma_{yz_2} & \sigma_{zx_2} \\ \sigma_{xx_3} & \sigma_{yy_3} & \sigma_{zz_3} & \sigma_{xy_3} & \sigma_{yz_3} & \sigma_{zx_3} \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \sigma_{xx_m} & \sigma_{yy_m} & \sigma_{zz_m} & \sigma_{xy_m} & \sigma_{yz_m} & \sigma_{zx_m} \end{bmatrix} \quad (3.6)$$

From MSc Patran 2014, a column vector of micro element strains was generated and stored in worksheet "MICRO_ELEM_VOLUME". Subsequently, the Micro RVE element volumes details from the worksheet "MICRO_ELEM_VOLUME" were also transferred to a second excel virtual array, which had a single row and its number of columns were the same as the number of micro RVE elements (i.e. the array in terms of rows by columns was one by the number of micro RVE elements). The structure of the micro element volumes array is also shown in equation (3.7).

$$V_{micro_elm_volumes} = [V_1 \quad V_2 \quad V_3 \quad \cdot \quad \cdot \quad \cdot \quad V_m] \quad (3.7)$$

The next step was to determine the homogeneous stiffness matrix (S_{micro}) of the micro RVE model, which involved a sequence of matrix multiplication and scalar division operations. To obtain the homogeneous stiffness matrix the results of equation (3.6) and (3.7) were multiplied and divided by the sum of the micro RVE element volumes. This operation is shown in equation (3.8).

$$S_{colm_case_i} = Transpose\left[V_{micro_elms_volumes} \times \sigma_{micro_elms_stress_array}\right] \times \frac{1}{\sum_{i=1}^{m} V_i} \quad (3.8)$$

The first column of the stiffness matrix was determined from the load case 1 stress array and micro element volumes. The resulting row vector was transposed to obtain the first column of the stiffness matrix. The other columns were determined from the stress arrays of the other 5 loading cases in Figure 3.21. Then the compliance matrix (C_{micro}) was obtained from the inverse of the stiffness matrix before calculating the effective engineering moduli. The compliance matrix and its relationship to the engineering moduli are shown in Appendix A. VBA script "micro_con_tangent ()" was used to calculate the micro RVE stiffness and compliance matrices. These matrices were stored in worksheet "MICRO_CONST_TANG". For example the Elastic modulus along the xx-direction was obtained from the reciprocal of the first component of the compliance matrix.

3.3.4 Meso Model

The meso scale model comprises of a matrix and plain woven carbon fibre tows, which were used to determine the effective material properties of the homogenised meso RVE model. The homogenised meso RVE material properties were used in the macro model for the ply material properties since the plies are a combination of epoxy matrix and composite tows materials. The geometric properties of the composite warp and weft tows are provided in Table 3.7.

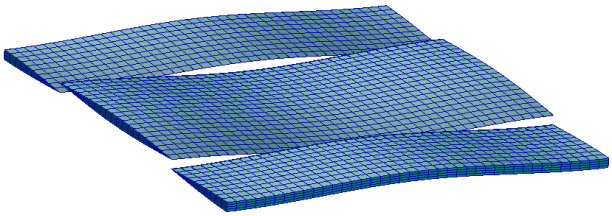
Table 3.7 Meso model geometric properties

Meso RVE details	Values (μm)
Warp and Weft Tow width (W)	2000
Warp and Weft Tow thickness (T)	90
Tow volume fraction (VT %)	60
Matrix volume fraction (VM %)	40
RVE Width (RW)	4000
RVE Thickness (RH)	182
RVE Length (RL)	4000

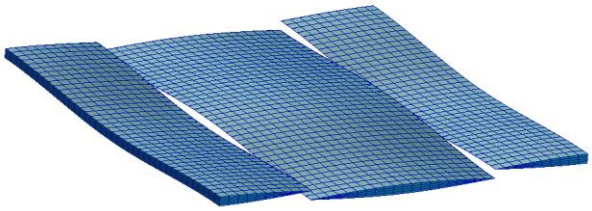
The meso RVE model was modelled with a warp and a weft tows together with two half-width weft and warps tows on their sides. The weft and warp tows had same thickness and width. They were placed perpendicular to each other. A sine wave was used to model the waviness associated with the woven carbon fibre model and the waviness was modelled using the equation given below.

$$Z = \frac{RH}{2} \times SIN\left(\frac{2X\pi}{RW}\right) \tag{3.9}$$

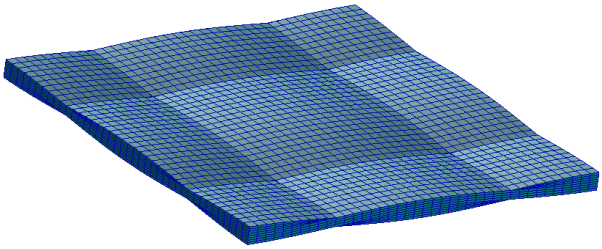
The parameters RH and RW are meso RVE thickness and width respectively as stated in Table 3.7. As a result of the waviness, gaps were created along their edges, which allowed the weft and warp tows to align perfectly to model the woven cross section of the meso RVE model. Figure 3.22 (a) and (b) show the weft and warp tows respectively.



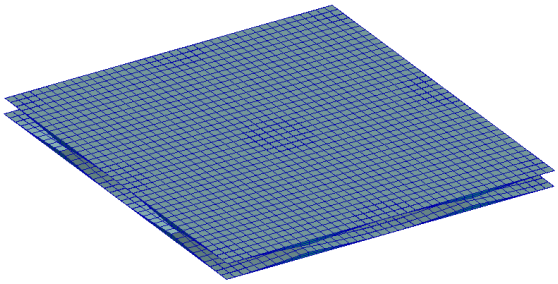
(a) Weft tows elements



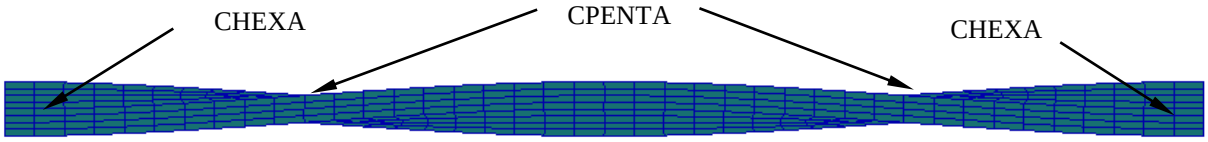
(b) Warp tows elements



(c) Isometric view without the matrix (plain weave)



(d) Matrix elements



(e) End view of the model without the matrix (plain weave)

Figure 3.22 Meso model element groups for a plain weave

The combined model of the weft and warp tows is show in Figure 3.22 (c). To complete the meso RVE model, the matrix phase (refer to Figure 3.22 (d)) was added to give flat surfaces on the top and bottom sides of the model. The matrix material properties of the meso RVE model are the same as those of the matrix material properties used in the micro RVE model. The tows material properties were determined from the homogenised material properties of the micro RVE model. Warp tow material properties in the global meso RVE coordinate system are given in the matrix below. Material card “MAT9” was used to insert these properties into the input text file.

$$10^9 \times \begin{bmatrix} 181 & 9.4 & 9.4 & 0 & 0 & 0 \\ 9.4 & 23.4 & 7.9 & 0 & 0 & 0 \\ 9.4 & 7.9 & 23.4 & 0 & 0 & 0 \\ 0 & 0 & 0 & 16.3 & 0 & 0 \\ 0 & 0 & 0 & 0 & 15.5 & 0 \\ 0 & 0 & 0 & 0 & 0 & 16.3 \end{bmatrix}$$

Weft tow material properties in the global meso RVE coordinate system are given in the matrix below. Material card "MAT9" was used to insert these properties into the input text file.

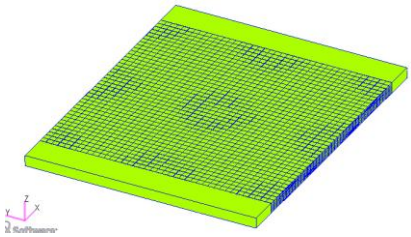
$$10^9 \times \begin{bmatrix} 23.4 & 9.4 & 7.9 & 0 & 0 & 0 \\ 9.4 & 181 & 9.4 & 0 & 0 & 0 \\ 7.9 & 9.4 & 23.4 & 0 & 0 & 0 \\ 0 & 0 & 0 & 15.5 & 0 & 0 \\ 0 & 0 & 0 & 0 & 16.3 & 0 \\ 0 & 0 & 0 & 0 & 0 & 16.3 \end{bmatrix}$$

CHEXA Solid brick elements were used to model the warp and weft tows in MSc Patran 2014. A combination of CHEXA solid brick and CPENTA solid wedge elements were used to model the tows and matrix phases of the meso RVE model. The original meso RVE model and its input text file were created in MSc Patran 2014. All the other parts of the input text file were defined using MSc Patran except the boundary conditions. The meso RVE input text file was transferred and stored in new worksheets within the same Excel spreadsheet used to store the micro RVE model element and nodes attributes. The meso RVE model worksheets started with the prefix "MESO". The information in the Patran input text file was thereafter split and stored into two excel worksheets as listed below.

1. The weft, warp and matrix element groups information was stored in worksheet "MESO_RVE_ELEM_PROPS".
2. The nodal coordinates were stored in worksheet "MESO_RVE_NODES".

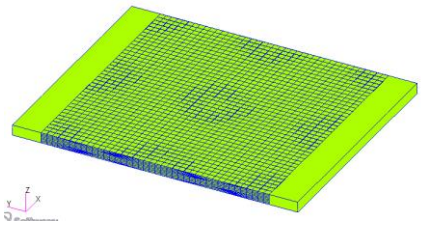
Before the meso RVE text files were solved by MSc Nastran 2014 (SOL 101), periodic boundary conditions were added to them. The same process used to add periodic boundary conditions to the micro RVE text input files was used to add periodic boundary conditions to the meso RVE input text files. The VBA scripts used for grouping the micro RVE master-slave nodes were also used to group the meso RVE model master-slave nodes. Slave-master node pairs for faces, edges and vertices were paired by VBA scripts "pair_edge ()" and "pair_surf ()". Worksheet "MESO_PAIR" was used to store groups of paired faces, edges and vertices nodes. VBA script "meso_mpc_pairs ()" was used to apply MPC displacement loads on all meso RVE face, edges and vertices nodes during the creation of the meso RVE analysis input text file. The MPC displacements were calculated in worksheet "MESO_EQUIV_DISP" using the equations explained in the micro model section 3.3.3. The creation of the meso RVE analysis input text files was executed by VBA scripts "meso_bdf_file ()". This VBA script was used to generate six input text files similar to the micro RVE input files. Similarly a linear static solver in MSc Nastran 2014 (SOL 101) was used to solve the six text input files. The following displacement results were obtained from the six models.

Case 1: extension in *xx*



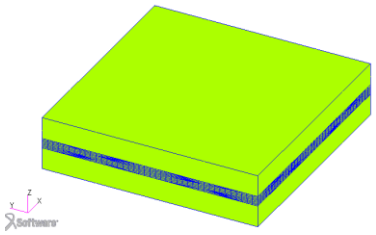
$$\begin{bmatrix} 0.1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

Case 2: extension in *yy*



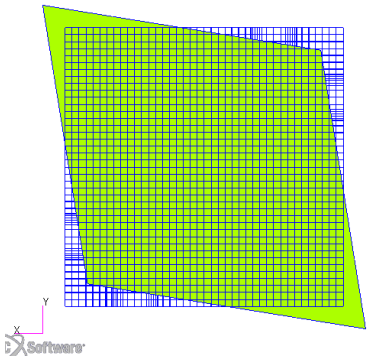
$$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0.1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

Case 3: extension in *zz*



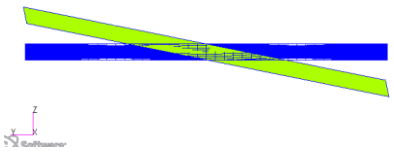
$$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0.1 \end{bmatrix}$$

Case 4: shear in *xy*



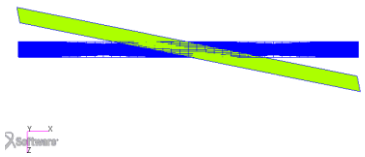
$$\begin{bmatrix} 0 & 0.1 & 0 \\ 0.1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

Case 5: shear in *yz*



$$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0.1 \\ 0 & 0.1 & 0 \end{bmatrix}$$

Case 6: shear in *xz*



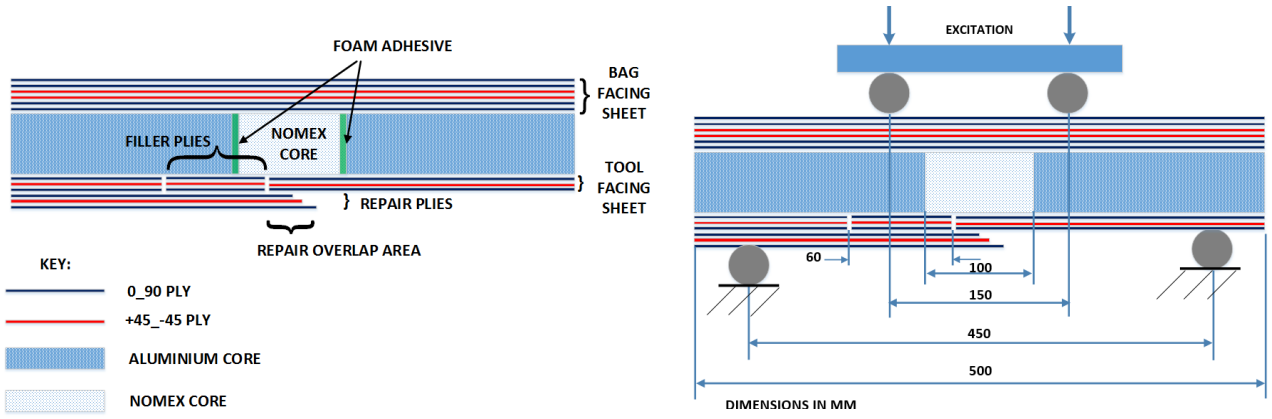
$$\begin{bmatrix} 0 & 0 & 0.1 \\ 0 & 0 & 0 \\ 0.1 & 0 & 0 \end{bmatrix}$$

Figure 3.23 Six cases to be solved for the homogenised material properties of the meso RVE

The strains and element stresses results were obtained from the finite element analysis. The same process used to determine the homogenised engineering properties in the micro RVE model case was used to determine the meso RVE model homogenised engineering properties. After determining the meso RVE stiffness matrix (S_{meso}), the compliance matrix (C_{meso}) was obtained from the inverse of the stiffness matrix thereafter the effective engineering moduli were calculated from this compliance matrix. VBA script "meso_con_tangent ()" was used to calculate the meso RVE stiffness and compliance matrices. These matrices were stored in worksheet "MESO_CONST_TANG". The homogenised meso RVE material properties were used in the macro model.

3.3.5 Macro Model

The macro model was created from geometric properties of the test specimen. These properties were outlined in the Airbus AITM 1-0018 test specification. A schematic of the test specimen in the four-point-bending fixture with important dimensions is shown in Figure 3.24 (b). The specimen width of 50 mm and a core thickness of 35 mm were used. From the manufacturer's specifications and practical measurements, the ply thickness used was 0.2 mm.



(a) Section view of the sandwich specimen (b) Specimen dimensions and fixture setup

Figure 3.24 Macro model geometry and materials

For the material properties in terms of ply layups direction, a stacking sequence of $0^o/45^o/0^o$ was used on the tool facing sheet of the test specimen and a stacking sequence $0^o/0^o/45^o/45^o/0^o/0^o$ was used on the bag facing sheet. The stacking sequences of both facing sheets for the test specimens were as stated in Table 3.8. Plies at 0 degrees used the direct homogenised meso RVE material properties but these properties were rotated by 45 degrees for plies at laid at 45 degrees.

Table 3.8 Stacking sequence for the laminate before repair implementation

Specimen	Stacking Sequence	Nominal Thickness [mm]
Tool Facing Sheet	$[0^o/45^o/0^o]$	0.2 by 3 = 0.6
Core	n/a	35
Bag Facing Sheet	$[0^o/0^o/45^o/45^o/0^o/0^o]$	0.2 by 6 = 1.2

Additional plies were modelled on the tool facing side for representing three filler and repair plies respetively. The filler plies were modelled with the stacking sequence of $0^o/45^o/0^o$. The repair plies were modelled with the stacking sequence of $0^o/45^o/0^o$ with their edges overlapping within the Nomex core, test area, as shown in Figure 3.6 (f). Beneath the overlapping plies were filler plies modelled directly over the Nomex core and aluminium core in the damage slot. The overlap distance between the three repair layup was 20 mm. The 0^o ply material properties in the global macro model coordinate system are given in the matrix below. Material card "MAT9" was used to insert these properties into the input text file.

$$10^9 \times \begin{bmatrix} 64 & 7.08 & 5 & 0 & 0 & 0 \\ 7.08 & 64 & 5 & 0 & 0 & 0 \\ 5 & 5 & 11.9 & 0 & 0 & 0 \\ 0 & 0 & 0 & 18.7 & 0 & 0 \\ 0 & 0 & 0 & 0 & 7.06 & 0 \\ 0 & 0 & 0 & 0 & 0 & 7.09 \end{bmatrix}$$

The 45^o ply material properties in the global macro model coordinate system are given in the matrix below. Material card "MAT9" was used to insert these properties into the input text file.

$$\begin{bmatrix} 39.1 & 32 & 5 & 0 & 0 & 0 \\ 32 & 39.1 & 5 & 0 & 0 & 0 \\ 5 & 5 & 11.9 & 0 & 0 & 0 \\ 0 & 0 & 0 & 12.9 & 0 & 0 \\ 0 & 0 & 0 & 0 & 12.9 & 0 \\ 0 & 0 & 0 & 0 & 0 & 57 \end{bmatrix}$$

Two types of cores were used for the specimen model. These were the reinforcement core in the form of the stronger aluminium honeycomb core and the test core in the form of Nomex core but the focus was on the test core (Nomex core). The cores were modelled with solid CHEXA (brick) elements as shown in Figure 3.25 (d). Their material properties are given in Table 3.9.

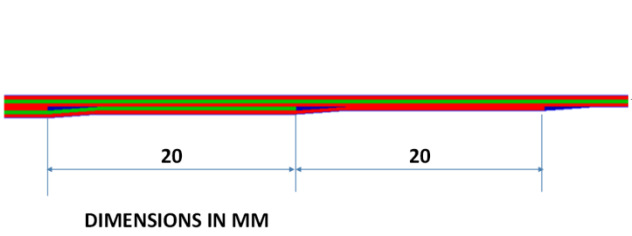
Table 3.9 Material properties for Aluminium and Nomex honeycombs

	Elastic Modulus (MPa)	Shear Modulus (MPa)
	E11, E22, E33	E12, E13, E23
Aluminium Honeycomb	140	54
Nomex Honeycomb	90	34

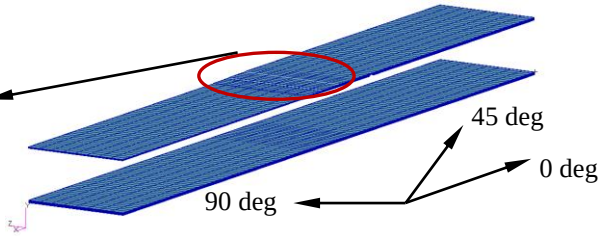
On the reinforced side of the specimen, the plies were modelled as thin plates of solid CHEXA elements as shown in Figure 3.25 (b). Their material properties were determined from the homogenised meso material properties. These material properties also apply to the tool facing side ply properties. On the tool facing side, thin plates of solid CHEXA elements were used for the filler, repair and original laminate plies. Matrix brick elements were used to fill the gaps between the original tool facing sheets and filler plies. They were also used at the ends of the repair plies in the form of solid CPENTA (wedge) elements to model the matrix areas (blue triangles) in Figure 3.25 (a).

The macro model was loaded by four rollers in a configuration as shown in Figure 3.24 (b). This loading configuration results in a pure bending moment, which is similar to loads applied to sandwich laminates in practical applications. It also allows tensile and compressive loads to be applied to the sandwich facing sheets without crushing the weaker test core. Inner rollers were modelled as grid point force loads and the outer rollers as grid point constrains to complete the four point bending setup. Macro loads application started with 10% of the experimental failure load and the loading increments were stored in worksheet “LOAD_CONTROL” of the analysis spreadsheet. Figure 3.25 (e,f) show the boundary conditions and their application points. The macro model input text file information was manually stored in worksheets of the same excel spreadsheet used for storing the input text files of the micro RVE and meso RVE models. The macro model nodes coordinates were stored in the “REPAIR NODES” worksheet. The macro elements were grouped according to ply, core and matrix properties and stored in worksheet “REPAIR ELEM PROPS”. The element groups are shown in Figure 3.25. VBA script “repair_bdf_file ()” was used to transfer data from the “REPAIR NODES”, “LOAD_CONTROL” and “REPAIR ELEM PROPS” worksheets and generate the macro analysis input file. The linear static solver in MSc Nastran 2014 (SOL 101) was used to solve the macro input text files and displacement and strain results were obtained from the finite element analysis.

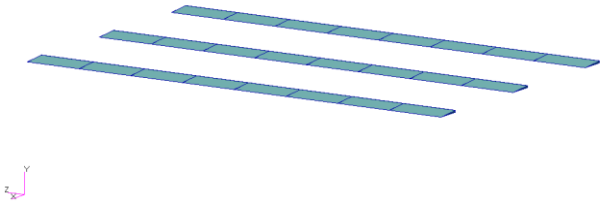
To determine the macro failure load, it was incremented by 10% until the most strained warp and weft tows reached the tensile strength stresses for the matrix and carbon fibre. Warp tow elements were assumed to have failed when their highest strain caused a continuous plane of carbon fibre elements to fail. Weft tows on the other hand were assumed to have failed when their highest strain caused a continuous plane of matrix elements to fail. Warp tow elements failed at a tensile stress of $S_{Static\ Warp\ Strength\ Stress} = 2400$ MPa and weft tow elements falied at a tensile stress of $S_{Static\ Weft\ Strength\ Stress} = 570$ MPa.



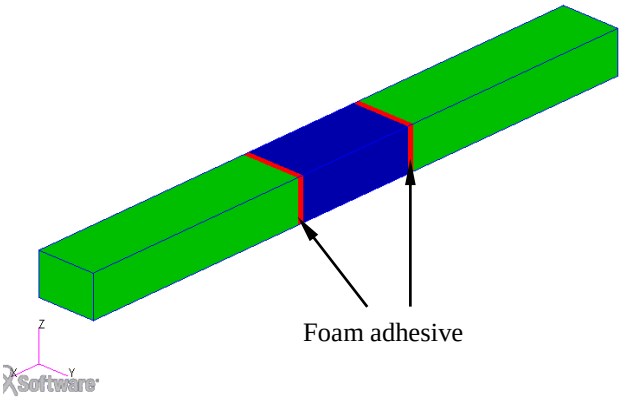
(a) Repair ply drops-offs and overlap distances



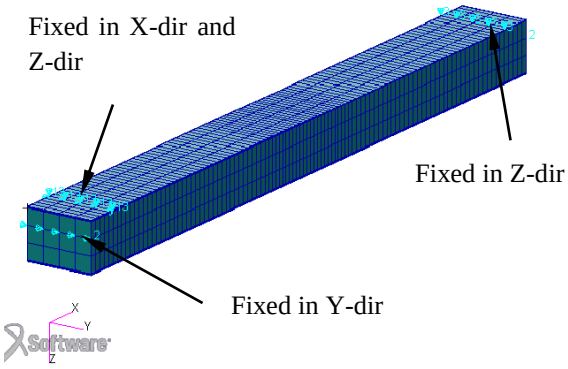
(b) Facing sheet elements for the macro model



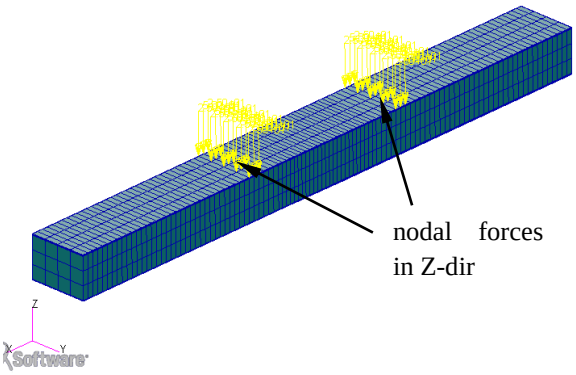
(c) Matrix ramps at repair ply drop-offs



(d) Nomex (blue) and aluminium (green) cores



(e) Boundary constraints of the macro model



(f) Nodal forces on the macro model

Figure 3.25 Macro model element groups

3.3.6 Static and Fatigue Progressive Failure Analysis

The multiscale modelling framework used in this research was based on a hierarchical coupling procedure. The modelling procedure individually analysed three length scales namely; micro scale, meso scale and macro scale, as shown in Figure 3.13. Now that the creation of the micro RVE, meso RVE and macro models has been explained , the cyclic MsM procedure will be discussed further. The VBA script "Looping Macro ()" was used to implement the MsM procedure, which started with determining the tow material properties as explained in the micro model section. This was immediately followed by the determination of the ply material properties as explained in the meso model section. With the tow and ply material properties determined, the macro model was created as explained in the macro model section. Thereafter the macro model critical three dimensional strain results (ε_{macro}) were extracted from critical points A or B. A sample of the critical macro strain is given below.

$$\varepsilon_{macro} = \begin{bmatrix} \varepsilon_{xx} & \varepsilon_{xy} & \varepsilon_{xz} \\ \varepsilon_{xy} & \varepsilon_{yy} & \varepsilon_{yz} \\ \varepsilon_{xz} & \varepsilon_{yz} & \varepsilon_{zz} \end{bmatrix} = 10^{-3} \times \begin{bmatrix} 2.8 & -0.019 & 0.09 \\ -0.019 & -0.654 & 0.058 \\ 0.09 & 0.058 & -0.885 \end{bmatrix} \tag{3.10}$$

At this step of the analysis, a meso RVE model was created using VBA script "crit_str_meso_bdf_file" and script "crit_meso_runnastran" was used to solve the meso model so that failed elements could be identified. Thereafter VBA script "meso_crit_strain" was implemented to read and extract the critical three dimensional strain tensor (ε_{meso}) from the warp or weft tows. A sample of this critical meso strain is given below.

$$\varepsilon_{meso} = \begin{bmatrix} \varepsilon_{xx} & \varepsilon_{xy} & \varepsilon_{xz} \\ \varepsilon_{xy} & \varepsilon_{yy} & \varepsilon_{yz} \\ \varepsilon_{xz} & \varepsilon_{yz} & \varepsilon_{zz} \end{bmatrix} = 10^{-3} \times \begin{bmatrix} 2.67 & -0.037 & 1.22 \\ -0.073 & -0.705 & 0.125 \\ 1.22 & 0.125 & -0.783 \end{bmatrix} \tag{3.11}$$

Consequently the elements stresses (S_{stress}) were also extracted from the meso model, which were then divided by the tensile strength ($S_{strength}$) of matrix, weft and warp tows individually. This calculation determined the failure indices (FI) of the meso RVE elements elements based on equation (3.12) to (3.13). Failure indices and stresses for the critical meso model elements were stored in worksheet "MESO_RVE_ELEM_PROPS".

$$FI_{STATIC_WEFT} = \frac{S_{Weft\ Tens\ Stress}}{S_{Weft\ Strength}} \tag{3.12}$$

$$FI_{STATIC_WARP} = \frac{S_{Warp\ Tens\ Stress}}{S_{Warp\ Strength}} \tag{3.13}$$

If the failure indices exceeds one then the corresponding element material properties were reduced by a factor of 0.001. The processes of extracting critical meso RVE element stresses, calculating failure indices and material properties degradation were executed by VBA script "meso_check_mat_update ()" (static).

A similar process was followed for the micro RVE model. The micro RVE model was created by VBA script "crit_str_micro_bdf_file" and solved by VBA script "crit_micro_RunNastran". Thereafter VBA script "micro_mat_update" extracted the matrix and fibre elements stresses and divided by the matrix and fibre tensile strength (as shown in Table 3.5. To obtain the failure indices as shown in equations (3.14) and (3.15).

For micro model elements whose failure indices exceeded one their material properties were reduced by a factor of 0.001. Failure indices and Von Mises stresses for the micro RVE model elements were stored in worksheet "MICRO_RVE_ELEM_PROPS".

$$FI_{STATIC_MATRIX} = \frac{S_{Von\ Mises\ Matrix\ Stress}}{S_{Matrix\ Stregnth}} \tag{3.14}$$

$$FI_{STATIC_FIBRE} = \frac{S_{Von\ Mises\ Fibre\ Stress}}{S_{Fibre\ Strength}} \tag{3.15}$$

If the failure indices of the micro RVE or meso RVE models were less than one then the material properties (C_{micro} and C_{meso}) as explained in the micro model and meso model sections were redetermined before proceeding to the next macro load increment.

The MsM progressive damage analysis strain and materials coupling processes were implemented between the modelling scales as shown in Figure 3.26. This coupling procedure allows load increments and material properties degradation processes to be implemented together within the multiscale modelling framework. For the sake of simplicity the cross-functional flow diagram is drawn for the three length scales. Arrows pointing from a large scale to a small scale imply strain coupling between modelling scales and arrows pointing from a small scale to a large scale imply material properties coupling between modelling scales. A “For” looping VBA script “Looping Macro ()” was used to increment the macro load of the four point bending repair MsM analysis.

The macro loading parameters were saved in worksheet "LOAD_CONTROL" of the main analysis spreadsheet and the first loading condition was set to be low such that it did not cause any element failure. The macro loads were incremented with large steps within the elastic range of the ply material properties and the steps were reduced as the matrix and fibre elements started failing. For the MsM progressive failure analysis, at the meso scale, macro elements were selected arbitrarily from critical points A or B. And the progressive failure analysis at the micro scale used warp ramp elements.

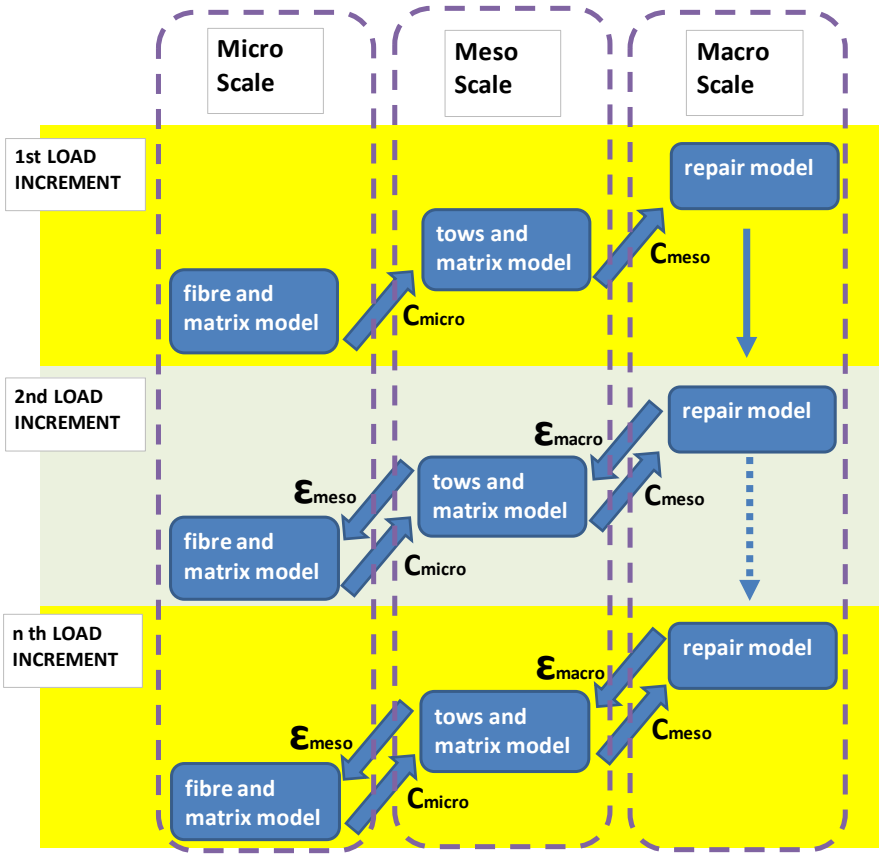


Figure 3.26 General MsM progressive failure analysis flow diagram

MsM fatigue analysis was implemented at the meso and micro scales by initially supplying fatigue load blocks in worksheet "LOAD_CONTROL". The fatigue loading blocks defined the constant load amplitude and the number of cycles (n_i). At the micro scale, VBA script "micro_mat_update" was used to extract micro element tensile stresses (σ_{Tens}) for calculating fatigue failure indices for a fatigue block loading ratio of 0.1. The general Equation (3.16) from the literature review section was used to determine fatigue critical cycles ($N_{Critical}$) for the micro element stresses. Since the general equation only needs the micro element material tensile strength ($S_{Tensile_Strength}$) it was modified for the matrix and fibre elements as presented in equation (3.17) and equation (3.18). Fatigue failure indices for micro RVE elements were stored in worksheet "MICRO_RVE_ELEM_PROPS".

$$\frac{\sigma_{Tens}}{S_{Tensile_Strength}} = -0.0506 \ln(N_{Critical}) + 1.059 \tag{3.16}$$

$$\mathrm{N_{Critical_matrix}} = \mathrm{e}^{\left(\frac{\sigma_{\mathrm{matrix_Tens}} - S_{matrix_Tens_Strength}}{-2}\right)} \tag{3.17}$$

$$\mathrm{N_{Critical_fibre}} = \mathrm{e}^{\left(\frac{\sigma_{\mathrm{fibre_Tens}} - S_{fibre_Tens_Strength}}{-85}\right)} \tag{3.18}$$

At the meso scale MsM fatigue analysis was implemented within VBA script "meso_check_mat_update ()". This script extracted meso element tensile stresses and calculated their fatigue failure indices. Equation (3.16) was used in the same way as it was for the micro scale. Fatigue critical cycles were calculated using equation (3.20) for the warp elements and equation (3.19) for the weft elements. The equations used the warp material tensile strength and weft material tensile strength by substituting them in the general equation (3.14). Fatigue failure indices for the meso RVE elements were stored in worksheet "MESO_RVE_ELEM_PROPS".

$$\mathrm{N_{Critical_weft}} = \mathrm{e}^{\left(\frac{\sigma_{\mathrm{weft_Tens}} - S_{weft_Tens_Strength}}{-20.85}\right)} \tag{3.19}$$

$$\mathrm{N_{Critical_warp}} = \mathrm{e}^{\left(\frac{\sigma_{\mathrm{warp_Tens}} - S_{warp_Tens_Strength}}{-84.6}\right)} \tag{3.20}$$

Each individual fatigue cycle was assumed to induce fatigue damage (D_i) within the element materials. Since the fatigue load amplitude within a fatigue loading block was constant the damage was based on equation (3.21) for a single load cycle. To determine the fatigue block damage the individual cycle damage (D_i) was multiplied by the number of fatigue loading cycles (n_i). The total fatigue damage by various loading blocks was calculated using Palmgren – Miner's rule as presented in equation (3.22).

$$D_i = \frac{1}{N_{Critical}} \tag{3.21}$$

$$D_T = \sum \frac{n_i}{N_{crititcal}} \tag{3.22}$$

Fatigue failure for any element was attained when the total fatigue damage reached a value of one. The accumulated fatigue values were stored in worksheet "MICRO_RVE_ELEM_PROPS" for micro RVE elements and worksheet "MESO_RVE_ELEM_PROPS" for meso RVE elements.

CHAPTER 4 - RESULTS AND DISCUSSION

Force-displacement graphs for the static experiments and MsM will be discussed first and subsequently the critical points on the repair scheme will be analysed. The multiscale modelling analysis results for the repair scheme will be discussed and the comparative analysis between the experimental and the MsM results will be presented separately for three length scale of micro, meso and macro.

4.1 Static Experimental Results

Figure 4.1 shows the applied load versus displacement of the four-point bending static experimental results. Two types of testing fixtures were used for the static experiments because the second testing fixute was used for the fatigue experiment thus faciliting the testing procedure . The results show a linear relationship between the applied load and the displacement for all the test. The plot was not linear below the applied load of 1 kN due to the stabilisation of the specimens at the beginning of the load cycle.

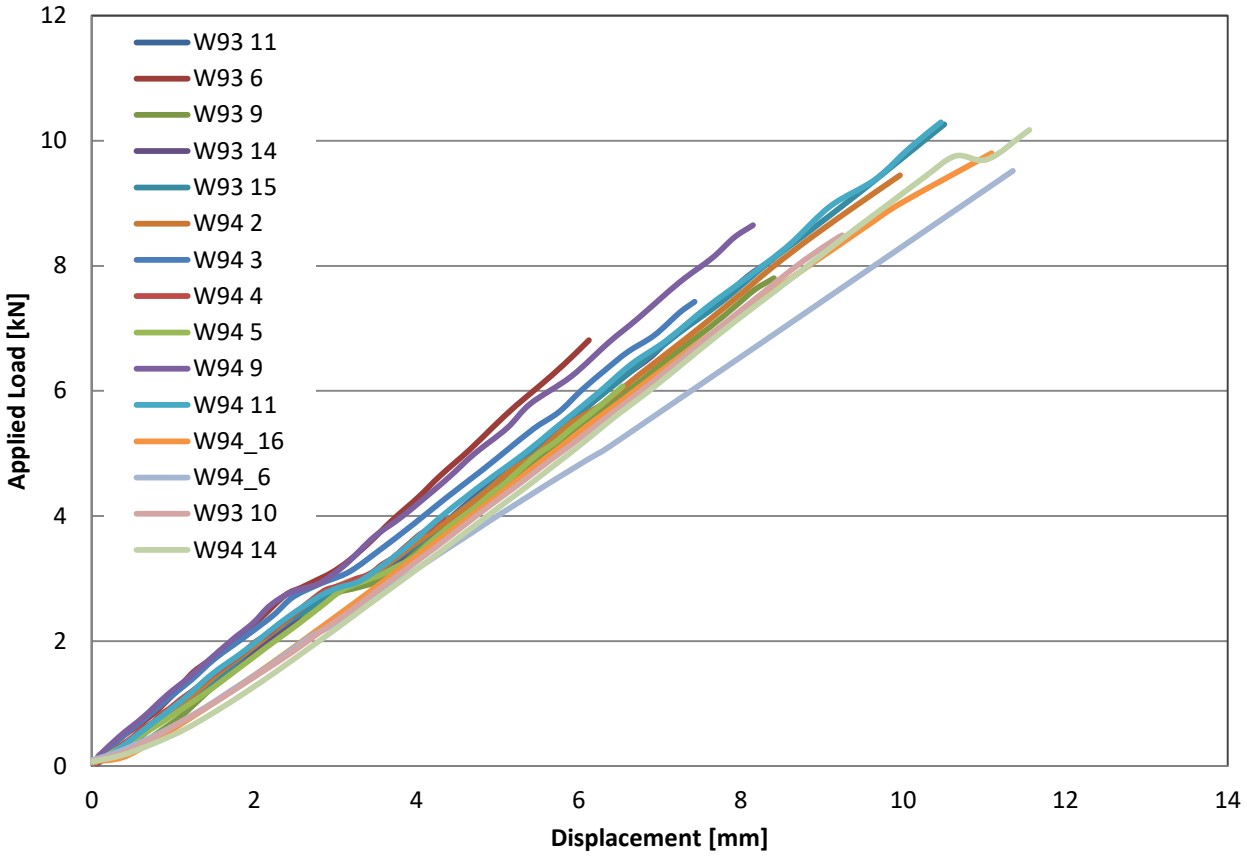


Figure 4.1 Displacement versus applied transverse load

Table 4.1 shows the static experimental failure loads of the facing sheets along with the mean and standard deviation. The standard deviation was found to be less than 5% of the mean failure load. The variance of the failure loads is shown in Figure 4.2. The values were closely clustered around the mean failure load as shown by short whiskers. The box also shows that the interquartile range is small within the 25th and 75th percentile values and is very close to the mean load (Red Cross). Given the fact that the statistical analysis of the experimental results exhibited a small variance relative to the mean failure load and Young's modulus was consistent for most of the loading

range, it can be concluded that the experimental results are representative value of the sandwich beam failure load.

The resultant facing sheet tensile failure stresses were calculated using the failure loads and given in Table 4.1.

Table 4.1 Applied failure loads for specimens

Specimen Number	Applied Failure Load [N]	Facing Sheet Failure Stress [MPa]	Failure Point
W94_2	9404	656	A
W93_15	10178	710	B
W94_11	10132	706	A
W94_14	10173	709	A
Mean	9972	695	
Standard	379	26	

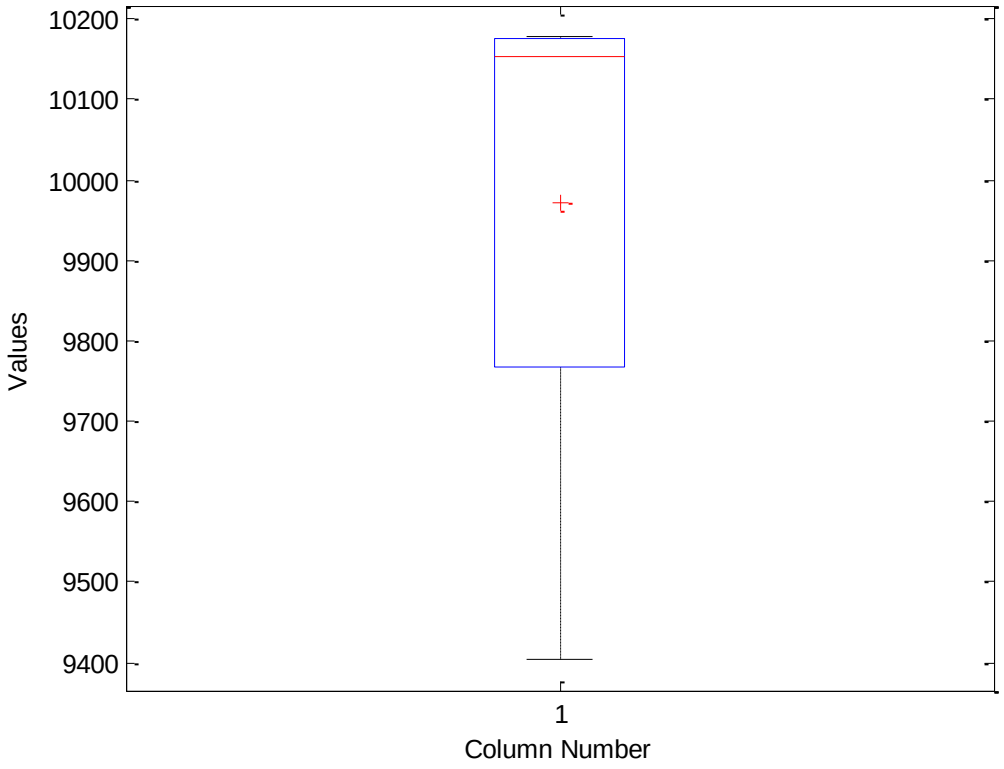


Figure 4.2 Box and whiskers plot of repaired facing sheet failure loads

During the static experiment, the longitudinal strains on the repaired facing sheet were recorded by using strain gauges on three specimens. The longitudinal positions of the strain gauges are shown in Figure 4.3 which includes a closeup of the strain gauge positions. Figure 4.4 shows the strain values at points E4 (x=235 mm), E3 (x=255 mm), E2 (x=275 mm) and E1 (x=285) corresponding to the applied load.

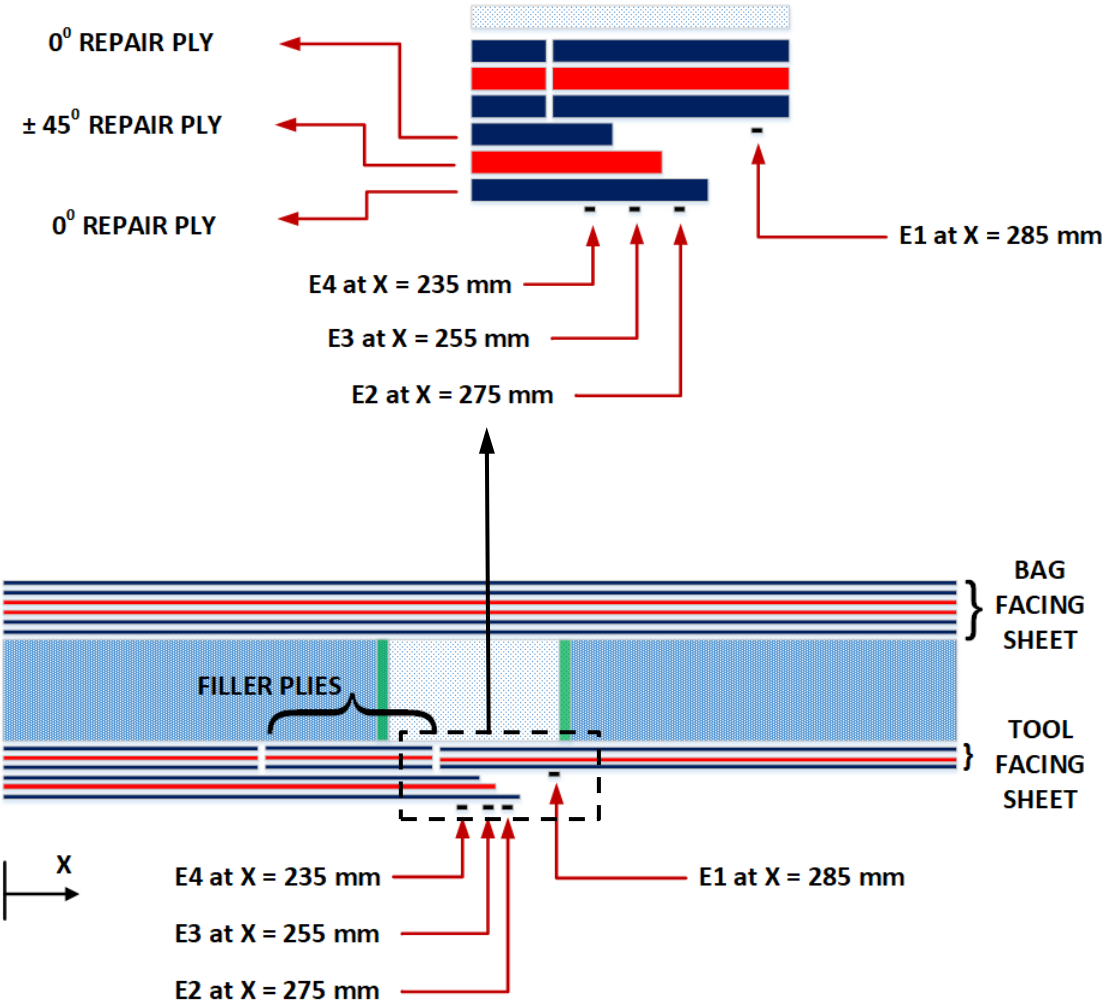
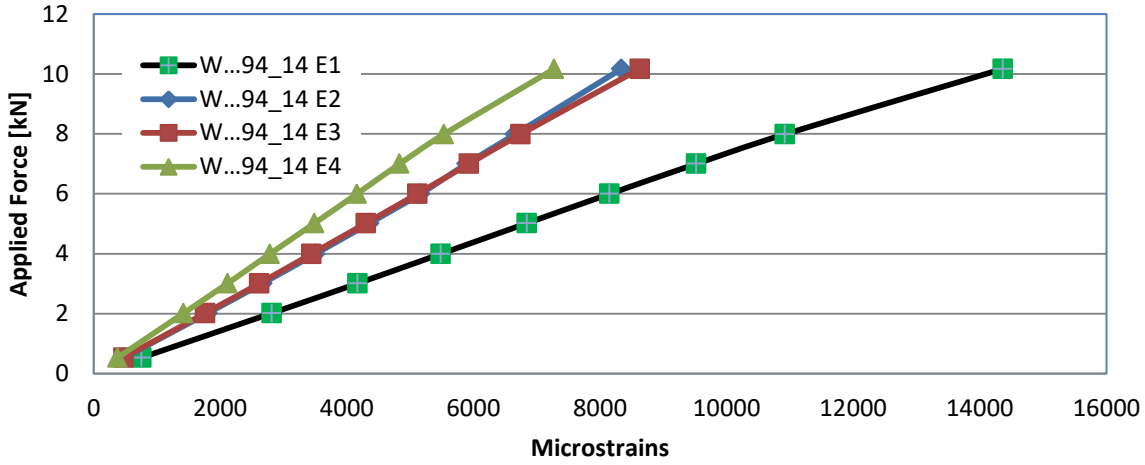
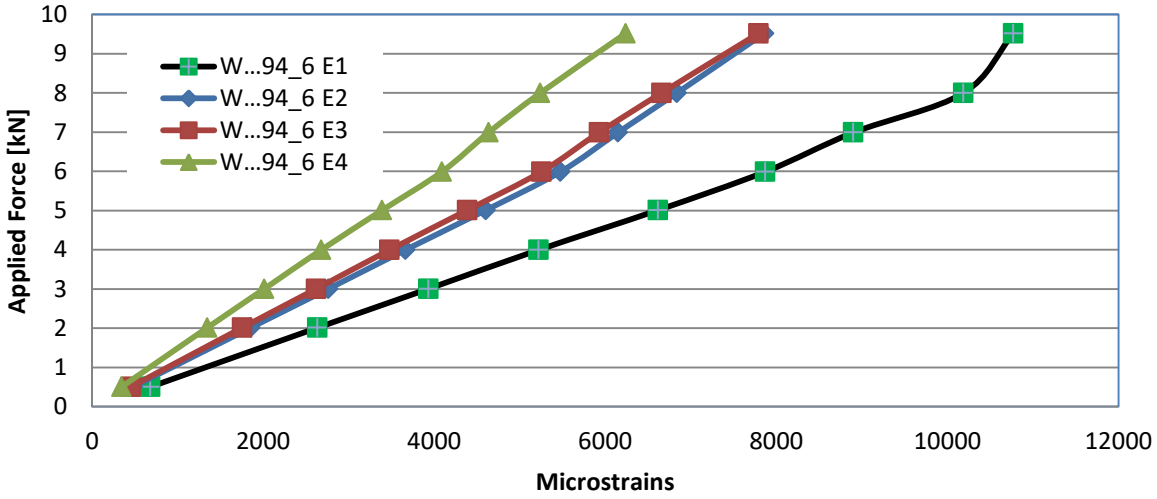


Figure 4.3 Strain gauge positions on the repaired facing sheet

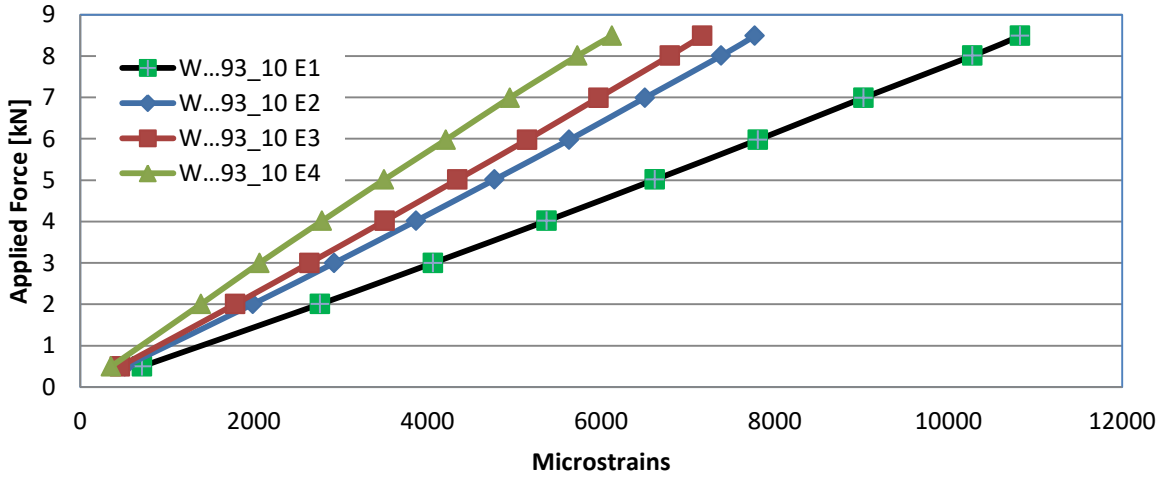
Strain E1, which was on the top surface of the original tool facing sheet, showed the highest strain values. This location was one of the most likely failure point for the specimens. Strain E2 was on the surface of the 0^0 repair ply and measured lower strain values because the 0^0 repair ply has significantly increased the stiffness of the tool facing sheet. Strain values at E3 were almost similar to the strain values at E2 because the strain gauge at E2 was pasted at a location which was strengthened by the 45^0 repair ply. This strengthened the tool facing sheet less than the much stiffer 0^0 repair ply hence the small difference between the E2 and E3 strain values. The strain values at E4 were lower than E2 and E3 strain values due to the additional stiffness of the second 0^0 repair ply. By comparing the measured strain values the critical failure position of the facing sheet was at X=285 mm (location of strain gauge E1). This was confirmed by the strain measurement shown in Figure 4.4 (a to c). The strain values at E1 was further used in obtaining the elastic modulus of the facing sheet. Since the strain values at the location of E1 are maximum and also critical for the sandwich beam failure behaviour, the local strain values (E1) were used to obtain the elastic modulus of the facing sheet.



a) Strain measurements from specimens from W94_14



b) Strain measurements from specimens from W94_6



c) Strain measurements from specimens from W93_10

Figure 4.4 Parent ply and repair ply strains of the sandwich beam experimental results

The facing sheet stresses with the strains at E1 were plotted for the three specimens as shown in Figure 4.5. As expected the facing sheet stresses and the measured E1 strain exhibited a linear relationship, which was then used to obtain the Young's modulus of the facing sheet as given in Table 4.2. The calculation of facing sheet stresses used the mathematical formula introduced in section 3.2.6. A linear relationship between the facing sheet stresses and strains was observed.

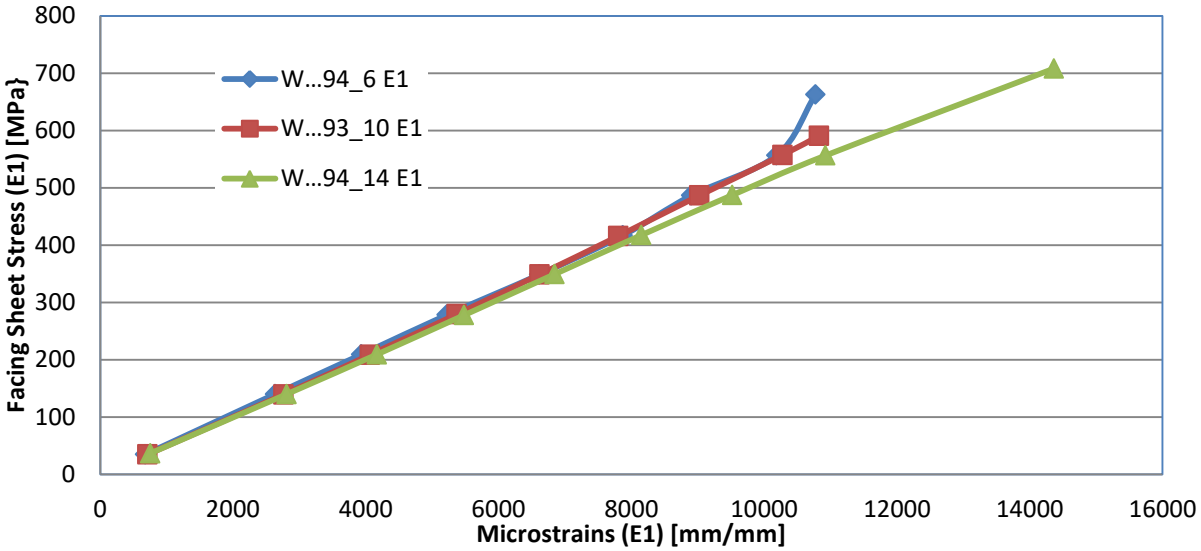


Figure 4.5 Experimental strains versus parent laminate stress at position E1

Table 4.2 Tool side facing sheet laminate experimental material properties results

Specimen	**Laminate Property** E [MPa]	
W94_6	55.8	
W94_14	50.4	
W93_10	53.7	
	Mean E	53.3 MPa
	STD DEV	2.7 MPa

The results of the calculated facing sheet stresses and measured strains were used to calculate Young's modulus (elastic modulus) of the original tool facing sheet by a linear regression fit in Microsoft Excel. Equation 4.1 from the regression fit shows that the slope or Young's modulus (mean) is 53.3 GPa and given in Table 4.2.

$$\sigma = 53.3 \times \varepsilon \tag{4.1}$$

Figure 4.6 shows the A and B (strain gauge E1 position) critical points where the specimens failed. These points were the least reinforced locations of the repaired facing sheet with olnly three plies and the bending moment was also highest within this section of the specimens. This cause the highest facing sheet stress at this locations.

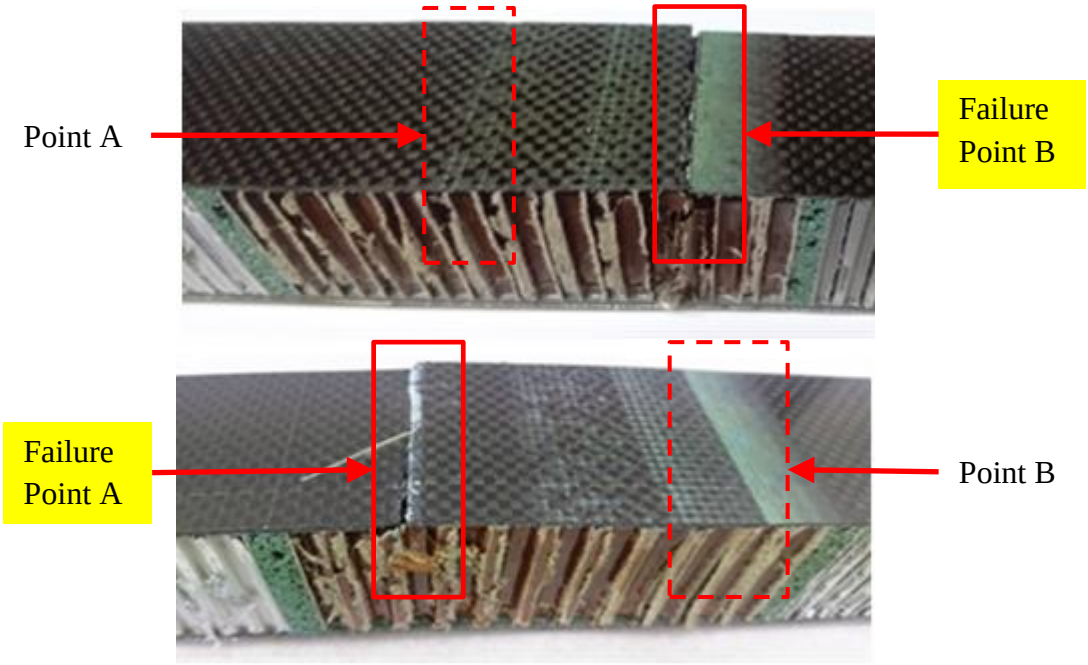


Figure 4.6 Tool facing sheet failure areas for experimental specimens

4.2 Validation of Static Experimental Results using MsM Analysis

The experimental results were validated using the MsM analysis. The results were validated for the three length scales (mirco, meso and macro scale). At firsrt the measured strain values using strain guage were validated using the macro scale model where the strains (E1, E2, E3 and E4) were extracted. Figure 4.7 shows the strain measurement from the MsM analysis. By comparing E4 and E1 strain values for an applied load of 9 kN the accuracy of the MsM approach can be determined. At 9 kN the experimental strain was 6000 microstrains for specimens W94_14 and W94_6 and the MsM strain was 6000 micro strains as well. E1 strains on the other hand were around 12000 microstrains for the same load. These MsM strain results closely matched two of the three specimens in Figure 4.4.

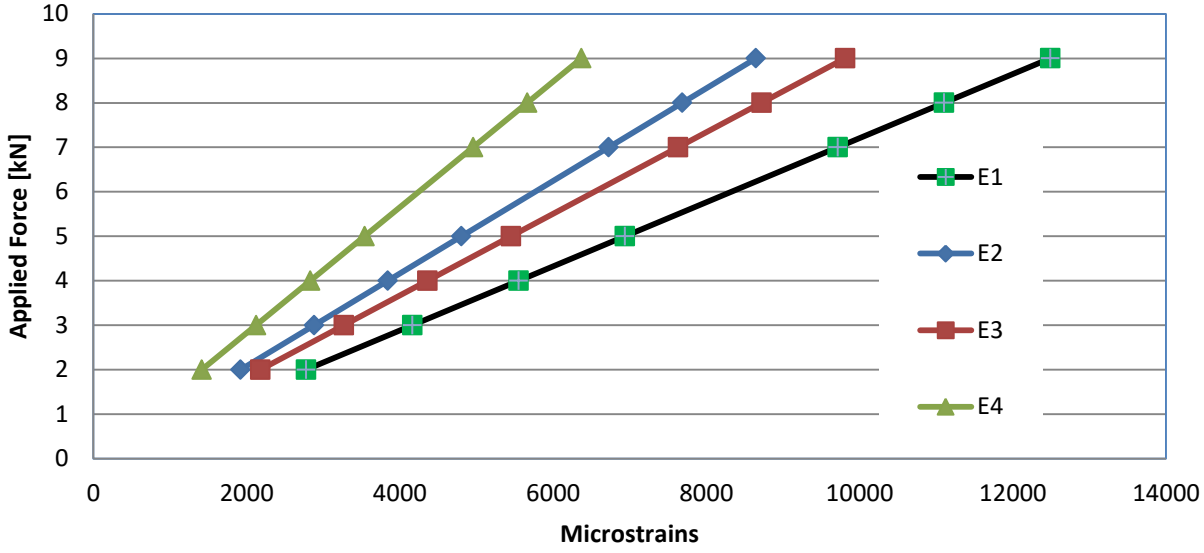


Figure 4.7 Load versus strains from macro model (MsM)

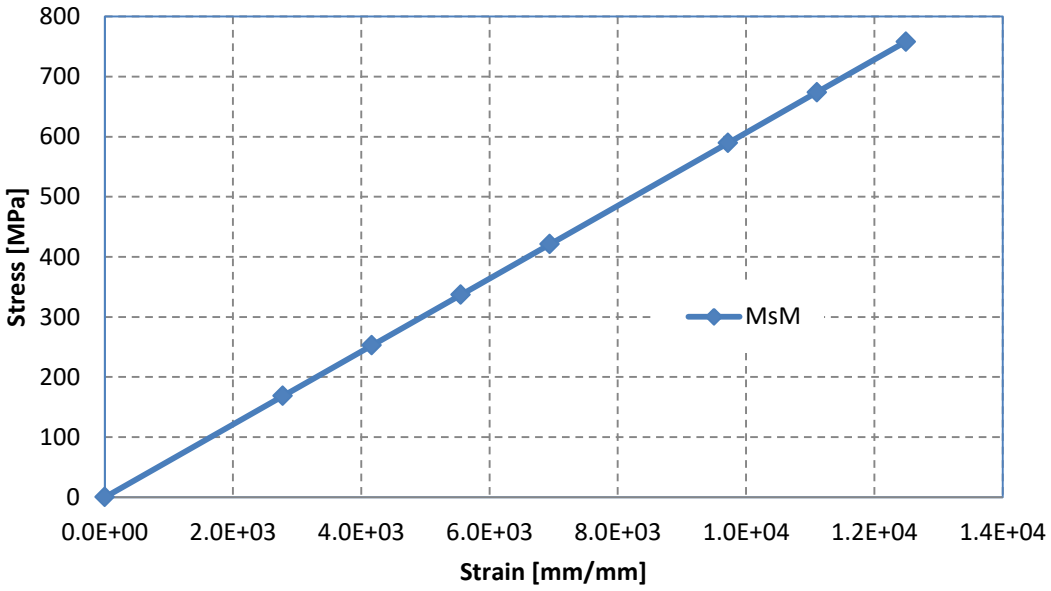


Figure 4.8 Macro model stress versus strain at critical point B (macro element 7688)

Strains and stresses from critical point B in the MsM macro model were used to plot a stress versus strain curve as shown in Figure 4.8 for comparison purposes to Figure 4.5.. Facing sheet elastic moduls was determined from the slope of the MsM stress against MsM strain curve. The resulting tool facing sheet elastic modulus value is higher than the experimental Young's modulus as shown in Table 4.3. This difference is due to using the outer layer stress for the MsM stress and the average stress for the experimental stresses values.

Table 4.3 Tool side facing sheet laminate MsM material properties results

Laminate Property	Experimental
E [MPa]	60 700

4.3 Static Failure Analysis

As explained in the experimental results, the MSM macro model analysis also showed the critical failure points at locations A and B as shown in Figure 4.9. Since the macro model was derived from the micro and meso models, the critical failure points at the macro scale evolved from the micro scale to meso scales and led to the failure of the macro model. Based on the above assumptions, the critical failure behaviour at points A and B were analysed from the micro scale, meso scale and finally macro scale.

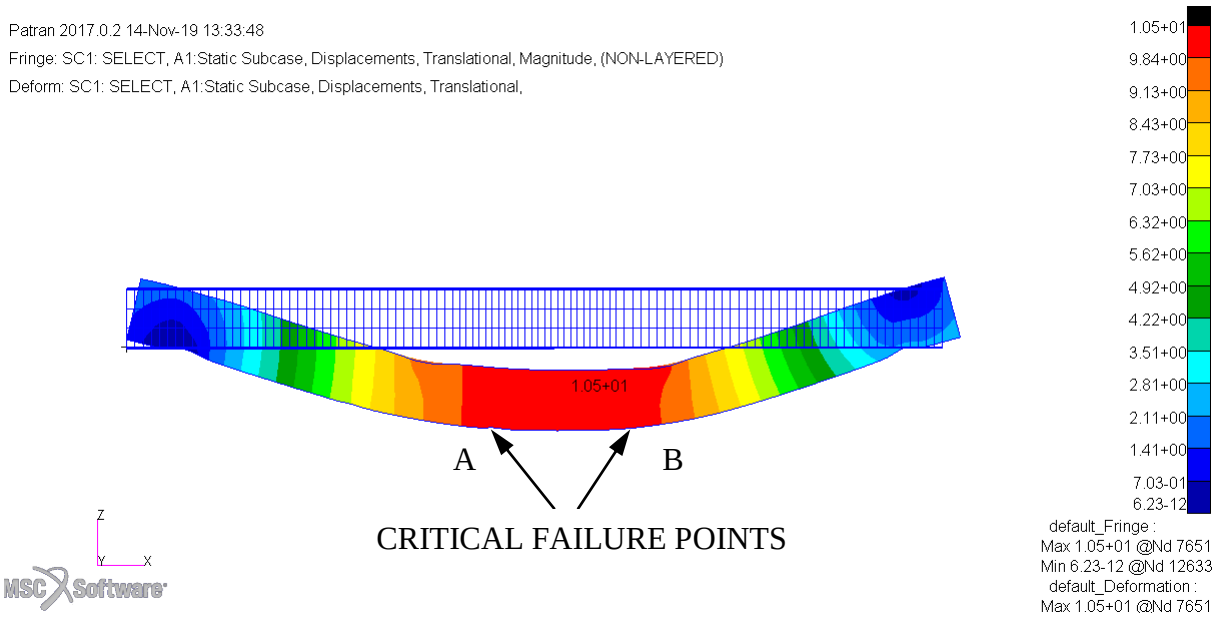


Figure 4.9 Critical points A and B on the repaired facing sheet

4.3.1 Experimental Micro Scale Failure

Figure 4.10 and Figure 4.11 show the scanning electron microscope (SEM) images of the micro scale matrix and carbon fibre failure. The matrix and carbon fibres failed due to the four-point bending load applied perpendicular to the fibre as shown in Figure 4.10 and Figure 4.11. These failure behaviour is typical of tensile loaded facing sheets. The matrix failed first before fibre failure, which showed that there was a load transfer between the fibre and the matrix within the repaired scheme, before a complete micro scale failure occurred. Moreover, the matrix failed at multiple locations before transferring the load to the fibres. At the critical points, the fibres fractured in a straight line across the failure surface, which confirmed that the fibres failed in a brittle manner. There were clean fibre surfaces showing signs of interface failure between the fibre and matrix as shown in Figure 4.10. Interface failure was a result of shear loads transferred from the matrix to the fibres. In conclusion the fibres and the matrix failed due to applied tensile loading. The matrix failed completely before the failure of carbon fibres at the micro scale.

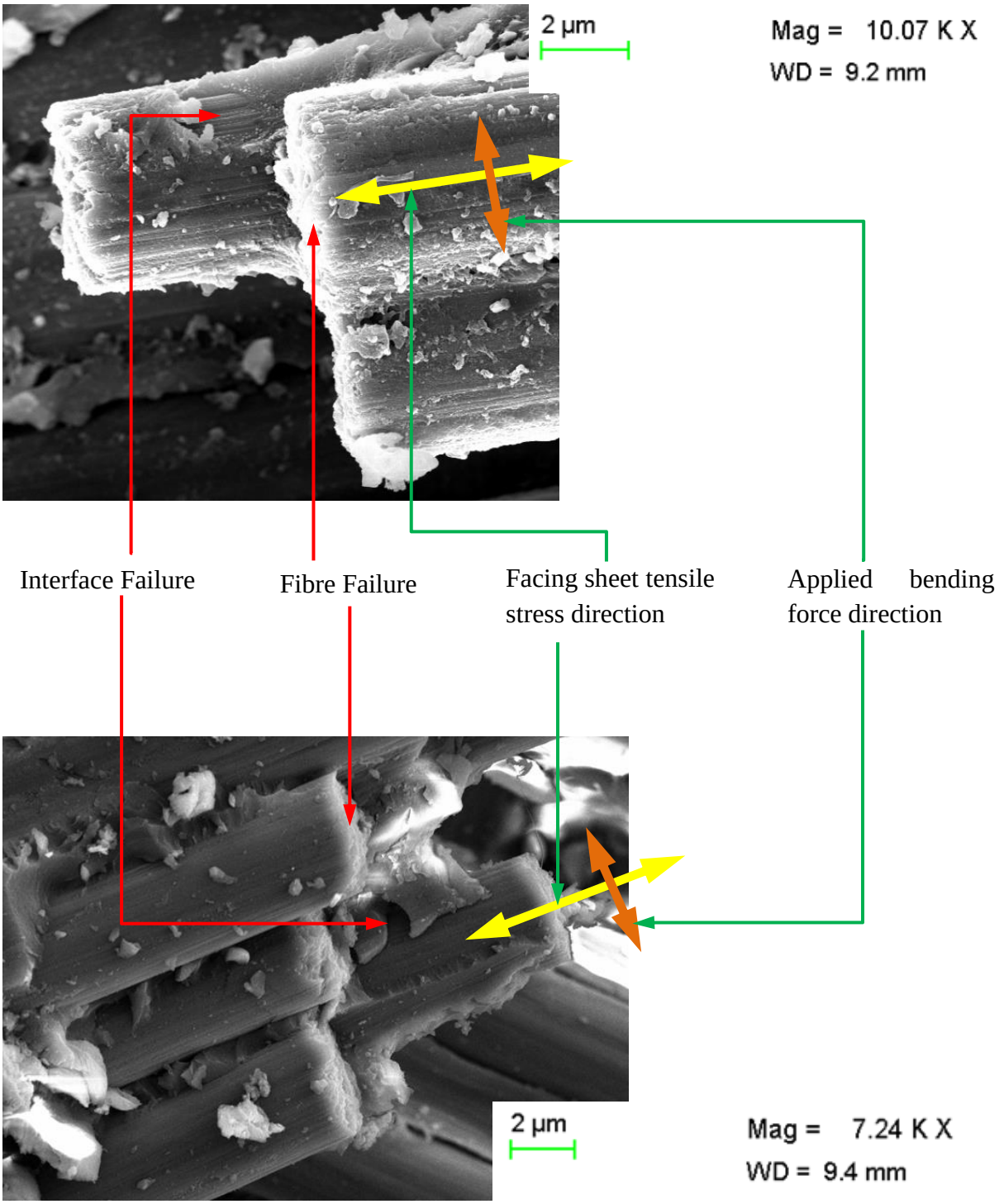


Figure 4.10 SEM pictures of static micro scale failure at critical points A (Warp Tows)

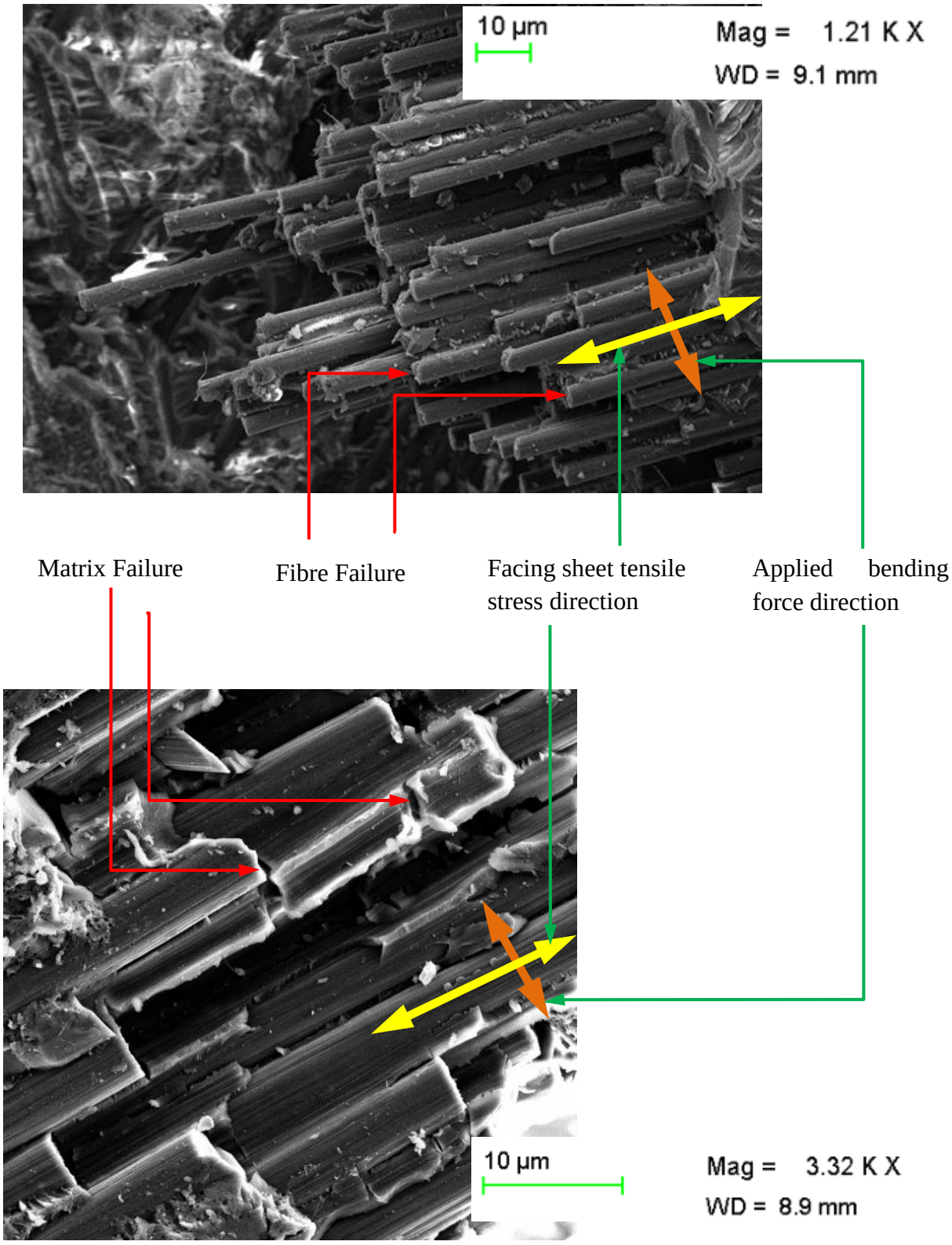


Figure 4.11 SEM pictures of static micro scale failure at critical points B (Warp Tows)

4.3.2 MsM Micro Scale Failure

Figures Figure 4.12 shows the Von Mises stress plot for the micro RVE matrix and fibres elements for an applied bending load of 3 kN. This load was choosen so that the stress distribution within the micro RVE model could be studied. Figure 4.13 shows the elemental stress distribution of the matrix and fibres seperately due to the same applied bending load of 3 kN. The results show that the stress values for the matrix are maximum at the top, bottom, left and right faces of the RVE due to the transfer of shear loads into the fibres. There is also a large diffrenece between the minimum and maximum matrix stresses. This indicates that the matrix will fail gradually as its elements reach their failure stress at different load levels. The fibre elements on the other hand have a very small difference between the minimum and maximum stress. This implies that the fibre elements will reach their fauilure stress all at the same time and fail in a brittle manner. No variation in stress values is observed for both the matrix and fibre elements in the X-direction.

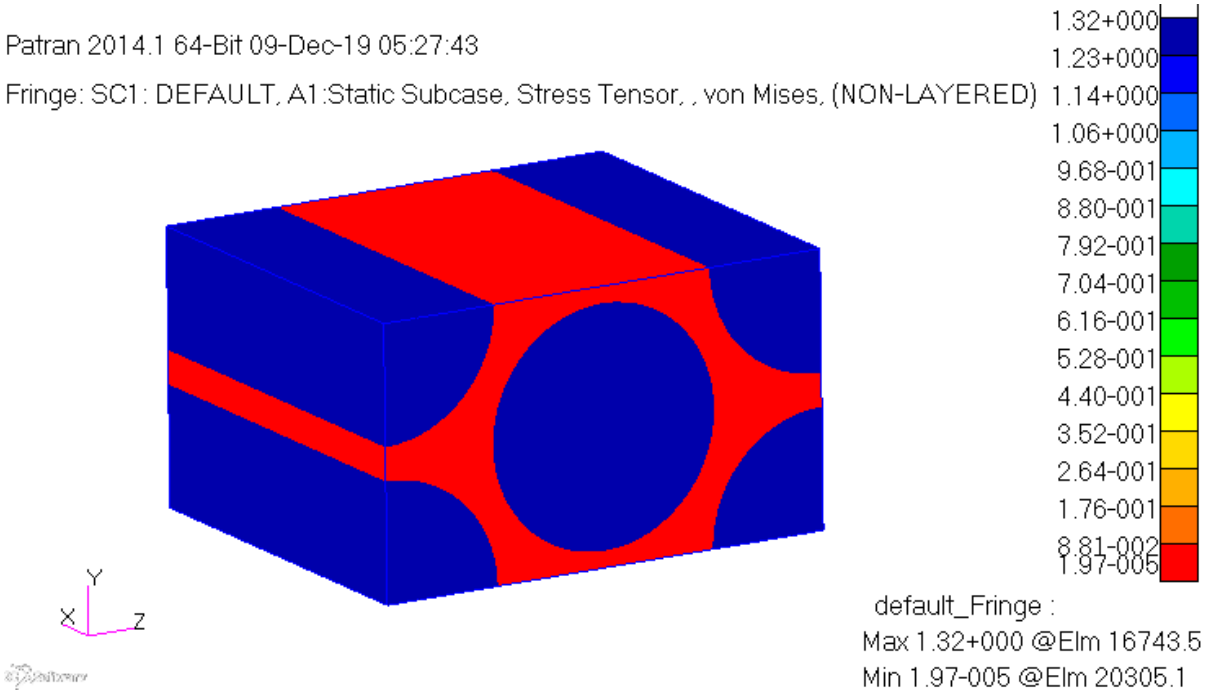
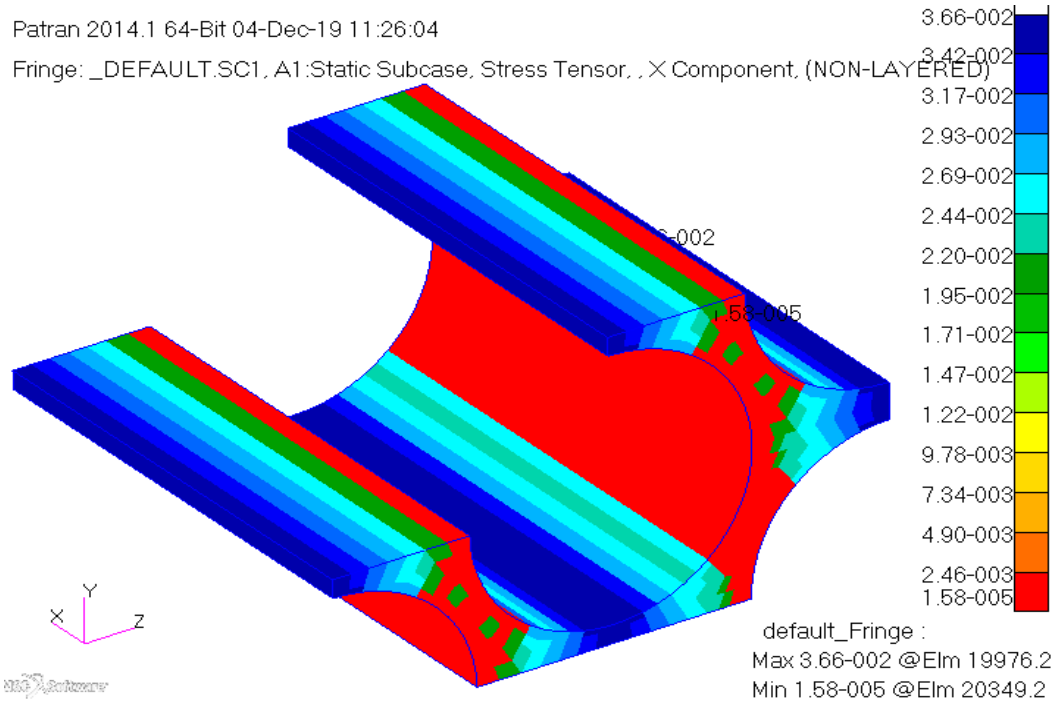
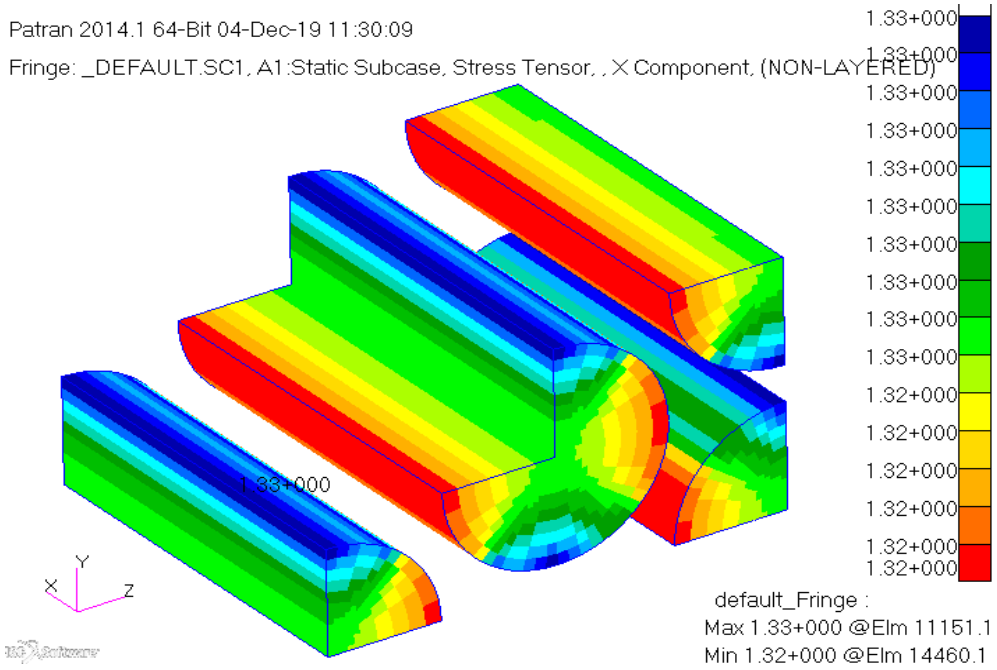


Figure 4.12 Von Mises stress plot of the micro model (GPa)



[a] Tensile stress plot of the matrix



[b] Tensile stress plot of the matrix

Figure 4.13 Tensile stress plot of the micro model matrix and fibres

The stress results from Figure 4.12 and Figure 4.13 were also reproduced as graphs of X, Y and ZX element stresses for two perpendicular paths going through the RVE centre in the YZ plane. The resultant graphs are shown in Figure 4.14 and Figure 4.15. X and Y stress components are

tensile stresses while the ZX is a shear stress. These figures show that the Y and ZX stress components are very small compared to the X stress component across the micro RVE. There is also a significant increase in the X stress component at the fibre and matrix boundary. The diference in the tensile stresses between the matrix and the fibre is due to their differences in stiffness. The stiffer fibres carry most of the tensile load in the facing sheet. However the matrix maximum X stress of 0.0366 GPa is not far from the matrix failure stress of 0.045 GPa. Meaning that the matrix will fail first as the applied macro load increases. This is in agreement with the observation from experimental results. A small variation in the Y stress component is observed at the matrix and fibre boundary since this component is related to the X stress component by Poission's ratio. ZX stress component is invariant across the micro RVE and this makes it the main component that could cause interface failure at the matrix and fibre boundary.

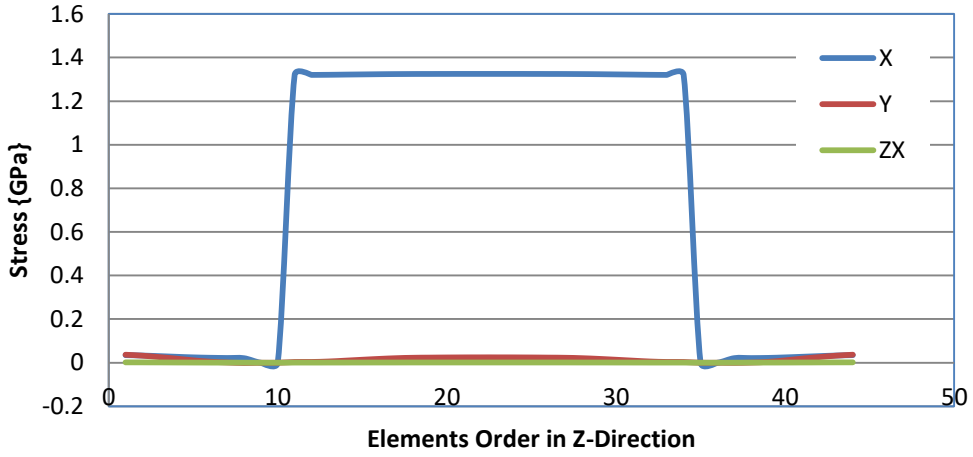


Figure 4.14 Element centroid stress components along Z-dir arrow

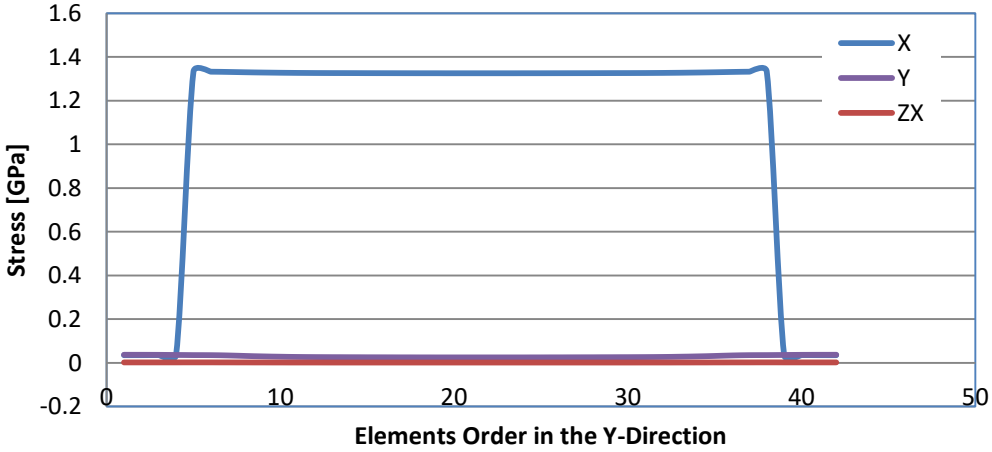
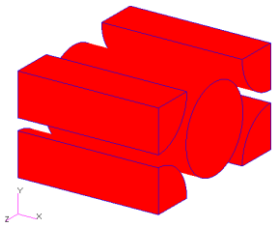


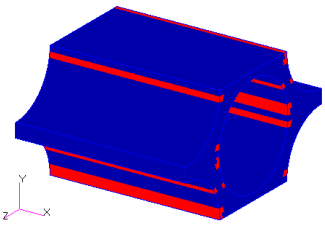
Figure 4.15 Element centroid stress components along Y-dir arrow

The micro scale failed elements results were generated with MPC displacements from meso element 6630 (strains) in the meso RVE model. This warp tow element position in the meso RVE model will be marked by a yellow circle in Figure 4.28. The progressive failure of the micro model can be studied closely using the MsM micro model. Figure 4.16 below shows the degree of

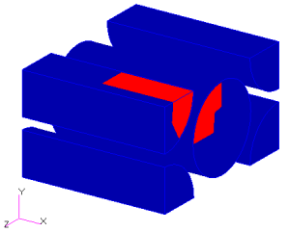
failure of the matrix and fibre elements as the applied macro load increased. In the following MsM results, the colour blue is for elements that have failed and red elements are intact elements. From no load to an applied load of 8.02 kN, the fibre elements did not fail but the majority of the matrix elements had already failed. Thereafter the matrix and fibre elements fail completely at the next load increment of 8.52 kN. At the next load increment of 8.76 kN, the micro elements failure profile does not change as the micro RVE has softened to the point where it could not carry additional load for the selected meso scale element strain.



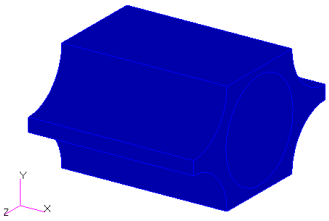
Warp fibre elements at 8.02 kN



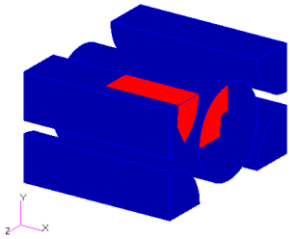
Warp matrix elements at 8.02 kN



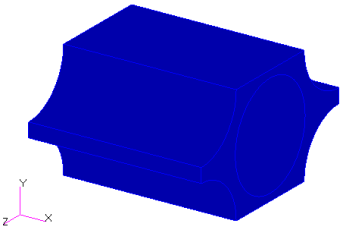
Warp fibre elements at 8.52 kN



Warp matrix elements at 8.52 kN



Warp fibre elements at 8.76 kN



Warp matrix elements at 8.76 kN

Figure 4.16 Static damage to micro RVE as a result of various loads (meso element 6630)

These results show that micro scale failure start in the matrix rather than the fibres in the facing sheet composite material. With the MsM approach it was possible to incrementaly apply load and study the failure of the micro scale model. To study micro scale failure experimentally one needs sophisticated and expensive equipment which was not available fot this experiment.

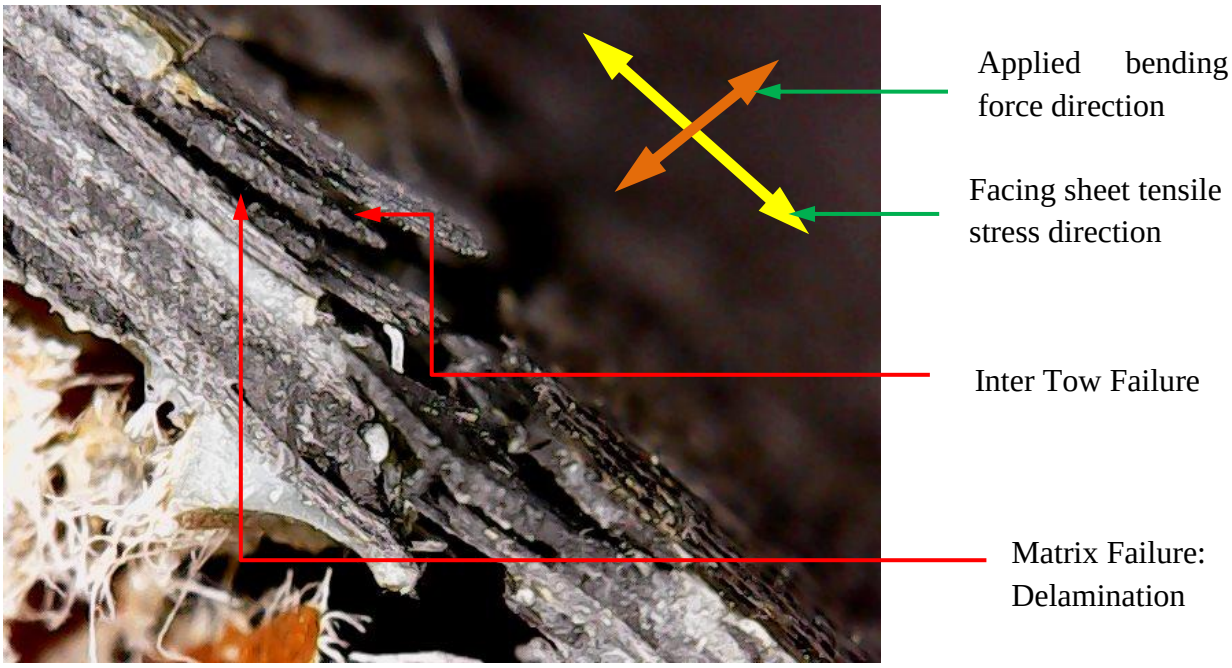
4.3.3 Experimental Meso Scale Failure

Since the meso scale consists of woven tows and matrix, the failure analysis within the meso scale was focused on analyzing the failure of the woven tows and matrix. The fibres in the micro scale are aligned with the warp tows in the meso scale and warp tow are aligned with the X direction of the macro scale model. Similar to the micro scale analysis, the matrix within the warp and weft tows failed first before leading to warp and weft tows failure. The warp tows failed by rupturing

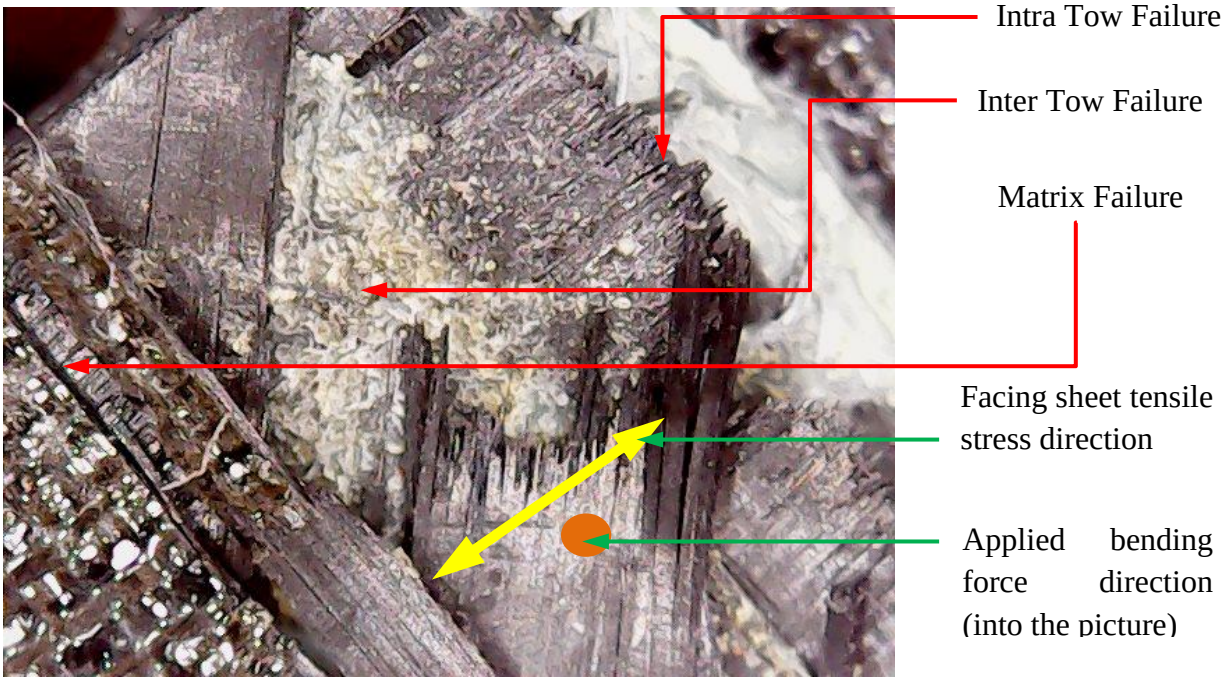
and weft tows failed by splitting. Warp tow rupturing is a combination of fibre and matrix failure and weft tow splitting is the failure of the matrix within the tows or around the individual fibres. Warp tow rupturing results from failure of the fibre and matrix at the micro scale due to a tensile stress in the direction of the fibres (micro and meso scale X-directions). Tensile failure of fibres can be seen in Figure 4.10 and Figure 4.11 and a collection of failed fibres create tow rupturing as seen in Figure 4.18 (a). Weft tow splitting is caused by matrix failure as a result of tensile stress perpendicular to the fibres direction as shown in Figure 4.18 (b). Weft tow splitting will be called intra-tow failure and the separation of weft and warp tows will be called inter-two failure. Inter-tow failure between tows belonging to adjacent plies leads to delamination. Warp tow rupturing occurs where the warp tows rise over or dip under the weft tows. The rupturing at the rises or dips of the warp tows is due to additional bending and shearing as a result of change in tow direction. Intra-tow failure of weft tows is more pronounced where the warp tows also fail, that is at the rises or dips of warp tows.

Figure 4.17 (a) shows delamination and inter-tow failure while

Figure 4.17 (b) shows isolated intra-tow failure and matrix failure. Figure 4.18 (a) shows tow rupturing while Figure 4.18 (b) show intra-tow failure. Figure 4.18 (b) shows matrix failure between the fibres, which is interpreted as inter-tow failure. These multiple forms of failure are all exhibited close to the failure line across the tool facing sheet. A simple tension load in the tool facing sheet sets off a complex mixture of different forms of failure. This behaviour shows that plain woven tows and matrix composites exhibit complex failure modes under tensile and shear stresses.

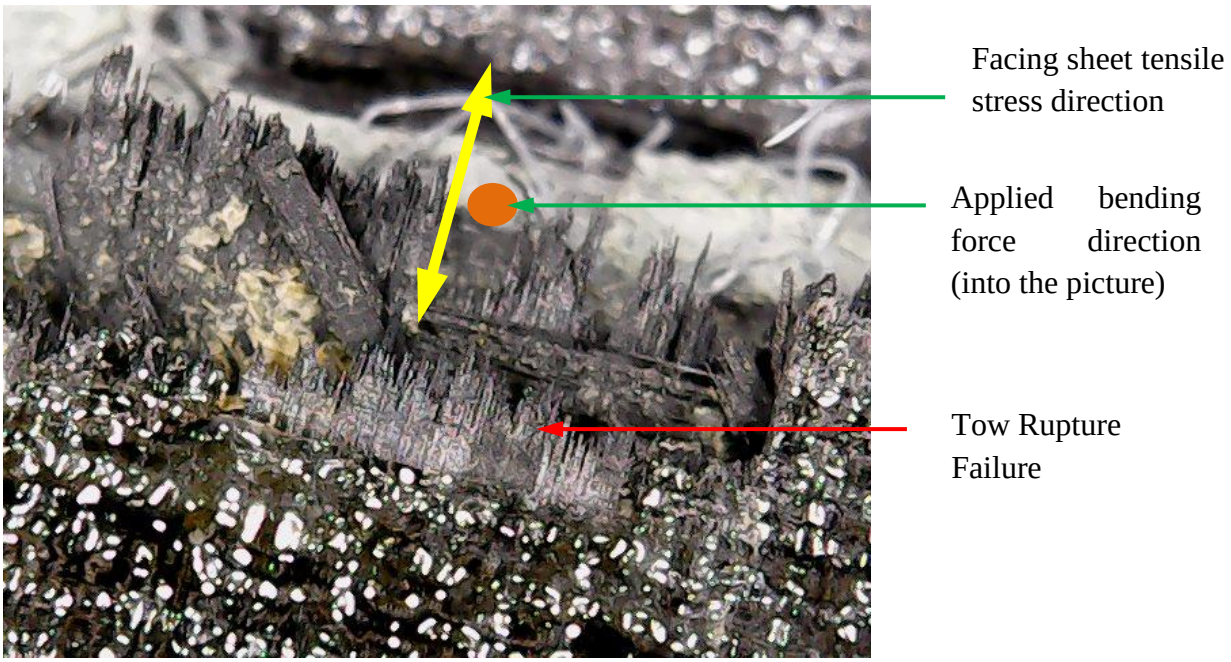


(a) Ruptured laminate delamination and inter-tow failure

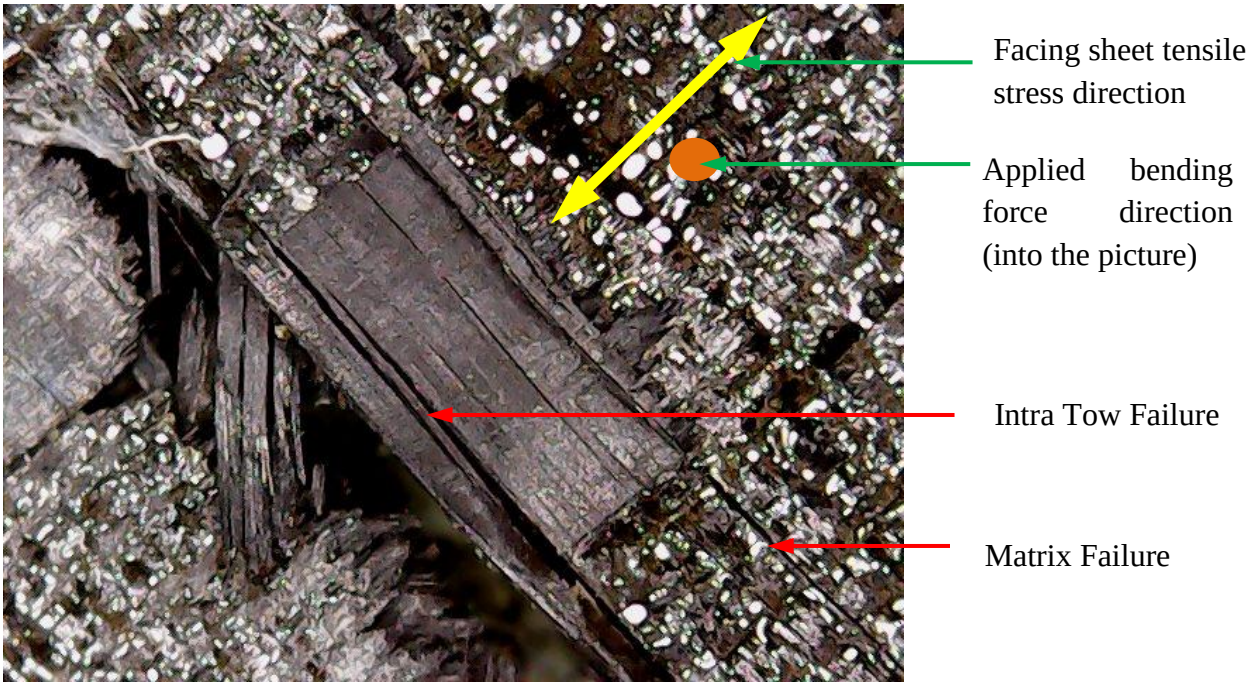


(b) Inter-tow and matrix failure

Figure 4.17 Ply delamination failure at critical point B



(a) Warp-tow rupturing across its cross-section



(b) Weft-tow splitting across its width

Figure 4.18 Warp tow rupture and weft tow splitting failures at critical point B (handheld microscope pictures)

4.3.4 MsM Meso Scale Failure

Figure 4.19, Figure 4.22 and Figure 4.25 show the stress plots of the warp tows, weft tows and matrix elements for an applied bending load of 3 kN. Warp tow element stresses in the X-direction reach a peak at the intersection points of warp and weft edges. The difference between the maximum and minimum stresses is large at 200 MPa. Amongst the meso RVE elements the warp tow elements have the highest stresses because they are stiffer followed by the weft tow elements. This is the reason for only analysising the warp tows at the micro scale. Warp tow failure occurs at their intersection points with weft tows and across the warp tows as was observed in the experimental results of Figure 4.18 (a). From the stress plot failure starts at the intersection points and develops across the width of the warp tows.

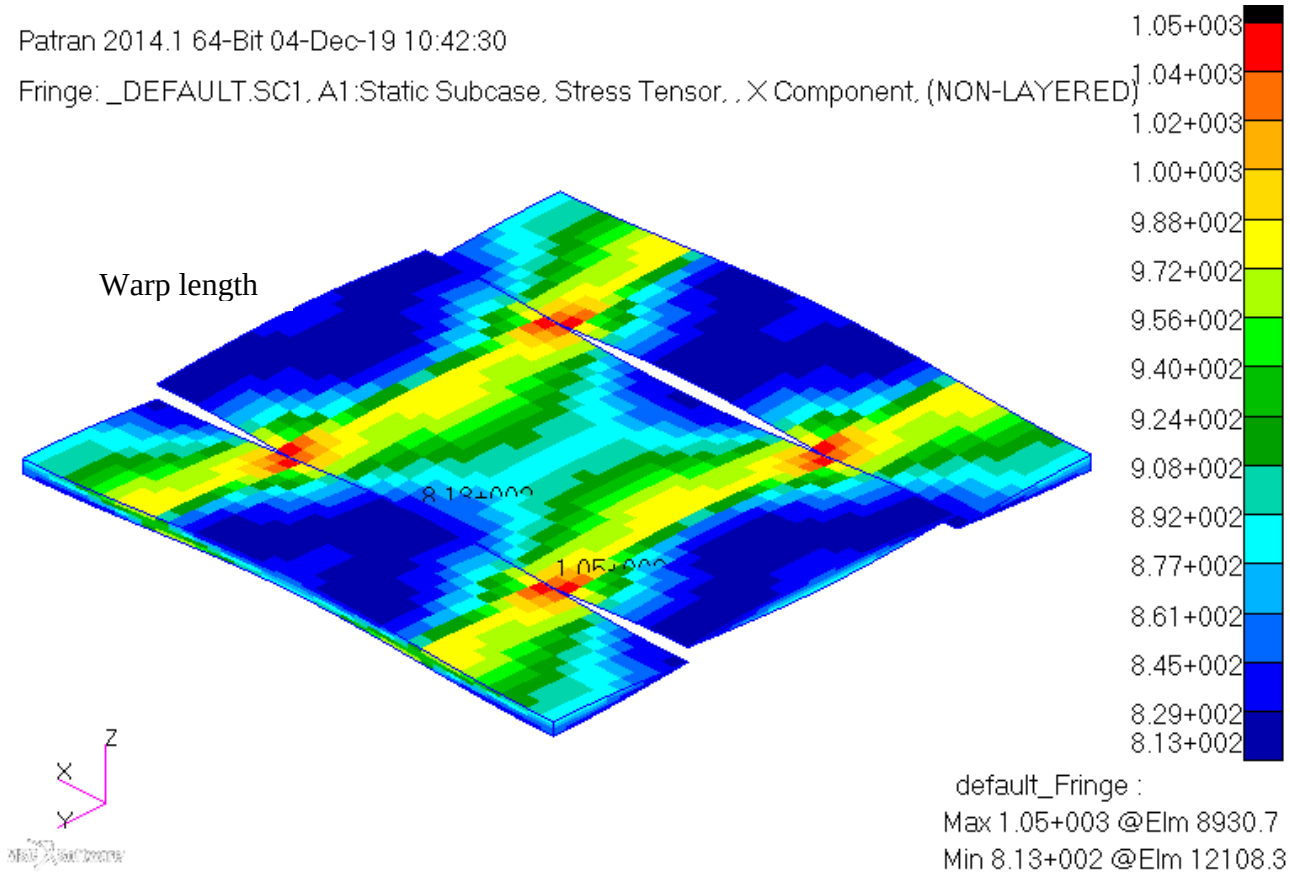


Figure 4.19 Element list positions on a stress plot of the meso model warp tows

The results of the stress plots of Figure 4.19 and Figure 4.22 are reproduced as graphs of X, Y and ZX stress components in Figure 4.20, Figure 4.21, Figure 4.23 and Figure 4.24 for two perpendicular paths along the length of the warp tow and across their width. The graphs across the warp width are positioned across their highest loaded points. For the warp tows Figure 4.20 and Figure 4.21 show that the X stress component is the dominant component when compared with the Y and ZX stress components. The X stress component also reaches its peak at the edges of the warp tows and is the tensile stress. There is insignificant variation of the Y and ZX stress components along the length and across the width of the warp tows.The Y stress component is also a tensile stress and the ZX component is a shear stress. The focus of the warp tows MsM analysis should therefore be on the X stress component with the warp tows.

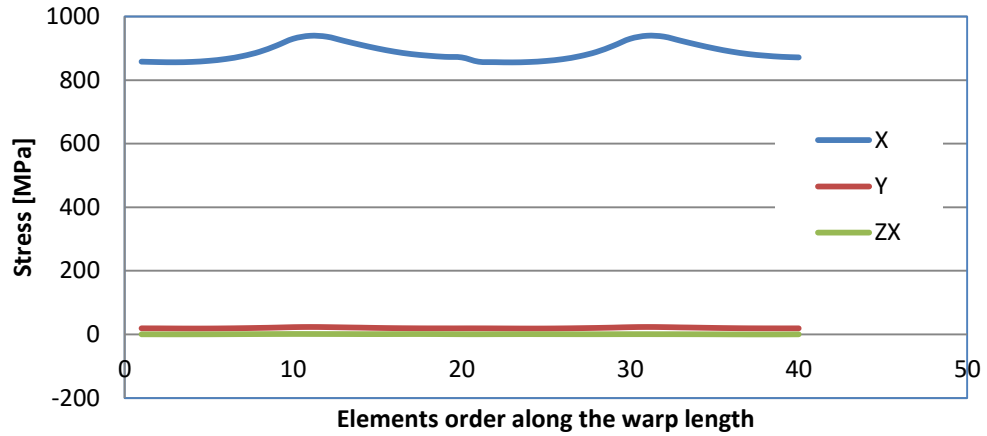


Figure 4.20 Element centroid stress components along the warp length

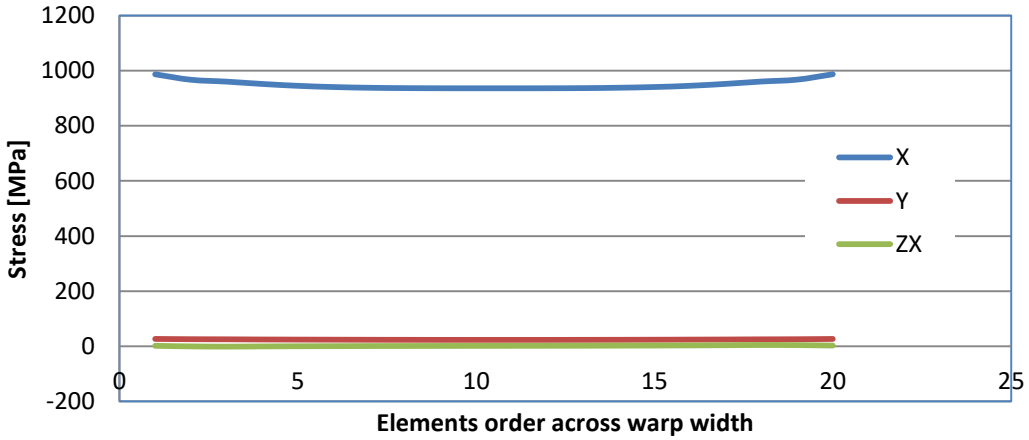


Figure 4.21 Element centroid stress components across the warp width

The edges of the weft tows are the most highly stressed areas for the weft tow elements group. Similar to the warp tow elements the highest stresses are at the intersection points of the weft tow edges and warp tow edges. These points have no matrix between the warp and weft tows as a result the meso RVE develops its highest stiffness at the points. The high stiffness means that the meso model carries a lot of its stresses at these points. MsM weft tows failure mostly occurs at the weft tow edges as noted in the experimental results of Figure 4.18 (b).

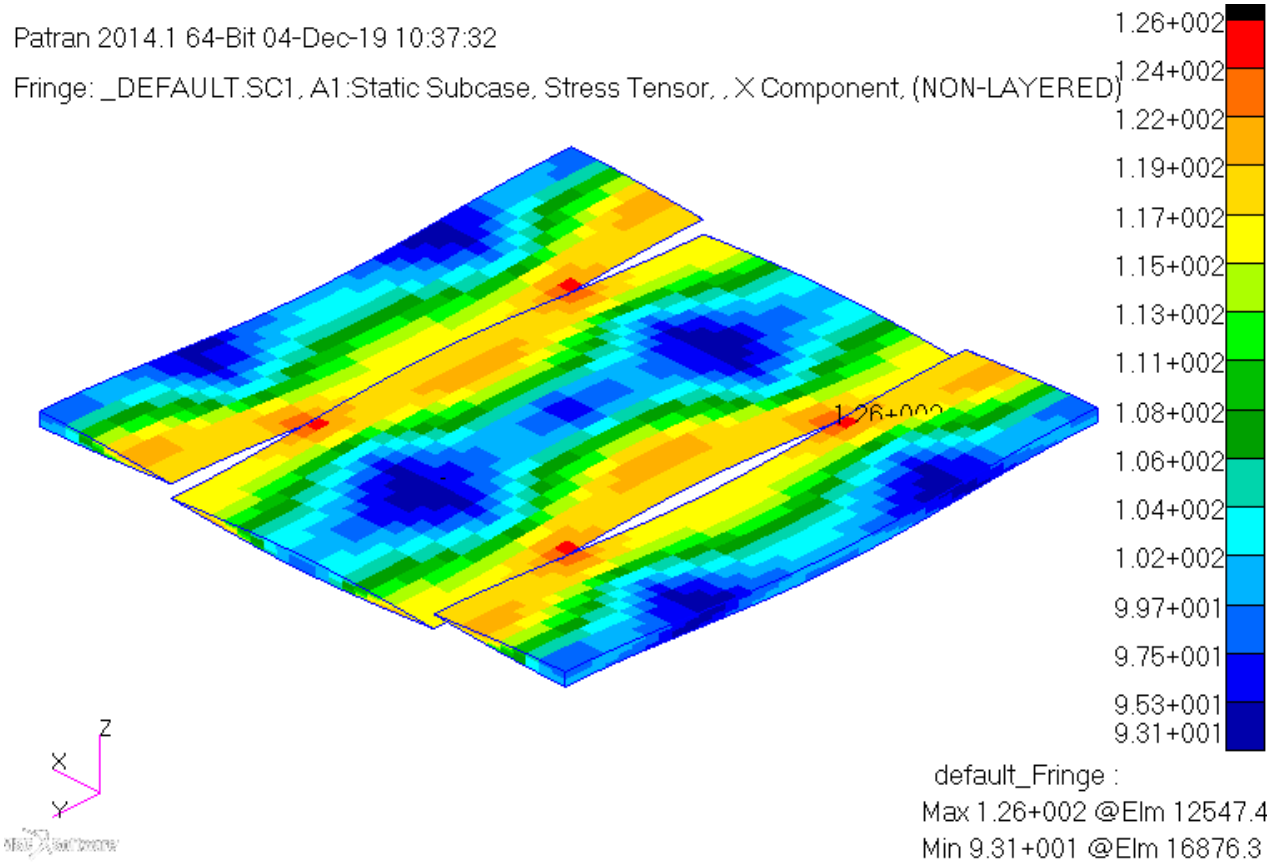


Figure 4.22 Element list positions on a stress plot of the meso model weft tows

Figure 4.23 and Figure 4.24 present the variation of the weft tow X, Y and ZX stress components in the weft length and weft width directions. The X stress components is a lot smaller than the same component from the warp tow. Relative to the Y and ZX stress components the X stress component still dominates but the Y stress component is more than half the X component in some elements. Although there is no load applied in the Y direction the weft tow has high stiffness in this direction. The influence of Poisson's ratio and high stiffness in the Y direction are strong enough to induce stress values that are almost half of the X stress component.

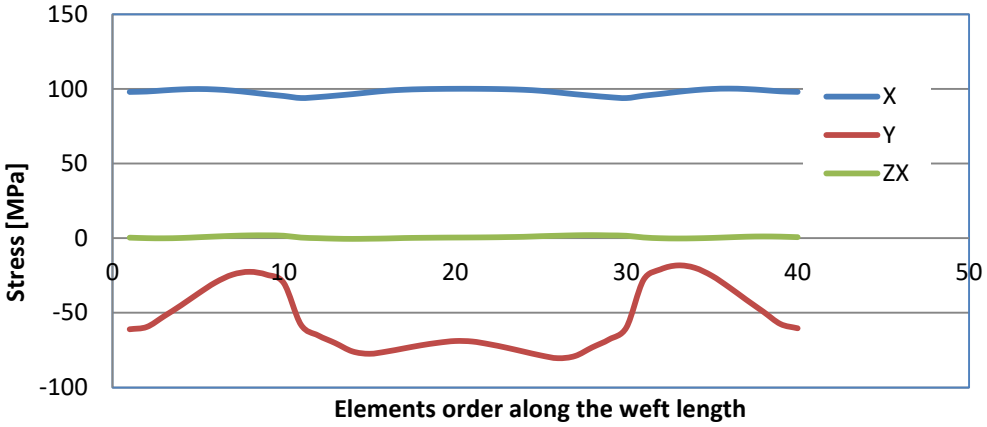


Figure 4.23 Element centroid stress components along weft length

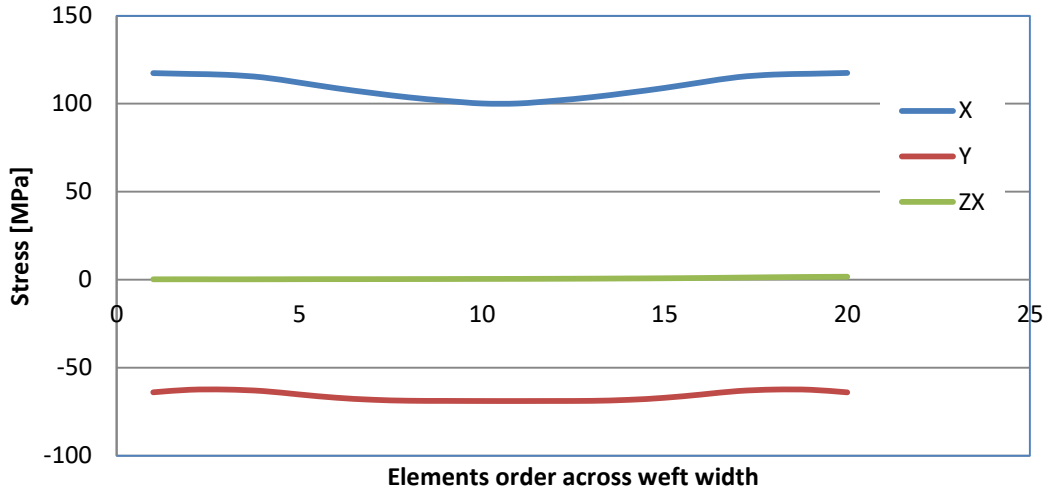


Figure 4.24 Element centroid stress components across the weft width

The lowest element stresses are from the meso scale matrix elements but it is also the least strong material in the meso model. Similar to the micro scale failure trend, the matrix fails first because at an applied bending load of 3 kN the peak matrix stress is 29 MPa. This value is already close to the matrix tensile strength of 45 MPa. Matrix failure occurs in straight lines in the weft direction as shown by the high stress elements in Figure 4.25. This confirms the observations made in the experimental results in Figure 4.17 and Figure 4.18.

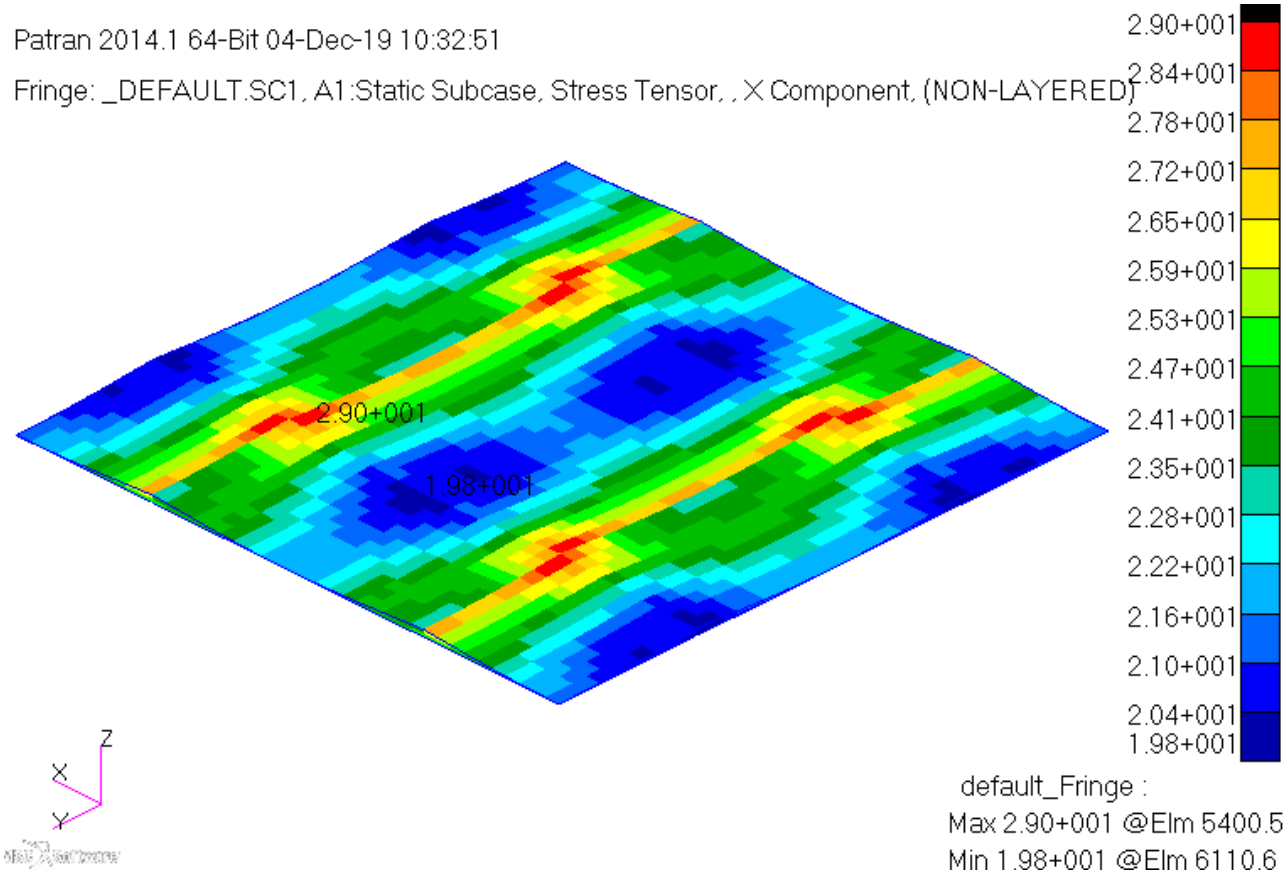


Figure 4.25 Element list positions on a stress plot of the meso model matrix

The stress plot results of Figure 4.25 are reproduced as graphs of X , Y and ZX stress components in Figure 4.26 and Figure 4.27. These graphs are for element paths in the weft and warp directions of the meso RVE. These figures show that the X stress component still dominates the other components in the meso scale model. The X stress component is a tensile load in the matrix hence meso scale failure lines are perpendicular to the X direction.

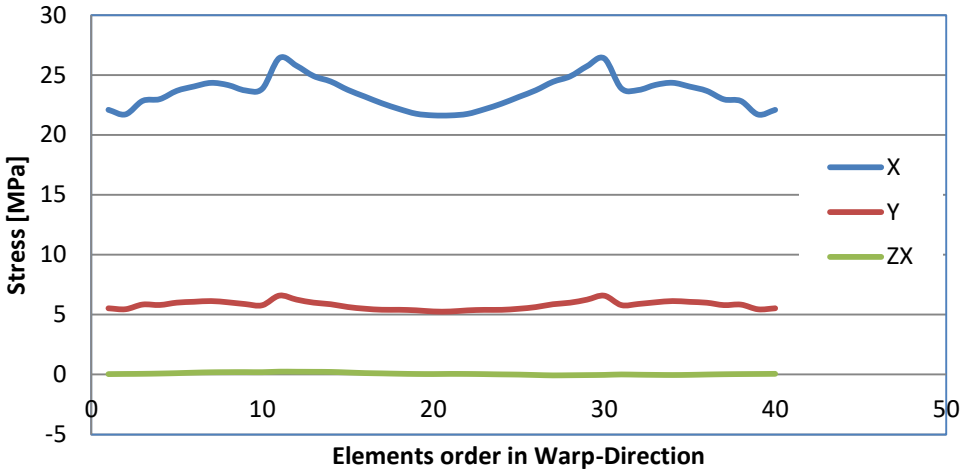


Figure 4.26 Element centroid stress components along warp length

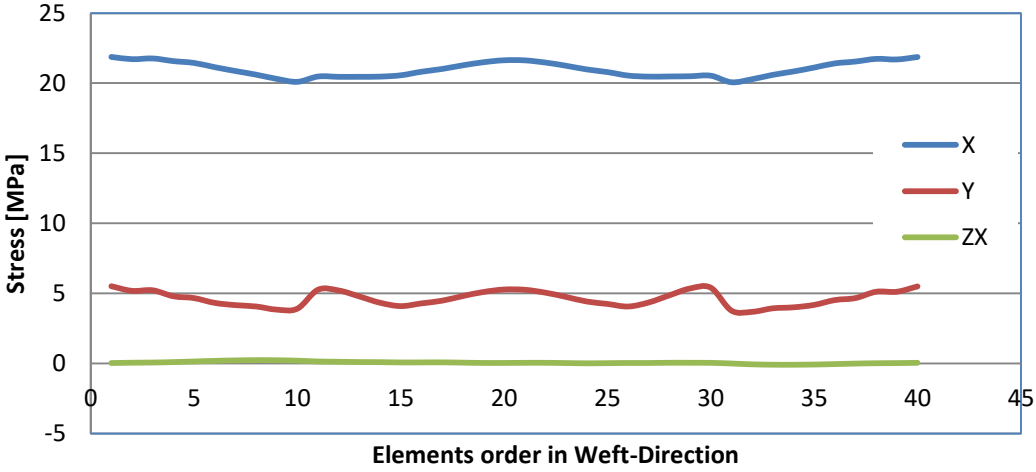
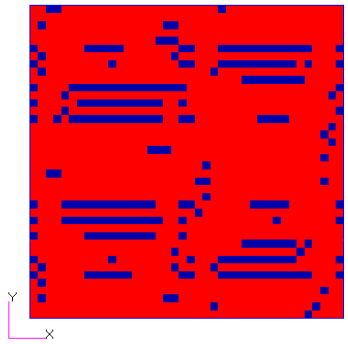
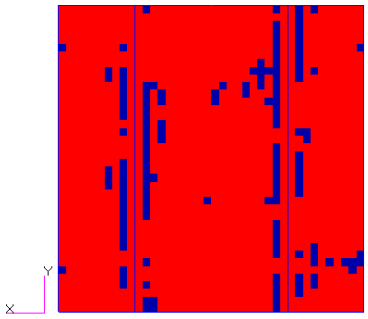


Figure 4.27 Element centroid stress components along weft length

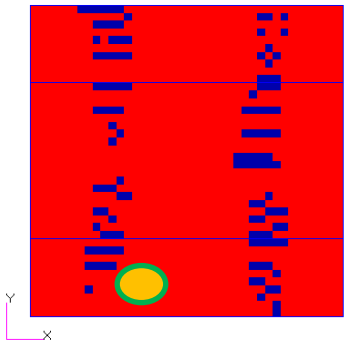
The meso scale failed elements results were prepared with MPC displacement values from the macro element 7686 strains at critical point A. At an applied load of 8.02 kN the majority of matrix elements failed before the warp and weft tow elements. These results are shown in Figure 4.28 below. The meso matrix elements exhibited little progressive failure as the load increased. Some warp and weft tow elements failed at 8.02 kN but the majority of the matrix elements failed at the next load increment. The matrix elements do not show any progressive damage failure because there is very little variation in the element tensile stress values in the warp and weft directions. Once the matrix element tesnsile stress values are above 45 MPa they all suddenly fail. The mode of failure for the warp tows was similar to what was observed in the experimental meso scale results in in Figure 4.17 and Figure 4.18. The failed warp tow ends were serrated as shown in Figure 4.29. The weft tows were dominated by intra-tow failure as shown in Figure 4.30. The meso model predicted a top ply failure of 9.02 kN which is not far from the experimental average of 9.96 kN.



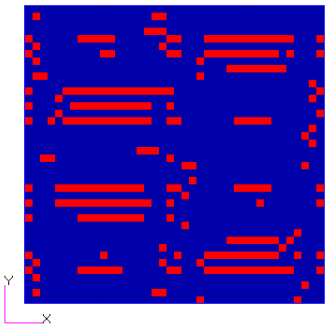
Matrix elements at 8.02 kN



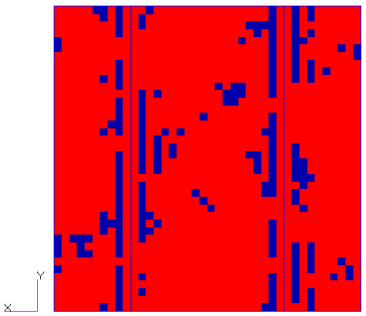
Weft tows elements at 8.02 kN



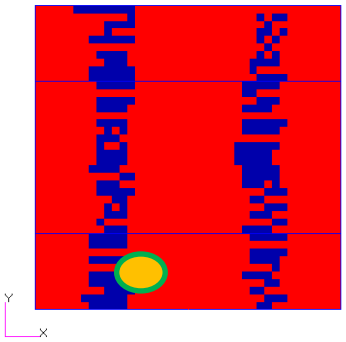
Warp tows elements at 8.02 kN



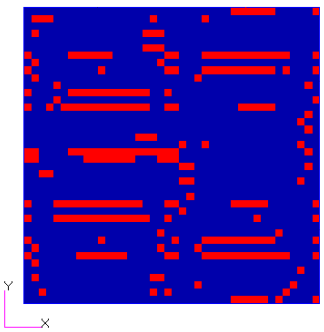
Matrix elements at 8.52 kN



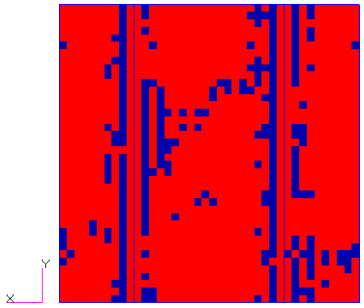
Weft tows elements at 8.52 kN



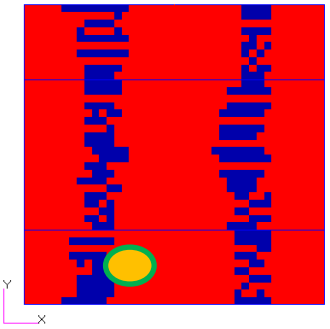
Warp tows elements at 8.52 kN



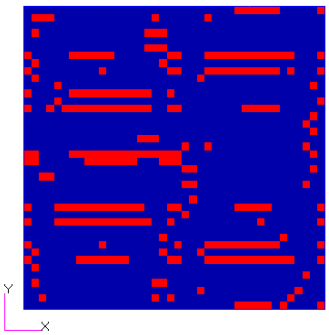
Matrix elements at 8.76 kN



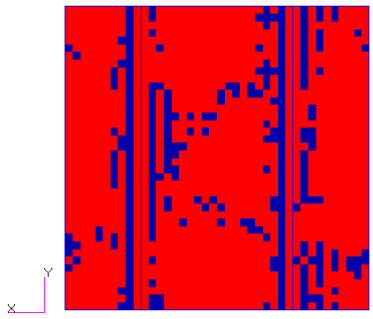
Weft tows elements at 8.76 kN



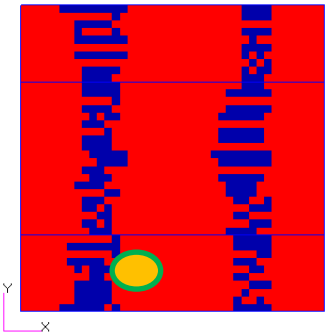
Warp tows elements at 8.76 kN



Matrix elements at 9.02 kN



Weft tows elements at 9.02 kN



Warp tows elements at 9.02 kN

Figure 4.28 Static damage to meso RVE as a result of various loads (macro element 7686)

Figure 4.29 shows that the warp tow elements failure at the point where the warp tows rise over or dip under the weft tows in the MsM and experimental results. The missing warp tow elements in the MsM image are failed elements. The change in warp tow direction induces bending and shear loads in addition to the applied tensile loads. This change in direction creates high stress points that lead to failure by warp tow rupturing.

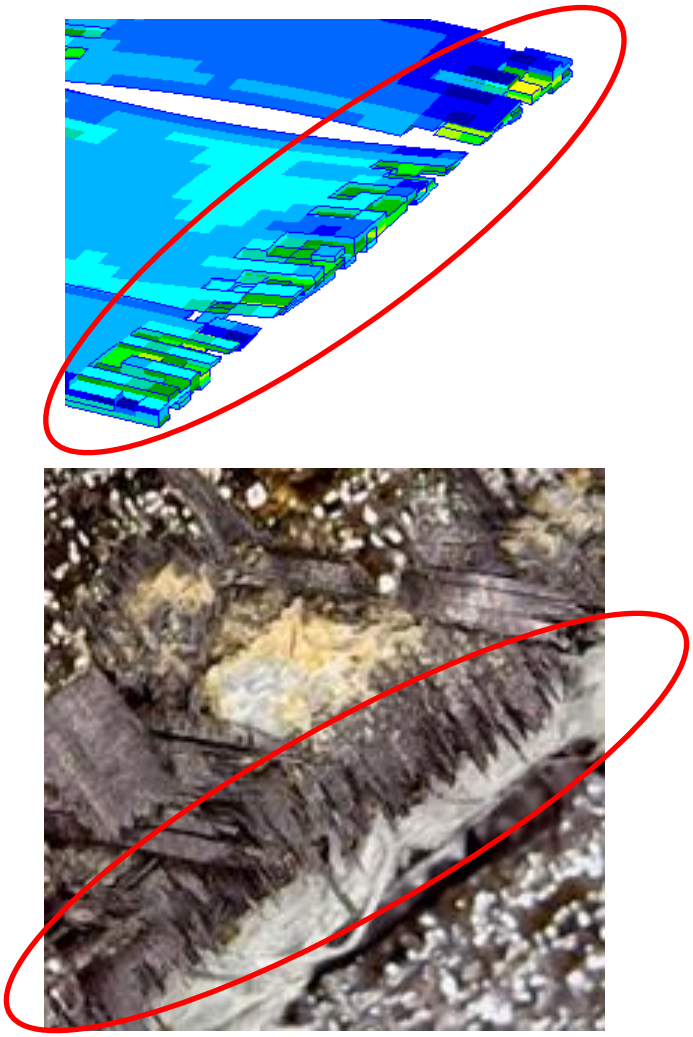


Figure 4.29 Meso scale warp tow failure edge for MsM and experimental specimens

Figure 4.30 shows results of weft tow failure along the lengthwise edges from the experimental and MsM results. The experimental camera image shows failure along the lengthwise edges of the weft tow and the MsM image shows highly stressed weft tow elements in red and failed element were removed from the picture. The high stress points on the weft tows are a result of warp tows tranfering loads to the weft tows as they rise over or dip under the weft tows.

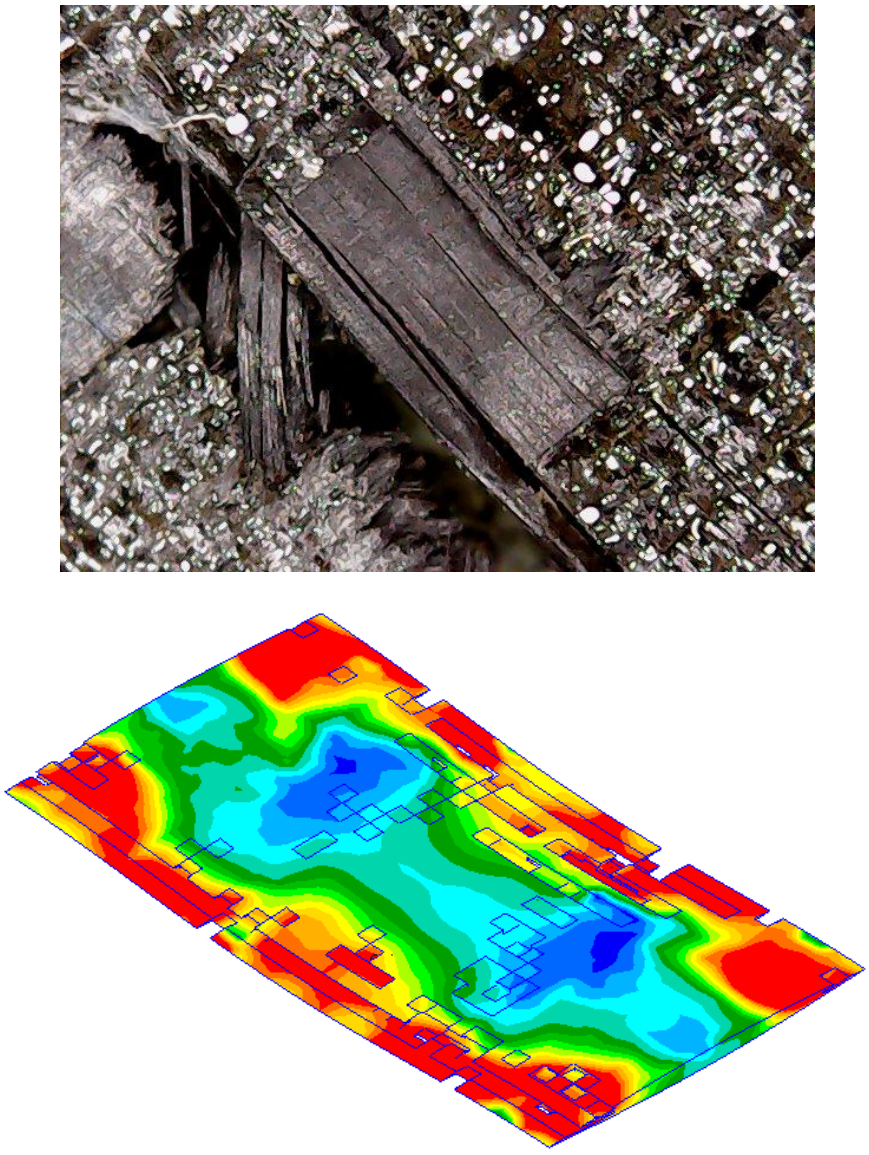
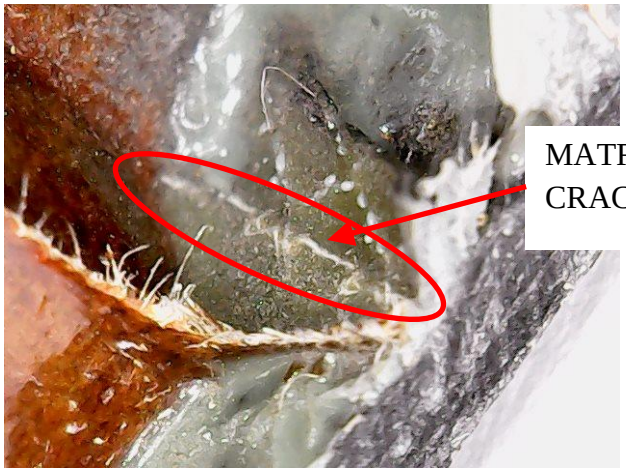


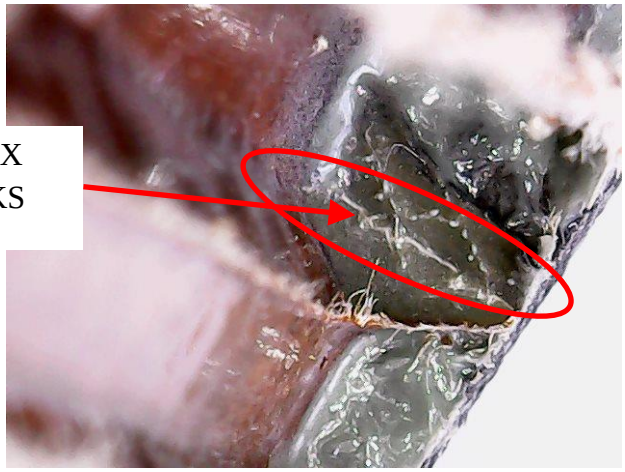
Figure 4.30 Meso scale weft tow failure morphology for MsM and experimental specimens

4.3.5 Experimental Macro Scale Failure

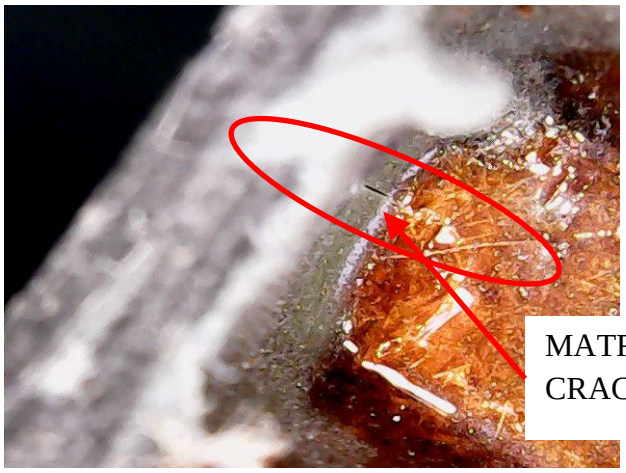
At the macro scale, two critical areas of failure were observed on the specimens. At a thickness of three plies these areas were the repaired facing sheet thinnest points. Another factor that made them crititcal was that they were in the highest bending moment region of the four-point bending experiment. These areas were identified as critical point A (CP-A) where the repaired plies failed and critical point B (CP-B) where the parent laminate plies failed. The critical points are highlighted in Figure 4.9. Point A is on the repair plies and point B is on the original laminate plies. Firstly, for critical point A, the matrix cracked in the inner surface of the repaired tool facing sheet. Matrix cracking occurred in this location since the facing sheet had only three load carrying repair plies over the joint of parent and filler plies. The plies at the macro scale failed as the warp tow were ruptured and weft tows split in a straight line across the repaired facing sheet. Warp tows ruptured and weft tows split due to repaired facing sheet tensile stresses in the repaired facing sheet. For applied macro loads from 6 kN to 10 kN, the cracking was observed along the Nomex core edge of the filler plies. Matrix cracking at applied load levels below 8 kN suggests that it precedes repair plies failure at critical point A. Figure 4.31 (a) and (b) show the matrix crack in its initiation stage at a load level of 6 kN. Figure 4.31 (c) and (d) show the crack opening just before facing sheet failure at applied loads higher that 8 kN. This means that the matrix crack opens up as the applied load increases. At repaired facing sheet failure load levels between 8 kN and 10 kN, the repair plies exhibited rupturing. This failure is shown in Figure 4.32.



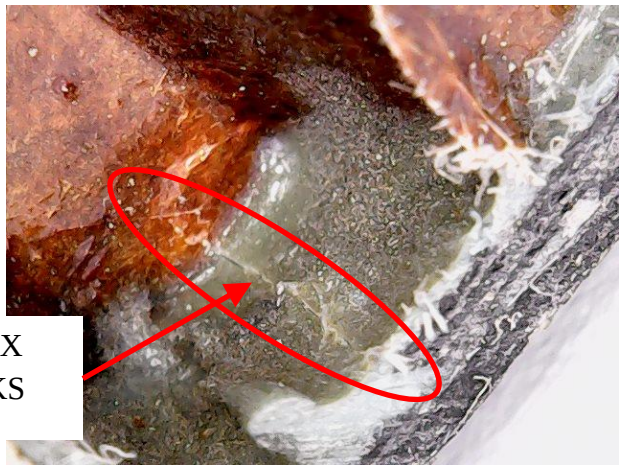
(a) Macro matrix cracking at 6 kN



(b) Another macro matrix cracking at 6 kN



(c) Macro matrix cracking at 8 kN



(d) Macro matrix cracking at 8 kN

Figure 4.31 Matrix failure at the macro scale at critical point A (handheld microscope pictures)

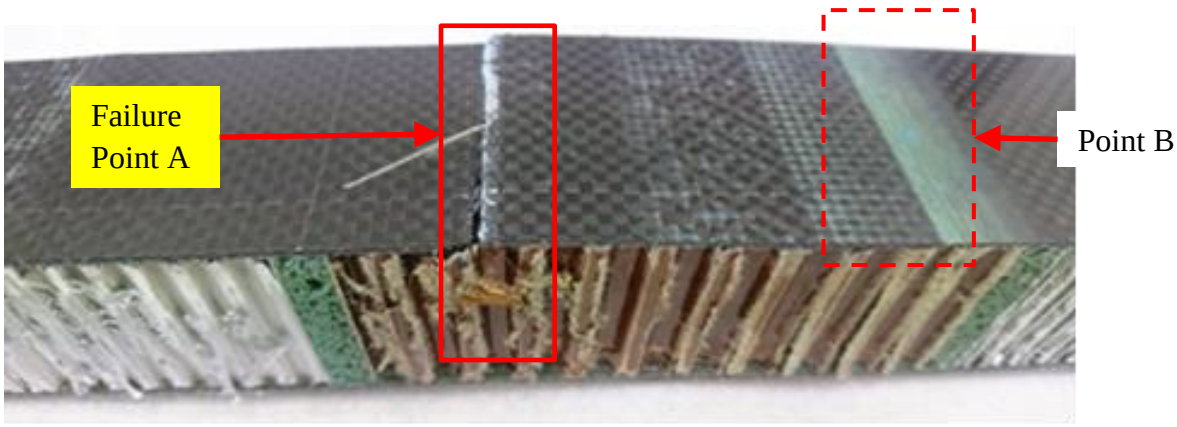


Figure 4.32 Macroscale: Tool facing sheet (repair plies) rupture line **(Critical Point A)**

Secondly, for critical point B, no cracking was observed prior to the failure of the parent laminate at the edge of the 0^0 repair ply. At applied loads between 7.4 kN and 10.2 kN, the original tool facing sheet exhibited rupturing across the repaired facing sheet. The failure at critical point B is shown in Figure 4.33. Again the facing sheet failed at this point because the tensile load was carried by only three plies.

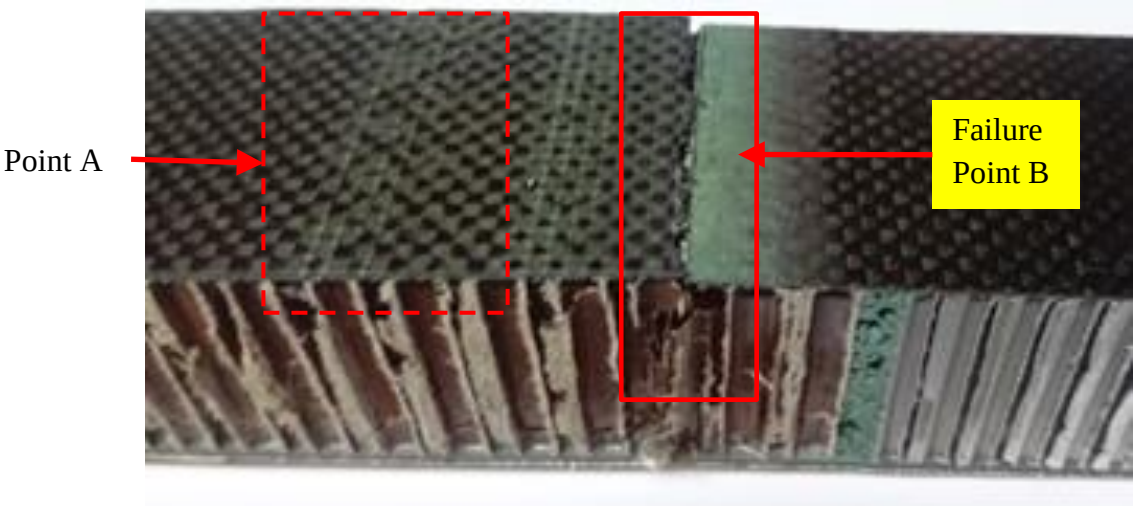


Figure 4.33 Macro scale: Tool facing sheet (parent laminate) rupture line **(Critical Point B)**

4.3.6 MsM Static Macro Scale Failure

Results from the MsM macro model also confirmed the criticality of the two failure locations as shown in Figure 4.6. These locations were over the Nomex core section of the sandwich beam. They are indicated by the two red strips on the facing sheets in Figure 4.34. Peak macro scale strains for the meso RVE MPC displacement loading were selected from these locations. The critical points A and B correlate well with peak macro scale strains for experimental specimens.

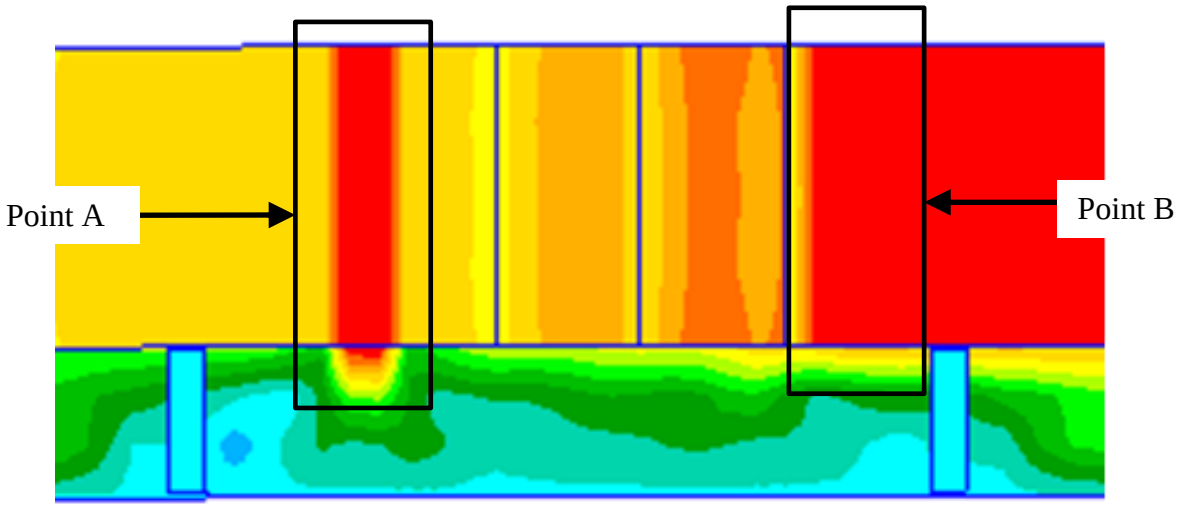


Figure 4.34 Tool facing sheet failure areas for MsM and experimental specimens

Comparing the MsM and experimental results lead to the following observations for the the macro scales. At the macro scale failure in the repaired tool facing sheet occurred in two critical areas marked as point A and B. The failure morphology was a straight line across the facing sheet as demonstrated by the experimental results pictures and highlighted by MsM results pictures in Figure 4.34. Both failure points are within the maximum bending moment region of the sandwich beam specimens. The bending moment can be resolve into couple forces that cause a tensile stress in the repaired facing sheet. The tensile stress was in the same direction as the warp tow direction. The straight line facing sheet failures occured when the three plies thick laminates reached their tensile stress strength. Theoriticaly the facing sheet tensile strength at points A and B was the same but manufacturing process variables randomly dictated the final weak point of failure. Figure 4.35 and Figure 4.35 show locations of macro element lists for graphs of the tow dominant facing sheet stress components. The images only show the repaired facing sheet within the Nomex core section. At critical point A the peak stress is in the first repair ply and at critical point B the peak stress is in the top most original ply. The graphs of the X and Y stress components show sharp increases in the X stress component at these critical points in Figure 4.37 and Figure 4.38. Although critical point A shows higher stresses manufacturing variables dictate the final failure location as witnessed from experimental results.

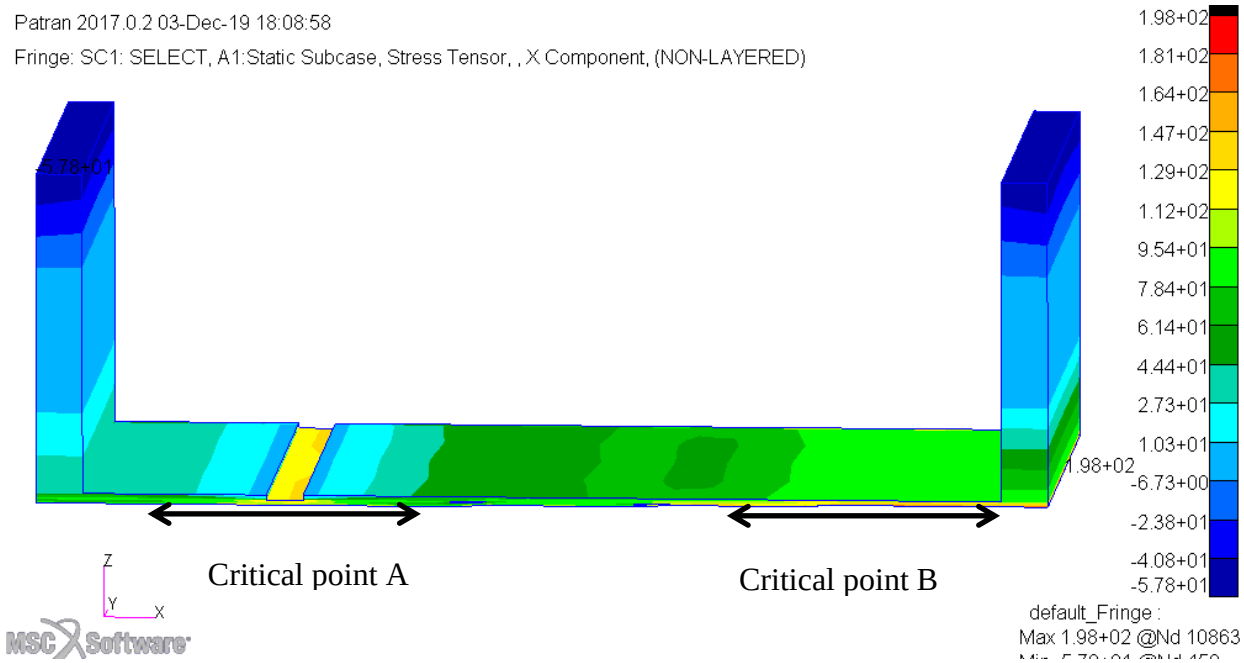


Figure 4.35 Element list positions on a stress plot of the macro model (critical point A)

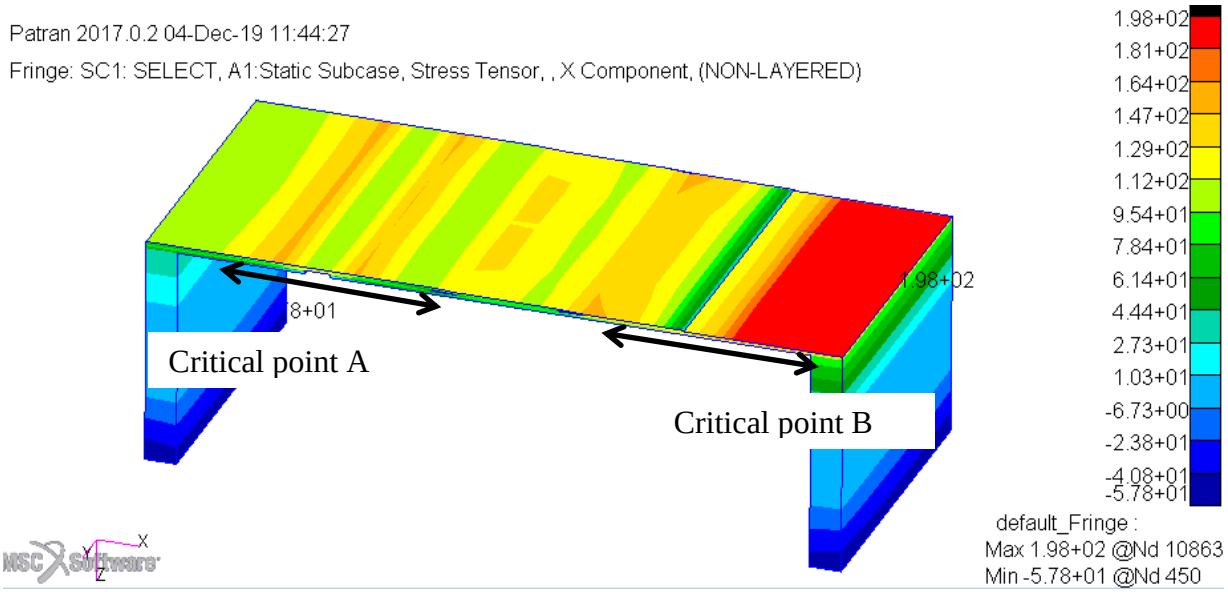


Figure 4.36 Element list positions on a stress plot of the macro model (critical point B)

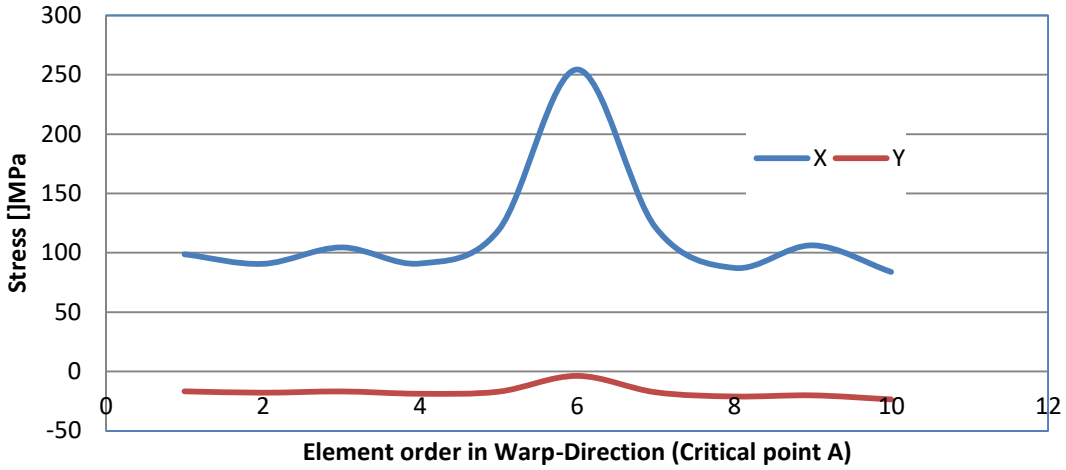


Figure 4.37 Element centroid stress components for critical point A

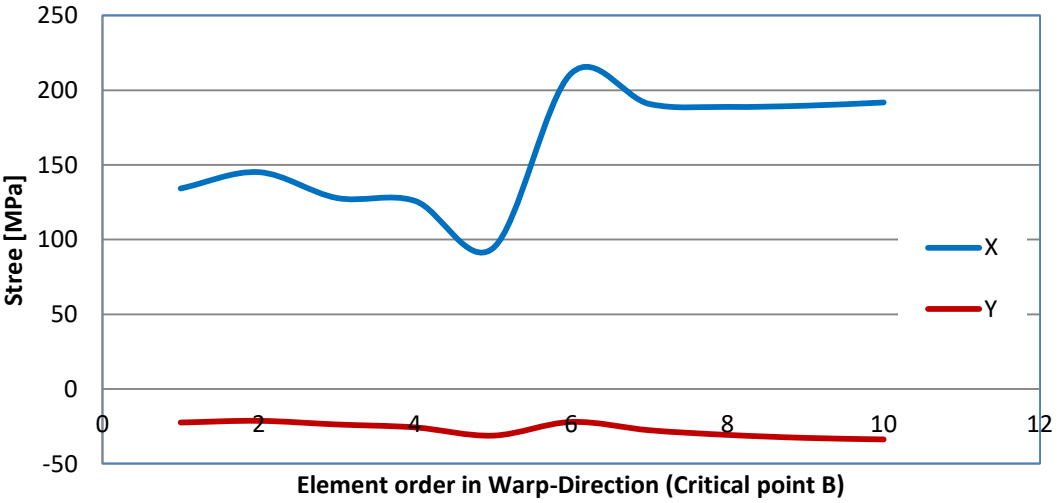


Figure 4.38 Element centroid stress components for critical point B

4.4 Fatigue Experimental Results

When fatigue loading cycles were plotted against tool facing sheet stresses the data points made a gentle downwards slope with increasing cycles. The points of interest, in this case, were the blue diamonds in Figure 4.39. These points were below most of the static failure points. When the tool facing sheet stress was reduced to about 75% of the final tension failure stress of the tool facing sheet other modes of failure started to dominate. These modes of failure were compression failure of the thicker facing sheet of the sandwich and facing sheet delaminations from the aluminum core. The last point at about 453 MPa facing sheet stress is from a specimen that did not fail to indicate that at this load the specimens are durable. An endurance limit line is set at 60% of the tensile failure stress 417 MPa. The information presented here is for the short cycle fatigue behaviour and the failure points are well above the endurance limit. Table 4.4 presents the fatigue results for the fatigue specimens.

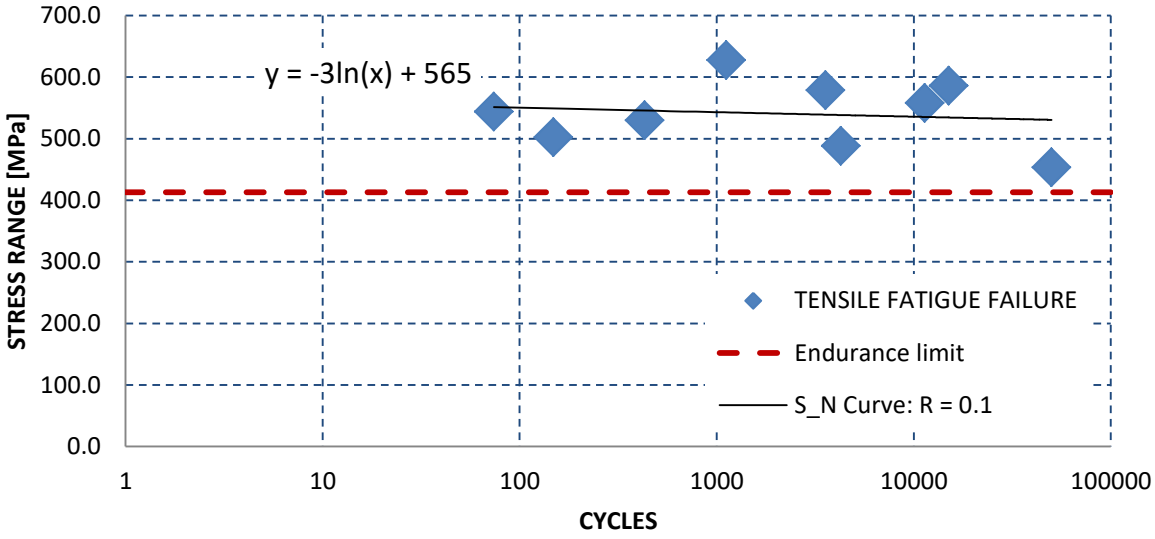


Figure 4.39 Experimental fatigue loading cycles against tool facing sheet stresses: S-N curve

Table 4.4 Applied fatigue loads for specimens

SPECIMEN ID	Load Amplitude [KN]	Mean Load [KN]	Number of Cycles	Failure Point
W93_5	2.92	3.57	4263	A
W93_7	3.78	4.61	148	A
W93_2	3.23	3.95	431	A
W93_3	3.14	3.84	74	A
W94_15	3.56	4.35	11341	B
W94_16	3.33	4.06	15022	B
W94_8	4.05	4.94	1116	B
W94_7	3.74	4.57	3560	B
W93_13	3.60	4.39	>50 000	

4.5 Validation of Fatigue Experimental Results Using MsM Analysis

The results from the MsM model are presented in Figure 4.40. The also exhibit a gentle downward slope as load cycles increase but at a much high rate than the experimental results. The vertical axis intercept from the fitted log regression curve equation however is close to the experimental result. The MsM regression results have a high slope because they are derived from a perfect model but the experimental regression results have the impact ot manufacturing process variables inherently included in them.

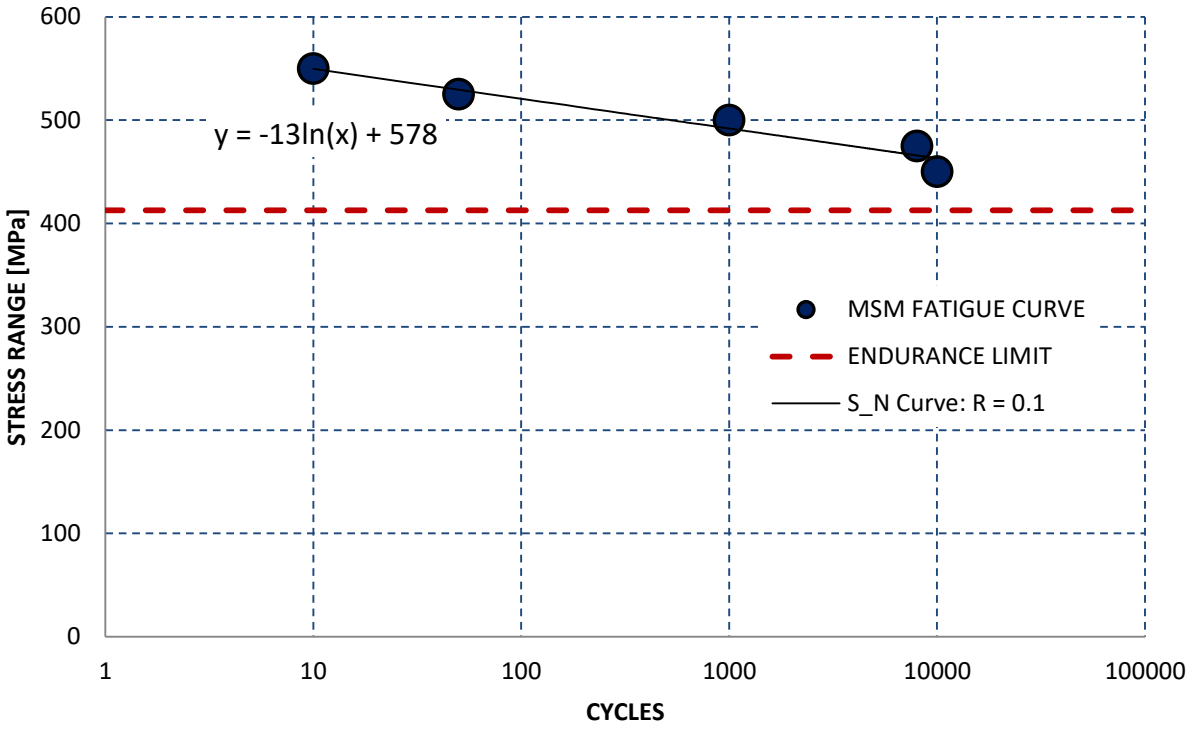


Figure 4.40 MsM fatigue loading cycles against tool facing sheet stresses: S-N curve

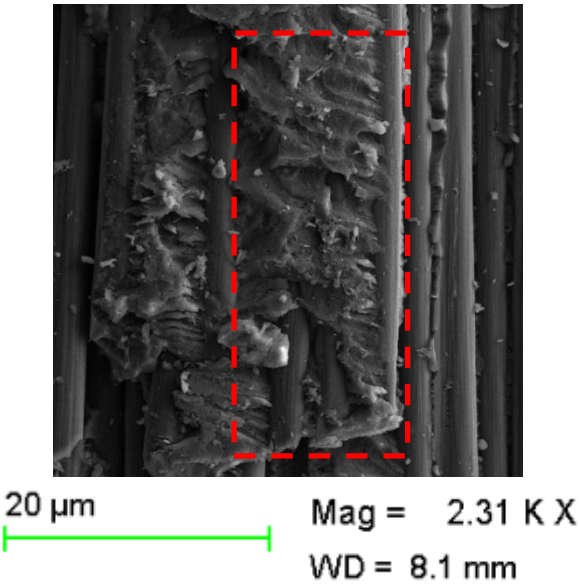
4.6 Fatigue Failure

In this section the fatigue failure results will be presented and discussed for the micro, meso and macro length scales following the same logical structure as in static failure section. Failure morphologies of the tool facing sheet, warp tows, weft tows, matrix and carbon fibres will be presented and discussed.

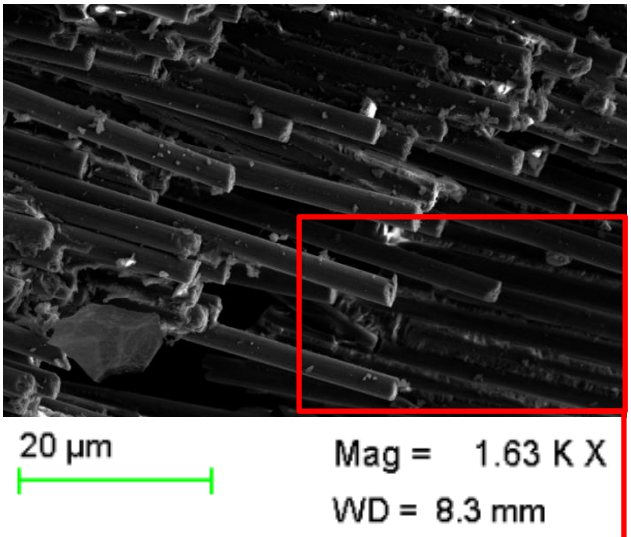
4.6.1 Experimental Micro Scale Failure

SEM pictures below show matrix and fibre failure at the micro scale due to fatigue loads. The repeating loads under fatigue loading create distinct failure features on the matrix that are different from the smoother matrix failure surfaces of static loads in Figure 4.11. As shown in Figure 4.41 (a) and (c) the matrix failure surface is rough and has cusp like features. The exposed failure lengths of the carbon fibres are also longer compared to static loading fibre failure ends. The surfaces of the carbon fibres are smoother than those of the statically loaded fibres, in Figure 4.11,

as can be seen in Figure 4.41 (b). These observations indicate that there was a lot of interfacial and matrix failure before final rupturing of the tool facing sheet.



(a) Matrix cusps formed by repeated loading



(b) Long protruding carbon fibres showing signs of pull-out



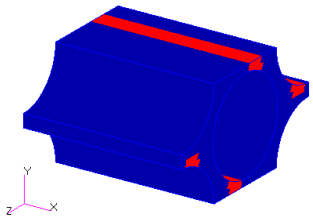
(c) Matrix flaking failure also caused by repeated loading

Figure 4.41 Fatigue microstructural failure at CP B.

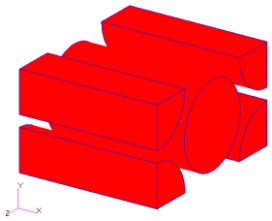
4.6.2 MsM Micro Scale Failure

The micro scale failed elements results were prepared with strain values from the macro element 6630 and its position in the meso RVE model is marked by the yellow circle in Figure 4.28. Figure 4.42 below shows the degree of failure of the matrix and fibre as the applied load cycles increased. Initially, the micro matrix elements failed in large numbers at 10 cycles 7.0 kN but the microfibre elements remained intact. This is because there are a lot of highly loaded matrix elements as shown in the static MsM results in Figure 4.13 [a]. Since the loading is cyclic the remaining matrix elements need to carry the load and eventually fail as well. This observation also agrees with the results of Figure 4.41 [b].After 100 cycles the micro matrix elements all failed while the

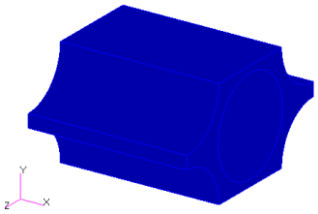
micro fibre elements remained intact. At the next load cycle increment of 5000 cycles, a large number of the micro fibre elements failed. After 5000 cycles the micro RVE model softened and stopped carrying additional load since its failure profile did not change at 9000 cycles.



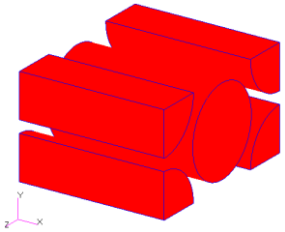
Matrix elements at 10 cycles



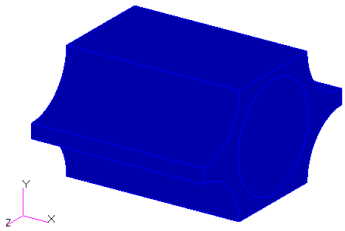
Fibre elements at 10 cycles



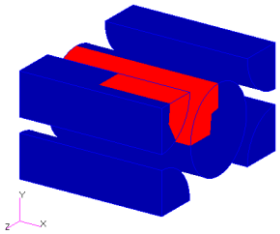
Matrix elements at 100 cycles



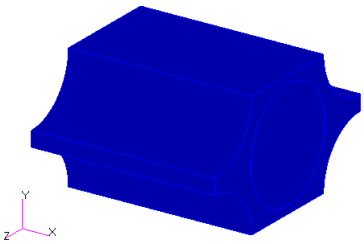
Fibre elements 100 cycles



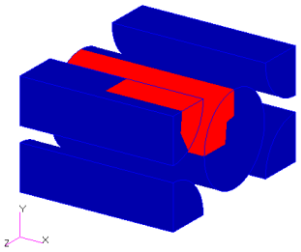
Matrix elements at 5000 cycles



Fibre elements at 5000 cycles



Matrix elements at 9000 cycles



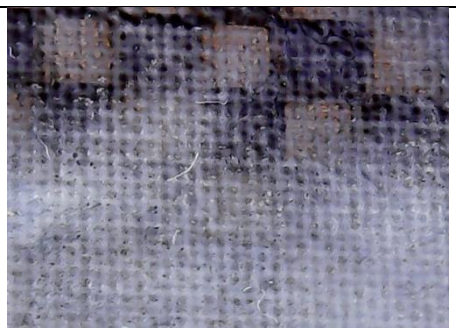
Fibre elements at 9000 cycles

Figure 4.42 Micro RVE Fatigue at 7.0 kN with increase cycles (meso element 6666)

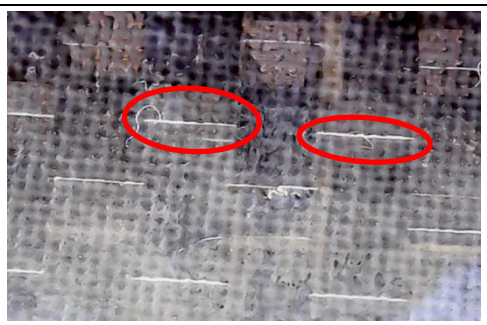
4.6.3 Experimental Meso Scale Failure

Mesoscale failure started in the matrix and this was observed as white streaks on the surface of the facing sheets. These streaks became more visible with increasing load cycles. They were also only observed over the weft tows as shown in Figure 4.43 (b), (c) and (d) below. Figure 4.43 (a) shows a damage free surface of the facing sheet. This interesting failure observation was not easy to see on a pure epoxy matrix covered areas of the tool facing sheet. It was mostly easy to see on surfaces covered by the thin film adhesive joining the repair plies to the parent laminates. The streaks had a distinct staggered pattern and the lines stopped abruptly at the edges of the warp

tows. In Figure 4.43 (e) and (f) mesoscale failure was also visible as tow rupture distributed randomly at the tow ramps. The weft and warp tows seemed to fail by extensive inter-tow failure over large surface areas compared to static inter-tow failure surface areas. Intra-tow failure was limited to the weft tow at the edge of the tool facing sheet fracture.



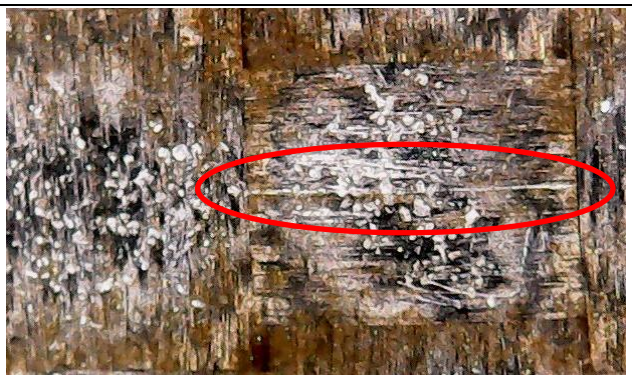
(a) Zero loads facing sheet surface



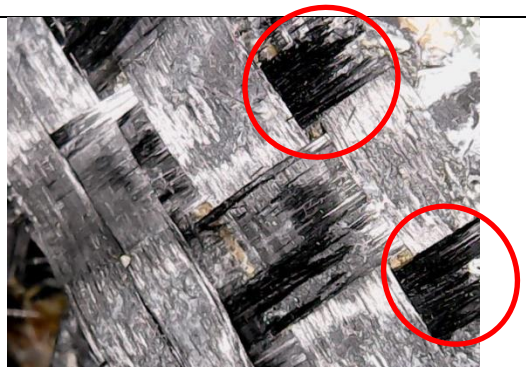
(b) Streaks of matrix cracking through facing sheet



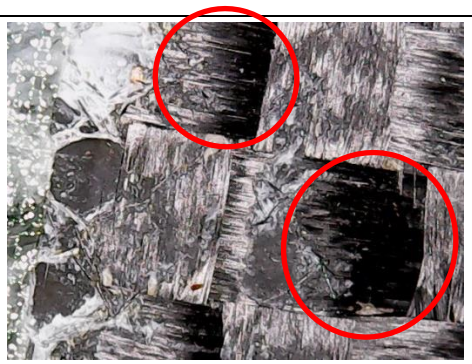
(c) Streaks of severe matrix cracking through facing sheet



(d) Close-up picture of matrix and underlying weft tow geometry



(e) Ruptured warp tows



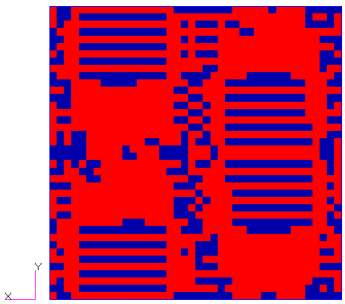
(f) More ruptured warp tows

Figure 4.43 Meso scale failure observations at CP B

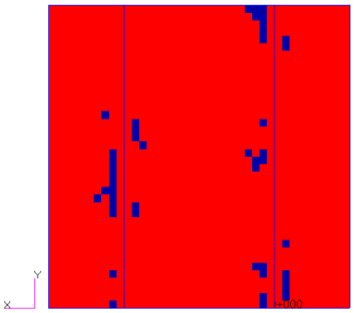
4.6.4 MsM Meso Scale Failure

The meso scale failed elements results in Figure 4.44 were prepared with strain values from the macro element 7677 in critical point A. The warp tow elements eventually fail although the applied load of 7.0 kN was lower than the failure load of 9.97 kN in the static experiment. This warp tow failure profile resulted in the same serrated failure edge, in Figure 4.29, obtained in the static MsM results. Weft tow failed elements had a similar failure pattern to the static meso weft

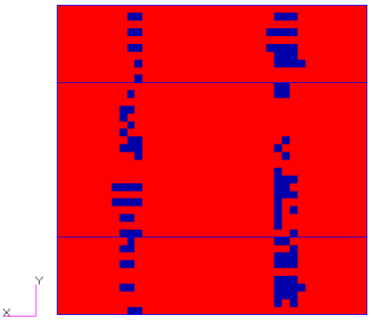
tow failure pattern. The observed MsM meso failure profile showed two similarities to the experimental observations. First, the meso matrix failed at lower applied cycles and failure in the warp tows was initiated at the weft tow edge and warp tow intersection. The meso RVE matrix failure occurred at loads that were lower than static failure loads. This type of failure was only visible in the fatigue experimental results as excessive weft and warp tow separation after complete tool facing sheet failure. On the surface of the tool facing sheet, matrix failure was observed as several white streaks in a staggered pattern. There is also a staggered failure pattern of the MsM meso matrix elements matching the white streaks. The white streak also confirm the claim that was made in the static MsM and experimental results that says matrix meso elements fail before the warp and weft tow elements.



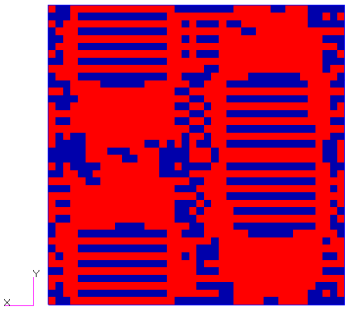
Matrix elements at 10 cycles



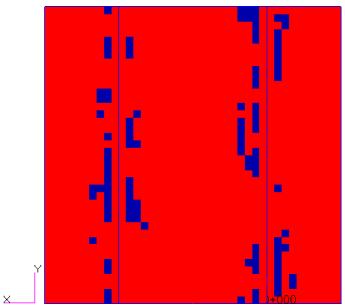
Weft elements at 10 cycles



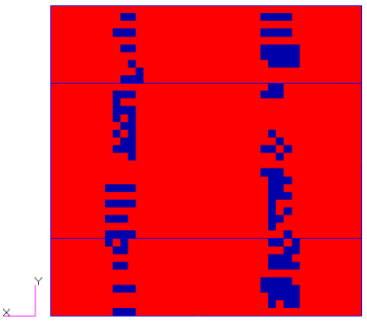
Warp elements at 10 cycles



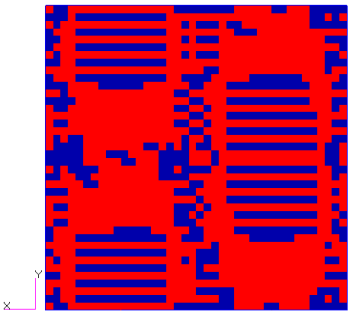
Matrix elements at 100 cycles



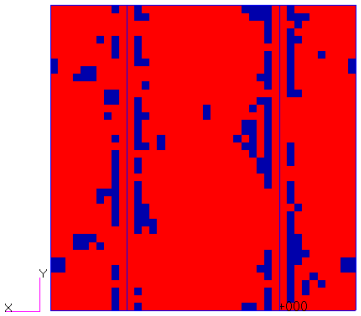
Weft elements at 100 cycles



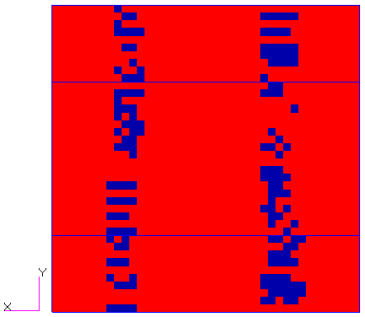
Warp elements at 100 cycles



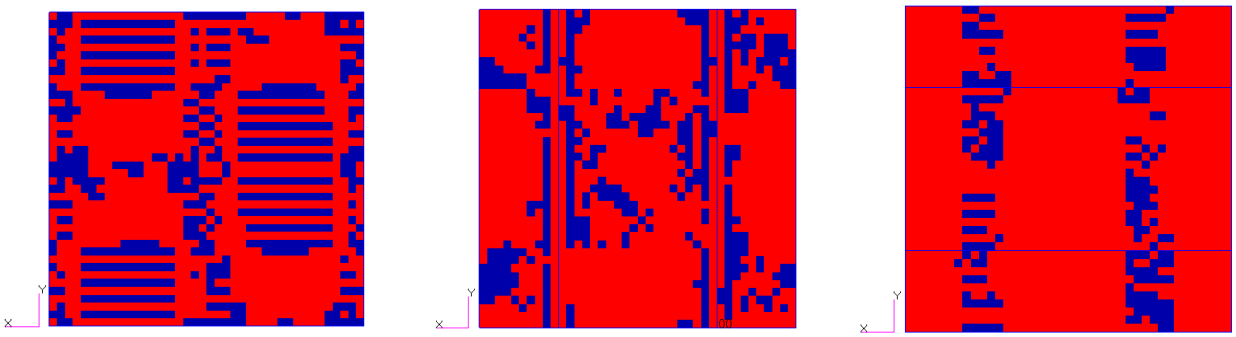
Matrix elements at 5000 cycles



Weft elements at 5000 cycles



Warp elements at 5000 cycles

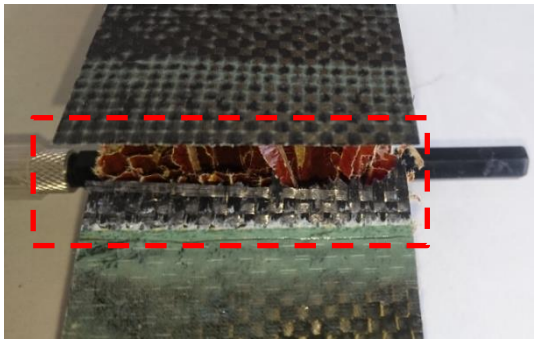


Matrix elements at 9000 cycles Weft elements at 9000 cycles Warp elements at 9000 cycles

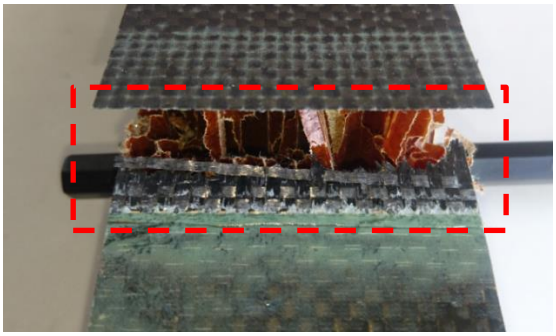
Figure 4.44 Fatigue damage to meso RVE as a result of 7.0 kN at various cycles (macro element 7677)

4.6.5 Experimental Macro Scale Failure

Unlike the narrow and vertical surface failures of the static test specimens, the failure surface of the fatigue specimens was wide and sloping across the tool facing sheet. Evidence of this failure surfaces can be seen in Figure 4.45 (a) and (b) at critical point B. This behaviour is a result of gradual matrix cracking over a long period of fatigue load application. Inspection of the second critical area at the filler plies bond line shows extensive cracking of the matrix at critical point A as shown in pictures of Figure 4.45 (c) and (d). These cracks were much deeper than the cracks noted in the static failure specimens. The greater depth was a result of progressive matrix cracking as more loading cycles were applied to the specimens. The experimental fatigue results of the various loading conditions are presented in Table 3.3.



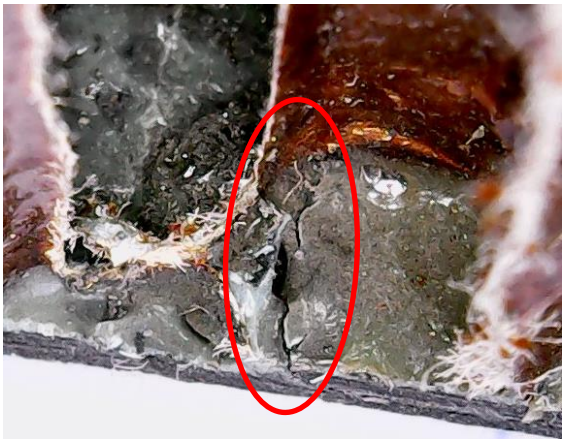
(a) Wide failure strip across the original tool facing sheet at CP B



(b) Wide failure strip across the original tool facing sheet at CP B



(c) Bond line matrix cracking at CP-A



(d) Bond line matrix cracking at CP-A

Figure 4.45 Macro scale failure cases of the repaired facing sheet

4.6.6 MsM Macro Scale Failure

The MsM macro scale results were presented in section 4.5 for the validation of the fatigue results. It should be noted as well that the S-N curves in Figure 4.39 and Figure 4.40 are for a fluctuating load with a ration R=0.1. The limitations of claims based on the results are not only limited to loading but extend to assuming a perfect manufacturing process in the case of the MsM approach. The effect of stress concentration effects on the results was however captured as shown in the static MsM results where two critical points A and B were highlighted at the macro scale. The model did not include various factors that also influence fatigues results therefore its results should be adopted with caution.

CHAPTER 5 - **CONCLUSION**

In the static and fatigue experiment carbon fibre sandwich panels with plain weave facing sheets were manufactured. A damage slot was introduced and thereafter a repaired with filler plies and overlapping reinforcing plies implemented on the tool facing sheet of the sandwich panels. Two sets of specimens were cut from the sandwich panels using a diamond coated cutting disc for static and fatigue experiments. The static experiment was conducted on a 5500R Instron testing machine and a four point bending testing setup and the fatigue experiment was conducted on an Isntron servo-hydraulic testing machine on a four point bending setup as well. A statistical analysis of the static experimental results validated the reliability of the static results before a fractographic analysis of the fracture areas was done at three length scales. At the macro scale two critical failure areas were observed at the thinnest points of the tool facing sheet. The thinnest points experienced the highest tensile stress from the applied four-point bending moment. Point A was located at the edge of the damage slot and point B was located at the edge of the top most repair ply. Matrix cracking occurred at point A before final facing sheet failure at either point A or B. Experimental results showed about a 50 % probability of facing sheet failure at either point A or B. At the meso scale various modes of failure were observed namely intra-tow cracking, inter-tow cracking, matrix cracking and tow rupturing. Tow rupturing only occurred at the facing sheet final failure stress due to the facing sheet tensile stress. At the micro scale matrix cracking occurred before final failure of the facing sheet due to tensile stresses. Fibre failure only occurred at facing sheet failure locations due to tensile stresses from the bending moment.

From the fatigue experiment a short cycle S-N curve was produced for a tension-tension loading ratio of R = 0.1. A qualitative evaluation of the fatigue results exhibited a downward trend of applicable facing sheet tension loading with increasing cycles as was expected. Fatigue specimens used for the S-N curve were thereafter fractographically analysed at the fracture areas. At the macro scale excessive matrix cracking at critical points A and B was observed to occur way before the facing sheet final tensile stress was reached. At the meso scale matrix cracking and intra-tow splitting was observed before the facing sheet tensile failure at critical point B. After facing sheet failure excessive inter-tow, delamination and intra-tow failure were observed. Compared to static experimental failure areas, fatigue failure areas were about twice as larger.

In the MsM three finite element models were created to analyse the experimental specimens at three length scales. At the macro scale the finite element model was created to capture the geometric features of the plies and cores. The material properties of the plies were determined at the meso scale. The macro model only used solid elements for plies and cores. For the meso scale the emphasis was to accurately model the plain weave and matrix geometric features. The meso model also used solid elements since they are able to model complex real life solids. Meso material properties were determined at the micro model. At the micro scale only one complete and four segments of fibres within a matrix were modelled to investigate stress profile at this scale. Micro material properties were obtained directly from the materials suppliers' specifications. To determine the facing sheet static failure load the MsM approach used an element property degradation scheme based on the finite elements stress values. For the fatigue failure loads the MsM approach used the finite elements stress values and load cycles for element properties degradation scheme. Miner's rule was used to incorporate the impact of the load cycles in the

element properties degradation scheme. The static MsM failure load and fatigue curve were lower than the experimental values.

At the macro scale MsM predicted the failure locations of critical point A and B exactly at the same positions as the experimental specimens' failure locations. At the meso scale failure of the matrix was accurately predicted to occur first for the static and fatigue models. Weft tow splitting in the meso scale model was exhibited like it happened in the static and fatigue experimental specimens. The fracture line location for the warp tows was the same for both the experimental specimens and the meso scale model. Very little progressive failure characterisation was exhibited by the experimental specimens and MsM model. The main conclusion about the micro scale was that both the matrix and carbon fibres failed in a brittle manner. The MsM approach performed well for static loads but did not do as well for fatigue load mainly due to the ommittion of manufacturing factors in the simulation process.

In practical application structural FRP material failure result from fatigue loads rather than static loads. Determining fatigue failure loads for composites with unique ply sequence and ply orientations is expensive hence the development of the MsM failure characterisation approach which could be extended to any FRP material. Since MsM failure characterisation works at different length scales, it highlights meso scale and micro scale failures that need to be monitored during structural inspections because these failures preceed failures at the macro scale. For most RFP applications macro scale failure determines structural failure but it behaves in a brittle manner. This means structural failure can occur suddenly and without warning, as in the case of the repaired facing sheets if monitored only at the macro scale. On the other hand MsM can highlight failure locations before the final structural failure at the macro scale. MsM failure characterisation is still in its infancy but has the potential to reduce the high cost of determining the suitability of FRP applications for highly demanding applications like aircraft primary structures. Rather than testing all possible material design configurations MsM could be used to explore the possible material designs and reserve testing for the final material configurations. In MRO operations it can also be used to support structural health monitoring activities.

CHAPTER 6 - **REFERENCES**

[1] E. Ghazali, M.-L. Dano, A. Gakwaya, and C.-O. Amyot, "Experimental and numerical studies of stepped-scarf circular repairs in composite sandwich panels," *Int. J. Adhes. Adhes.*, vol. 82, pp. 41–49, Apr. 2018.

[2] Roland Thevenin, "Advanced Repair Technologies to Meet Future MRO Demand." Aircraft Composite Repair Management Forum, Aviation Week, 26-Sep-2011.

[3] *AMC-20 General Acceptable Means of Compliance for Airworthiness of Products, Parts and Appliances*. Germany: EASA, 2010.

[4] A. Baker, A. J. Gunnion, and J. Wang, "On the Certification of Bonded Repairs to Primary Composite Aircraft Components," *J. Adhes.*, vol. 91, no. 1–2, pp. 4–38, Jan. 2015.

[5] U. D. O. Defense, *Composite Materials Handbook-MIL 17, Volume 2: Polymer Matrix Composites: Materials Properties*. CRC Press, 2000.

[6] A. M. G. Pinto, R. D. S. G. Campilho, M. F. S. F. de Moura, and I. R. Mendes, "Numerical evaluation of three-dimensional scarf repairs in carbon-epoxy structures," *Int. J. Adhes. Adhes.*, vol. 30, no. 5, pp. 329–337, Jul. 2010.

[7] F. A. A. S. & A. (ASA), *Aviation Maintenance Technician Handbook—Airframe: FAA-H-8083-31 Volume 1*, 2012 Edition edition. Newcastle, Wash.: Aviation Supplies and Academics, Inc., 2012.

[8] R. Talreja, "Multiscale Modeling of Failure in Composite Materials," *Proc. Indian Natl. Sci. Acad.*, vol. 0, no. 0, Jun. 2016.

[9] D. Groen, B. Bosak, V. Krzhizhanovskaya, A. Hoekstra, and P. Koumoutsakos, "Multiscale Modelling and Simulation, 14th International Workshop," *Procedia Comput. Sci.*, vol. 108, pp. 1811–1812, 2017.

[10] K. B. Katnam and L. F. M. da Silva, "Special Issue on Primary Bonded Repairs in Composite Structures," *J. Adhes.*, vol. 91, no. 1/2, pp. 1–3, Jan. 2015.

[11] K. B. Katnam, L. F. M. Da Silva, and T. M. Young, "Bonded repair of composite aircraft structures: A review of scientific challenges and opportunities," *Prog. Aerosp. Sci.*, vol. 61, pp. 26–42, Aug. 2013.

[12] K. B. Katnam, A. J. Comer, D. Roy, S. Da, and T. M. Young, "Composite repair in wind turbine blades: An overview," *J. Adhes.*, vol. 91, no. 1–2, pp. 113–139, 2015.

[13] G. Gardiner, "Primary structure repair: The quest for quality," *High Perform. Compos.*, vol. 19, no. 6, pp. 48–54, Nov. 2011.

[14] E. J. Barbero, *Finite Element Analysis of Composite Materials using Abaqus(TM)*, 1 edition. Boca Raton, Fla.: CRC Press, 2013.

[15] M. Lidgett *et al.*, "Multi-scale modelling of polymer composite materials under blast and ballistic loading," presented at the ICCM International Conferences on Composite Materials, 2011.

[16] S. Das, J Warren, D West, "Global Carbon Fiber Composites Supply Chain Competitiveness Analysis.pdf." Clean Energy Manufacturing Analysis Center (CEMAC), May-2016.

[17] Dr. E Witten, T. Kraus, M. Kuhnel, "The Composites Market 2014-2024 - MAT0009." AVK, Oct-2014.

[18] J. Z. Zhang, "Study on Carbon Fiber Composite Materials in Sports Equipment," *Appl. Mech. Mater.*, vol. 329, pp. 105–108, Jun. 2013.

[19] L. Zhang, "The Application of Composite Fiber Materials in Sports Equipment," 2015.

[20] M. Pervaiz, S. Panthapulakkal, B. Kc, M. Sain, and J. Tjong, "Emerging Trends in Automotive Lightweighting through Novel Composite Materials," *Mater. Sci. Appl.*, vol. 07, no. 01, pp. 26–38, 2016.

[21] The Faculty of Operation and Economics of Transport The Air Transport Department Univerzitna 1, 010 26 Zilina, Slovak Republic, mraz.marie@gmail.com and M. Maria,

"Advanced composite materials of the future in aerospace industry," *INCAS Bull.*, vol. 5, no. 3, pp. 139–150, Sep. 2013.
[22] Nikhil V Nayak, "Composite Materials in Aerospace Applications." International Journal of Scientific and Research Publications (IJSRP), Sep-2014.
[23] Industry Expert Panel, "Composite Joining and Repair (CJAR)." Georgia Institute of Technology, Mar-2015.
[24] G. Marsh, "The challenge of composite fuselage repair," *Reinf. Plast.*, vol. 56, no. 3, pp. 30–35, May 2012.
[25] H. M. Chong *et al.*, "Out-of-autoclave scarf repair of interlayer toughened carbon fibre composites using double vacuum debulking of patch," *Compos. Part Appl. Sci. Manuf.*, vol. 107, pp. 224–234, Apr. 2018.
[26] E. Kelly, "Composite challenge," *Asian Aviat. Mag.*, vol. 13, no. 8, pp. 40–41, Oct. 2015.
[27] F. Z. Hu and C. Soutis, "Strength prediction of patch-repaired CFRP laminates loaded in compression," *Compos. Sci. Technol.*, vol. 60, no. 7, pp. 1103–1114, May 2000.
[28] B. G. Falzon, "AG-28-TP-155 - Impact damage and repair of composite structures," London: Imperial College, 2007.
[29] M. D. Banea and L. F. M. da Silva, "Adhesively bonded joints in composite materials: An overview," *Proc. Inst. Mech. Eng. Part J. Mater. Des. Appl.*, vol. 223, no. 1, pp. 1–18, Jan. 2009.
[30] "DOT FAA AR-03 74, Bonded Repair of Aircraft Composite Sandwich Structures," USA: Federal Aviation Administration (FAA), 2004.
[31] R. H. Stone, "Repair techniques for graphite/epoxy structures for commercial transport applications," Jan. 1983.
[32] J. S. Jones and S. R. Graves, "Repair techniques for celion/LARC-160 graphite/polyimide composite structures," Jun. 1984.
[33] S. Liu, Z. Guan, X. Guo, K. Sun, and J. Kong, "Edgewise compressive performance of repaired composite sandwich panels – Experiment and finite element analysis," *J. Reinf. Plast. Compos.*, vol. 32, no. 18, pp. 1331–1347, Sep. 2013.
[34] S. Liu, Z. Guan, X. Guo, K. Sun, J. Kong, and D. Yan, "Process Factors and Edgewise Compressive Properties of Scarf-repaired Honeycomb Sandwich Structures," *Appl. Compos. Mater.*, vol. 21, no. 5, pp. 689–705, Nov. 2013.
[35] S. Mahdi, A. J. Kinloch, F. L. Matthews, and M. A. Crisfield, "The Static Mechanical Performance of Repaired Composite Sandwich Beams: Part I - Experimental Characterisation," *J. Sandw. Struct. Mater.*, vol. 5, no. 2, pp. 179–202, Apr. 2003.
[36] D. Zenkert and M. Burman, "Failure mode shifts during constant amplitude fatigue loading of GFRP/foam core sandwich beams," *Int. J. Fatigue*, vol. 33, no. 2, pp. 217–222, Feb. 2011.
[37] I. M. Daniel and J. L. Abot, "Fabrication, testing and analysis of composite sandwich beams," *Compos. Sci. Technol.*, vol. 60, Sep. 2000.
[38] D. A. Ramantani, R. D. S. G. Campilho, M. F. S. F. De Moura, and A. T. Marques, "Stress and Failure Analysis of Repaired Sandwich Composite Beams using a Cohesive Damage Model," *J. Sandw. Struct. Mater.*, vol. 12, no. 3, pp. 369–390, May 2010.
[39] M. Kaminski, F. Laurin, J. F. Maire, C. Rakotoarisoa, and E. Hémon, "Fatigue damage modeling of composite structures: the onera viewpoint," *AerospaceLab*, no. 9, pp. 1–12, Aug. 2015.
[40] M. S. Vinod, S. B. J, V. Nayaka, R. Shenoy, M. S. Murali, and A. Nafidi, "Fractography of compression failed carbon fiber reinforced plastic composite laminates," *J. Mech. Eng. Res.*, vol. 2, no. 1, pp. 001–009, Feb. 2010.
[41] J. Bieniaś, M. Ostapiuk, and B. Surowska, "Fractography and Damage Analysis of Carbon/Epoxy Composites Under Static and Dynamic Loading at Elevated Temperatures," *Acta Mech. Autom.*, vol. Vol. 6, no. 1, pp. 17–20, 2012.

[42] D. A. Ramantani, R. D. S. G. Campilho, M. F. S. F. D. Moura, and A. T. Marques, “Stress and Failure Analysis of Repaired Sandwich Composite Beams using a Cohesive Damage Model,” *J. Sandw. Struct. Mater.*, vol. 12, no. 3, pp. 369–390, May 2010.
[43] S. Mahdi, A. J. Kinloch, F. L. Matthews, and M. A. Crisfield, “The Static Mechanical Performance of Repaired Composite Sandwich Beams: Part II - Finite Element Modelling,” *J. Sandw. Struct. Mater.*, vol. 5, no. 3, pp. 267–303, Jul. 2003.
[44] Broughton, W R, Crocker, L E, Gower, M R L, “Design requirements for bonded and bolted composite structures,” NPL Report MATC(A)65, 2002.
[45] X.-J. G. Pengcheng Cheng, “Tensile behaviour of patch-repaired CFRP laminates,” *Compos. Struct.*, vol. 93, no. 2, pp. 582–589, 2011.
[46] X. Liu and G. Wang, “Progressive failure analysis of bonded composite repairs,” *Compos. Struct.*, vol. 81, no. 3, pp. 331–340, Dec. 2007.
[47] A. Sayyidmousavi, H. Bougherara, and Z. Fawaz, “A multiscale approach for fatigue life prediction of polymer matrix composite laminates,” *J. Reinf. Plast. Compos.*, p. 0731684415588936, May 2015.
[48] Z.-M. Huang, “Micromechanical modeling of fatigue strength of unidirectional fibrous composites,” *Int. J. Fatigue*, vol. 24, no. 6, pp. 659–670, Jun. 2002.
[49] X. He, “A review of finite element analysis of adhesively bonded joints,” *Int. J. Adhes. Adhes.*, vol. 31, no. 4, pp. 248–264, Jun. 2011.
[50] M. M. Abdel Wahab, “Fatigue in Adhesively Bonded Joints: A Review,” *ISRN Mater. Sci.*, vol. 2012, pp. 1–25, 2012.
[51] E. J. Barbero, *Finite Element Analysis of Composite Materials Using ANSYS®, Second Edition*. CRC Press, 2013.
[52] X. Yue and Weinan E, “The local microscale problem in the multiscale modeling of strongly heterogeneous media: Effects of boundary conditions and cell size,” *J. Comput. Phys.*, vol. 222, no. 2, pp. 556–572, 2007.
[53] S. Tsai, *Strength and life of composites*, 1st ed. jec composites, 2008.
[54] A. R. de O. S. S. Melro, “Analytical and numerical modelling of damage and fracture of advanced composites,” Mar. 2013.
[55] J. Fish, *Practical Multiscaling*, 1 edition. Chichester: Wiley, 2013.
[56] X. S. S. J. Z. Mao, “A Modeling Approach Across Length Scales for Progressive Failure Analysis of Woven Composites,” *Appl. Compos. Mater.*, vol. 20, no. 3, pp. 213–231, 2013.
[57] G. Couégnat, E. Martin, and J. Lamon, “3D Multiscale Modeling of the Mechanical Behavior of Woven Composite Materials,” in *Mechanical Properties and Performance of Engineering Ceramics and Composites V: Ceramic Engineering and Science Proceedings, Volume 31*, John Wiley & Sons, Inc., 2010, pp. 185–194.
[58] Y. Cao, Y. Feng, W. Wang, D. Wu, and Z. Zhu, “Estimation of Lamina Stiffness and Strength of Quadriaxial Non-Crimp Fabric Composites Based on Semi-Laminar Considerations,” *Appl. Sci.*, vol. 6, no. 9, p. 267, Sep. 2016.
[59] W. R. McLendon and J. D. Whitcomb, “Characteristic progressive damage modes in a plain weave textile composite under multiaxial loads,” *J. Compos. Mater.*, vol. 51, no. 11, pp. 1539–1556, May 2017.
[60] G. Nilakantan, M. Keefe, T. A. Bogetti, R. Adkinson, and J. W. Gillespie, “On the finite element analysis of woven fabric impact using multiscale modeling techniques,” *Int. J. Solids Struct.*, vol. 47, no. 17, pp. 2300–2315, Aug. 2010.
[61] G. Nilakantan, M. Keefe, T. A. Bogetti, and J. W. Gillespie, “Multiscale modeling of the impact of textile fabrics based on hybrid element analysis,” *Int. J. Impact Eng.*, vol. 37, no. 10, pp. 1056–1071, Oct. 2010.
[62] F. Otero, X. Martinez, S. Oller, and O. Salomón, “An efficient multi-scale method for non-linear analysis of composite structures,” *Compos. Struct.*, vol. 131, pp. 707–719, Nov. 2015.
[63] G. Ernst, M. Vogler, C. Hühne, and R. Rolfes, “Multiscale progressive failure analysis of textile composites,” *Compos. Sci. Technol.*, vol. 70, no. 1, pp. 61–72, Jan. 2010.

[64] Y. Wan, B. Sun, and B. Gu, “Multi-scale structure finite element analyses of damage behaviors of multi-axial warp-knitted composite materials subjected to quasi-static and high strain rate compressions,” *J. Text. Inst.*, vol. 107, no. 7, pp. 879–904, Jul. 2016.
[65] B. Sun, R. Liu, and B. Gu, “Numerical simulation of three-point bending fatigue of four-step 3-D braided rectangular composite under different stress levels from unit-cell approach,” *Comput. Mater. Sci.*, vol. 65, pp. 239–246, Dec. 2012.
[66] L. Wu, B. Sun, and B. Gu, “Numerical analyses of bending fatigue of four-step three-dimensional rectangular-braided composite materials from unit cell approach,” *J. Text. Inst.*, vol. 106, no. 1, pp. 67–79, Jan. 2015.
[67] L. Wu, F. Zhang, B. Sun, and B. Gu, “Finite element analyses on three-point low-cyclic bending fatigue of 3-D braided composite materials at microstructure level,” *Int. J. Mech. Sci.*, vol. 84, pp. 41–53, Jul. 2014.
[68] L. Wu and B. Gu, “Fatigue behaviors of four-step three-dimensional braided composite material: a meso-scale approach computation,” *Text. Res. J.*, vol. 84, no. 18, pp. 1915–1930, Nov. 2014.
[69] G. Yang, B. Sun, and B. Gu, “Large-scale finite element analysis of a 3D angle-interlock woven composite undergoing low-cyclic three-point bending fatigue,” *J. Text. Inst.*, vol. 105, no. 3, pp. 275–293, Mar. 2014.
[70] L. Wang, B. Wang, S. Wei, Y. Hong, and C. Zheng, “Prediction of long-term fatigue life of CFRP composite hydrogen storage vessel based on micromechanics of failure,” *Compos. Part B Eng.*, vol. 97, pp. 274–281, Jul. 2016.
[71] J. Song, W. Wen, H. Cui, and S. Zhao, “Study on static and fatigue behaviors of carbon fiber bundle and the statistical distribution by experiments,” *J. Compos. Mater.*, vol. 49, no. 25, pp. 3157–3168, Oct. 2015.
[72] Z.-M. Huang, “Micromechanical life prediction for composite laminates,” *Mech. Mater.*, vol. 33, no. 4, pp. 185–199, Apr. 2001.

A. APPENDIX A

COMPLIANCE MATRIX RELATIONSHIP TO ENGINEERING MATERIAL PROPERTIES

Micro RVE element stress state results in a matrix form for all the micro RVE elements are shown below.

$$\sigma_{micro_elements_stress_array} = \begin{bmatrix} \sigma_{xx_1} & \sigma_{yy_1} & \sigma_{zz_1} & \sigma_{xy_1} & \sigma_{yz_1} & \sigma_{zx_1} \\ \sigma_{xx_2} & \sigma_{yy_2} & \sigma_{zz_2} & \sigma_{xy_2} & \sigma_{yz_2} & \sigma_{zx_2} \\ \sigma_{xx_3} & \sigma_{yy_3} & \sigma_{zz_3} & \sigma_{xy_3} & \sigma_{yz_3} & \sigma_{zx_3} \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \sigma_{xx_m} & \sigma_{yy_m} & \sigma_{zz_m} & \sigma_{xy_m} & \sigma_{yz_m} & \sigma_{zx_m} \end{bmatrix}$$

Micro RVE element volumes vector is shown below

$$V_{micro_elements_volumes} = [V_1 \quad V_2 \quad V_3 \quad \cdot \quad \cdot \quad \cdot \quad V_m]$$

The stress matrix and the element volumes vector were multiplied together to give the columns of the stiffness matrix below.

$$S_{colm_case_i} = Transpose\left[V_{micro_elements_volumes} \times \sigma_{micro_elements_stress_array}\right] \times \frac{1}{\sum_{i=1}^{m} V_i}$$

Stiffness matrix's columns are obtained from the results of the six load cases in Figure 3.21. The zeros are a result of the decoupled direct stresses and shear stresses in the micro RVE. This means that pure extension or contraction does not cause shear deformations and a pure shear does not induce tension or compression of the micro RVE. Although the constituents were isotropic material the average material properties of the micro RVE are orthotropic. Hence there are zeros in the stiffness matrix below.

$$S = \begin{bmatrix} S_{xx} & S_{yx} & S_{zx} & 0 & 0 & 0 \\ S_{xy} & S_{yy} & S_{zy} & 0 & 0 & 0 \\ S_{xz} & S_{yz} & S_{zz} & 0 & 0 & 0 \\ 0 & 0 & 0 & S_{44} & 0 & 0 \\ 0 & 0 & 0 & 0 & S_{55} & 0 \\ 0 & 0 & 0 & 0 & 0 & S_{66} \end{bmatrix}$$

To simplify the calculation of the calculation of the engineering material properties from the stiffness matrix it first inverted to a compliance matrix. The inverse of the stiffness matrix results in the compliance matrix below. There is a simple relationship between its components and the required engineering material properties. For example the Young's modulus in the xx-direction was calculated from the reciprocal of the first component of the first column in the compliance matrix. The poisson's ratio in the xy-plane was obtained from multiplying the second component of the first column by the Young's modulus in the xx-direction.

$$C = \begin{bmatrix} C_{xx} & C_{yx} & C_{zx} & 0 & 0 & 0 \\ C_{xy} & C_{yy} & C_{zy} & 0 & 0 & 0 \\ C_{xz} & C_{yz} & C_{zz} & 0 & 0 & 0 \\ 0 & 0 & 0 & C_{44} & 0 & 0 \\ 0 & 0 & 0 & 0 & C_{55} & 0 \\ 0 & 0 & 0 & 0 & 0 & C_{66} \end{bmatrix} = \begin{bmatrix} \frac{1}{E_{xx}} & \frac{v_{yx}}{E_{yy}} & \frac{v_{zx}}{E_{zz}} & 0 & 0 & 0 \\ \frac{v_{xy}}{E_{xx}} & \frac{1}{E_{yy}} & \frac{v_{zy}}{E_{xx}} & 0 & 0 & 0 \\ \frac{v_{xz}}{E_{xx}} & \frac{v_{yz}}{E_{yy}} & \frac{1}{E_{zz}} & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{2G_{xy}} & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{2G_{yz}} & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{1}{2G_{xz}} \end{bmatrix}$$

B. APPENDIX B

EXCEL MACROS: VBA CODE

1. **VBA Scripts pair_surf ().**

```
Option Explicit
Option Base 1
Dim datasht, matchsht, pairsht As Worksheet
Sub pair_surf()
Dim res_range, res_rangex, res_rangey, res_rangez, nodes_range, dist_range As Range
Dim front(), back(), right(), left(), top(), bottom(), dist_1(), dist_2(), dist_3() As Double
Dim cols, rows As Long
Dim minifz, maxifz, minify, maxify, minifx, maxifx, dist, dist_min As Double
Dim i, j, w, k, n, q, h, ss1, ss2, ss3, ss4, ss5, ss6, ss, sz1, sz2, sx1, sx2, sy2, sy1, r_min As Integer
Set datasht = Worksheets("RVE_NODES")
Set pairsht = Worksheets("PAIR")
   With datasht
   Set res_range = .Range(.Cells(3, 3), .Cells(11125, 6))
   End With
   cols = res_range.Columns.Count
   rows = res_range.rows.Count
   pairsht.Cells(2, 2) = cols
   pairsht.Cells(3, 2) = rows
   Set res_rangex = res_range.Columns(2)
   Set res_rangey = res_range.Columns(3)
   Set res_rangez = res_range.Columns(4)
```

```
    pairsht.Cells(2, 5) = res_rangez.Columns.Count
    pairsht.Cells(3, 5) = res_rangez.rows.Count
    Set nodes_range = res_range.Columns(1)
    minifz = Application.WorksheetFunction.Min(res_rangez)
    maxifz = Application.WorksheetFunction.Max(res_rangez)
    minify = Application.WorksheetFunction.Min(res_rangey)
    maxify = Application.WorksheetFunction.Max(res_rangey)
    minifx = Application.WorksheetFunction.Min(res_rangex)
    maxifx = Application.WorksheetFunction.Max(res_rangex)
'**************BEGINING of front back pairing**********************
    For i = 1 To rows
        'Front face nodes
        If res_rangez.Cells(i, 1) = maxifz And res_rangey.Cells(i, 1) <> maxify And
res_rangey.Cells(i, 1) <> minify _
                        And res_rangex.Cells(i, 1) <> maxifx And res_rangex.Cells(i, 1) <> minifx
Then
            sz1 = sz1 + 1
        End If
        'Back face nodes
        If res_rangez.Cells(i, 1) = minifz And res_rangey.Cells(i, 1) <> maxify And
res_rangey.Cells(i, 1) <> minify _
                        And res_rangex.Cells(i, 1) <> maxifx And res_rangex.Cells(i, 1) <> minifx
Then
            sz2 = sz2 + 1
        End If
    Next
    ReDim front(1 To sz1, 1 To 4)
    ReDim back(1 To sz1, 1 To 4)
    ReDim dist_1(1 To sz1)
    ss1 = 1
```

```
ss2 = 1
For i = 1 To rows
    'Front face nodes
    If res_rangez.Cells(i, 1) = maxifz And res_rangey.Cells(i, 1) <> maxify And res_rangey.Cells(i, 1) <> minify _
                And res_rangex.Cells(i, 1) <> maxifx And res_rangex.Cells(i, 1) <> minifx Then
        front(ss1, 2) = res_rangex.Cells(i, 1)
        front(ss1, 3) = res_rangey.Cells(i, 1)
        front(ss1, 4) = res_rangez.Cells(i, 1)
        front(ss1, 1) = nodes_range.Cells(i, 1)
        ss1 = ss1 + 1
    End If
    'Back face nodes
    If res_rangez.Cells(i, 1) = minifz And res_rangey.Cells(i, 1) <> maxify And res_rangey.Cells(i, 1) <> minify _
                And res_rangex.Cells(i, 1) <> maxifx And res_rangex.Cells(i, 1) <> minifx Then
        back(ss2, 2) = res_rangex.Cells(i, 1)
        back(ss2, 3) = res_rangey.Cells(i, 1)
        back(ss2, 4) = res_rangez.Cells(i, 1)
        back(ss2, 1) = nodes_range.Cells(i, 1)
        ss2 = ss2 + 1
    End If
Next
'calculate distances
For j = 1 To sz1
    For w = 1 To sz1

        dist_1(w) = ((front(j, 2) - back(w, 2)) ^ 2 + _
```

```
                (front(j, 3) - back(w, 3)) ^ 2 + (front(j, 4) - back(w, 4)) ^ 2) ^ 0.5
    Next
    'pair nodes
    dist_min = WorksheetFunction.Min(dist_1)
    r_min = WorksheetFunction.Match(dist_min, dist_1, 0)
    pairsht.Cells(4 + j, 3) = front(j, 1)
    pairsht.Cells(4 + j, 4) = back(r_min, 1)
    pairsht.Cells(4 + j, 5) = dist_min
  Next
'**************END of front back pairing*****************************


'************BEGINING of right left pairing************************
  For i = 1 To rows


    'Right face nodes
    If res_rangex.Cells(i, 1) = maxifx And res_rangey.Cells(i, 1) <> maxify And
res_rangey.Cells(i, 1) <> minify _
                And res_rangez.Cells(i, 1) <> maxifz And res_rangez.Cells(i, 1) <> minifz
Then
      sx1 = sx1 + 1
    End If
    'Left face nodes
    If res_rangex.Cells(i, 1) = minifx And res_rangey.Cells(i, 1) <> maxify And
res_rangey.Cells(i, 1) <> minify _
                And res_rangez.Cells(i, 1) <> maxifz And res_rangez.Cells(i, 1) <> minifz
Then
      sx2 = sx2 + 1
    End If
  Next
  ReDim right(1 To sx1, 1 To 4)
```

```
ReDim left(1 To sx1, 1 To 4)

ReDim dist_2(1 To sx1)

ss3 = 1

ss4 = 1

For i = 1 To rows

    'right face nodes

    If res_rangex.Cells(i, 1) = maxifx And res_rangey.Cells(i, 1) <> maxify And
res_rangey.Cells(i, 1) <> minify _

                And res_rangez.Cells(i, 1) <> maxifz And res_rangez.Cells(i, 1) <> minifz
Then

        right(ss3, 2) = res_rangex.Cells(i, 1)

        right(ss3, 3) = res_rangey.Cells(i, 1)

        right(ss3, 4) = res_rangez.Cells(i, 1)

        right(ss3, 1) = nodes_range.Cells(i, 1)

        ss3 = ss3 + 1

    End If

    'left face nodes

    If res_rangex.Cells(i, 1) = minifx And res_rangey.Cells(i, 1) <> maxify And
res_rangey.Cells(i, 1) <> minify _

                And res_rangez.Cells(i, 1) <> maxifz And res_rangez.Cells(i, 1) <> minifz
Then

        left(ss4, 2) = res_rangex.Cells(i, 1)

        left(ss4, 3) = res_rangey.Cells(i, 1)

        left(ss4, 4) = res_rangez.Cells(i, 1)

        left(ss4, 1) = nodes_range.Cells(i, 1)

        ss4 = ss4 + 1

    End If

Next

'calculate distances

For k = 1 To sx1
```

```
        For n = 1 To sx1


            dist_2(n) = ((right(k, 2) - left(n, 2)) ^ 2 + _
                    (right(k, 3) - left(n, 3)) ^ 2 + (right(k, 4) - left(n, 4)) ^ 2) ^ 0.5
        Next
        'pair nodes
        dist_min = WorksheetFunction.Min(dist_2)
        r_min = WorksheetFunction.Match(dist_min, dist_2, 0)
        pairsht.Cells(4 + k, 7) = right(k, 1)
        pairsht.Cells(4 + k, 8) = left(r_min, 1)
        pairsht.Cells(4 + k, 9) = dist_min
    Next
'************END of right left pairing************************************


'*****************BEGINING of top bottom pairing************************
    For i = 1 To rows
        'Top face nodes
        If res_rangey.Cells(i, 1) = maxify And res_rangez.Cells(i, 1) <> maxifz And res_rangez.Cells(i, 1) <> minifz _
                        And res_rangex.Cells(i, 1) <> maxifx And res_rangex.Cells(i, 1) <> minifx Then
            sy1 = sy1 + 1
        End If
        'Bottom face nodes
        If res_rangey.Cells(i, 1) = minify And res_rangez.Cells(i, 1) <> maxifz And res_rangez.Cells(i, 1) <> minifz _
                        And res_rangex.Cells(i, 1) <> maxifx And res_rangex.Cells(i, 1) <> minifx Then
            sy2 = sy2 + 1
        End If
```

```
Next
ReDim top(1 To sy1, 1 To 4)
ReDim bottom(1 To sy1, 1 To 4)
ReDim dist_3(1 To sy1)
ss5 = 1
ss6 = 1
For i = 1 To rows
    'Top face nodes
    If res_rangey.Cells(i, 1) = maxify And res_rangez.Cells(i, 1) <> maxifz And res_rangez.Cells(i, 1) <> minifz _
                And res_rangex.Cells(i, 1) <> maxifx And res_rangex.Cells(i, 1) <> minifx Then
        top(ss5, 2) = res_rangex.Cells(i, 1)
        top(ss5, 3) = res_rangey.Cells(i, 1)
        top(ss5, 4) = res_rangez.Cells(i, 1)
        top(ss5, 1) = nodes_range.Cells(i, 1)
        ss5 = ss5 + 1
    End If
    'Bottom face nodes
    If res_rangey.Cells(i, 1) = minify And res_rangez.Cells(i, 1) <> maxifz And res_rangez.Cells(i, 1) <> minifz _
                And res_rangex.Cells(i, 1) <> maxifx And res_rangex.Cells(i, 1) <> minifx Then
        bottom(ss6, 2) = res_rangex.Cells(i, 1)
        bottom(ss6, 3) = res_rangey.Cells(i, 1)
        bottom(ss6, 4) = res_rangez.Cells(i, 1)
        bottom(ss6, 1) = nodes_range.Cells(i, 1)
        ss6 = ss6 + 1
    End If
Next
```

```
    'calculate distances
    For q = 1 To sy1
        For h = 1 To sy1

            dist_3(h) = ((top(q, 2) - bottom(h, 2)) ^ 2 + _
                    (top(q, 3) - bottom(h, 3)) ^ 2 + (top(q, 4) - bottom(h, 4)) ^ 2) ^ 0.5
        Next
        'pair nodes
        dist_min = WorksheetFunction.Min(dist_3)
        r_min = WorksheetFunction.Match(dist_min, dist_3, 0)
        pairsht.Cells(4 + q, 11) = top(q, 1)
        pairsht.Cells(4 + q, 12) = bottom(r_min, 1)
        pairsht.Cells(4 + q, 13) = dist_min
    Next
'**********************END of Top bottom pairing**********************
End Sub
```

2. VBA Script pair_edge ()

```
Option Explicit

Option Base 1

Dim datasht, matchsht, pairsht As Worksheet

Sub pair_edge()

Dim res_range, res_rangex, res_rangey, res_rangez, nodes_range, dist_range As Range

Dim F_T(), F_Bo(), F_R(), F_L(), R_T(), T_L(), L_Bo(), R_Bo(), R_Ba(), Ba_L(), T_Ba(), Ba_Bo() As Double

Dim dist_1(), dist_2(), dist_3(), dist_4(), dist_5(), dist_6() As Double

Dim cols, rows As Long

Dim minifz, maxifz, minify, maxify, minifx, maxifx, dist, dist_min As Double

Dim i, j, w, k, n, q, h, ss1, ss11, ss2, ss22, ss3, ss33, ss4, ss44, ss5, ss55, ss6, ss66 As Integer

Dim sz1, sz11, sz2, sz22, sx1, sx11, sx2, sx22, sy2, sy22, sy1, sy11, r_min As Integer

Set datasht = Worksheets("RVE_NODES")

Set pairsht = Worksheets("PAIR")

    With datasht

    Set res_range = .Range(.Cells(3, 3), .Cells(11125, 6))

    End With

    'check rows and columns

    cols = res_range.Columns.Count

    rows = res_range.rows.Count

    pairsht.Cells(2, 2) = cols

    pairsht.Cells(3, 2) = rows

    Set res_rangex = res_range.Columns(2)

    Set res_rangey = res_range.Columns(3)

    Set res_rangez = res_range.Columns(4)

    'check rows and columns
```

```
    pairsht.Cells(2, 5) = res_rangez.Columns.Count
    pairsht.Cells(3, 5) = res_rangez.rows.Count
    Set nodes_range = res_range.Columns(1)
    minifz = Application.WorksheetFunction.Min(res_rangez)
    maxifz = Application.WorksheetFunction.Max(res_rangez)
    minify = Application.WorksheetFunction.Min(res_rangey)
    maxify = Application.WorksheetFunction.Max(res_rangey)
    minifx = Application.WorksheetFunction.Min(res_rangex)
    maxifx = Application.WorksheetFunction.Max(res_rangex)
'**********************VERTICES PAIR**************
For i = 1 To rows
'FTL
    If res_rangez.Cells(i, 1) = maxifz And res_rangey.Cells(i, 1) = maxify And res_rangex.Cells(i,
1) = minifx Then
        pairsht.Cells(5, 40) = nodes_range.Cells(i, 1)
    End If
'FTR
    If res_rangez.Cells(i, 1) = maxifz And res_rangey.Cells(i, 1) = maxify And res_rangex.Cells(i,
1) = maxifx Then
        pairsht.Cells(6, 40) = nodes_range.Cells(i, 1)
    End If
'FBoR
    If res_rangez.Cells(i, 1) = maxifz And res_rangey.Cells(i, 1) = minify And res_rangex.Cells(i,
1) = maxifx Then
        pairsht.Cells(7, 40) = nodes_range.Cells(i, 1)
    End If
'FBoL
    If res_rangez.Cells(i, 1) = maxifz And res_rangey.Cells(i, 1) = minify And res_rangex.Cells(i,
1) = minifx Then
        pairsht.Cells(8, 40) = nodes_range.Cells(i, 1)
```

```
   End If
'BaBoR
   If res_rangez.Cells(i, 1) = minifz And res_rangey.Cells(i, 1) = minify And res_rangex.Cells(i, 1) = maxifx Then
      pairsht.Cells(5, 42) = nodes_range.Cells(i, 1)
   End If
'BaBoL
   If res_rangez.Cells(i, 1) = minifz And res_rangey.Cells(i, 1) = minify And res_rangex.Cells(i, 1) = minifx Then
      pairsht.Cells(6, 42) = nodes_range.Cells(i, 1)
   End If
'BaTL
   If res_rangez.Cells(i, 1) = minifz And res_rangey.Cells(i, 1) = maxify And res_rangex.Cells(i, 1) = minifx Then
      pairsht.Cells (7, 42) = nodes_range.Cells(i, 1)
   End If
'BaTR
   If res_rangez.Cells(i, 1) = minifz And res_rangey.Cells(i, 1) = maxify And res_rangex.Cells(i, 1) = maxifx Then
      pairsht.Cells(8, 42) = nodes_range.Cells(i, 1)
   End If
Next
'**********BEGINING of F_T Ba_Bo pairing**********************
   For i = 1 To rows
      'F_T edge nodes
      If res_rangez.Cells(i, 1) = maxifz And res_rangey.Cells(i, 1) = maxify And res_rangex.Cells(i, 1) <> maxifx And res_rangex.Cells(i, 1) <> minifx Then
         sz1 = sz1 + 1
      End If
      'Ba_Bo edge nodes
```

```
    If res_rangez.Cells(i, 1) = minifz And res_rangey.Cells(i, 1) = minify And res_rangex.Cells(i, 1) <> maxifx And res_rangex.Cells(i, 1) <> minifx Then

        sz2 = sz2 + 1

    End If

Next

ReDim F_T(1 To sz1, 1 To 4)

ReDim Ba_Bo(1 To sz1, 1 To 4)

ReDim dist_1(1 To sz1)

ss1 = 1

ss2 = 1

For i = 1 To rows

    'F_T edge nodes

    If res_rangez.Cells(i, 1) = maxifz And res_rangey.Cells(i, 1) = maxify And res_rangex.Cells(i, 1) <> maxifx And res_rangex.Cells(i, 1) <> minifx Then

        F_T(ss1, 2) = res_rangex.Cells(i, 1)

        F_T(ss1, 3) = res_rangey.Cells(i, 1)

        F_T(ss1, 4) = res_rangez.Cells(i, 1)

        F_T(ss1, 1) = nodes_range.Cells(i, 1)

        ss1 = ss1 + 1

    End If

    'Ba_Bo edge nodes

    If res_rangez.Cells(i, 1) = minifz And res_rangey.Cells(i, 1) = minify And res_rangex.Cells(i, 1) <> maxifx And res_rangex.Cells(i, 1) <> minifx Then

        Ba_Bo(ss2, 2) = res_rangex.Cells(i, 1)

        Ba_Bo(ss2, 3) = res_rangey.Cells(i, 1)

        Ba_Bo(ss2, 4) = res_rangez.Cells(i, 1)

        Ba_Bo(ss2, 1) = nodes_range.Cells(i, 1)

        ss2 = ss2 + 1

    End If
```

```
    Next
    'calculate distances
    For j = 1 To sz1
        For w = 1 To sz1
            dist_1(w) = ((F_T(j, 2) - Ba_Bo(w, 2)) ^ 2 + _
                    (F_T(j, 3) - Ba_Bo(w, 3)) ^ 2 + (F_T(j, 4) - Ba_Bo(w, 4)) ^ 2) ^ 0.5
        Next
        'pair nodes
        dist_min = WorksheetFunction.Min(dist_1)
        r_min = WorksheetFunction.Match(dist_min, dist_1, 0)
        pairsht.Cells(4 + j, 15) = F_T(j, 1)
        pairsht.Cells(4 + j, 16) = Ba_Bo(r_min, 1)
        pairsht.Cells(4 + j, 17) = dist_min
    Next
'**************END of F_T Ba_Bo pairing************************

'******************BEGINING of F_Bo T_Ba pairing******************
    For i = 1 To rows

        'F_Bo edge nodes
        If res_rangez.Cells(i, 1) = maxifz And res_rangey.Cells(i, 1) = minify And res_rangex.Cells(i, 1) <> maxifx And res_rangex.Cells(i, 1) <> minifx Then
            sx1 = sx1 + 1
        End If
        'T_Ba edge nodes
        If res_rangez.Cells(i, 1) = minifz And res_rangey.Cells(i, 1) = maxify And res_rangex.Cells(i, 1) <> maxifx And res_rangex.Cells(i, 1) <> minifx Then
            sx2 = sx2 + 1
        End If
```

```
Next
ReDim F_Bo(1 To sx1, 1 To 4)
ReDim T_Ba(1 To sx1, 1 To 4)
ReDim dist_2(1 To sx1)
ss3 = 1
ss4 = 1
For i = 1 To rows
    'F_Bo edge nodes
    If res_rangez.Cells(i, 1) = maxifz And res_rangey.Cells(i, 1) = minify And res_rangex.Cells(i, 1) <> maxifx And res_rangex.Cells(i, 1) <> minifx Then
        F_Bo(ss3, 2) = res_rangex.Cells(i, 1)
        F_Bo(ss3, 3) = res_rangey.Cells(i, 1)
        F_Bo(ss3, 4) = res_rangez.Cells(i, 1)
        F_Bo(ss3, 1) = nodes_range.Cells(i, 1)
        ss3 = ss3 + 1
    End If
    'T_Ba edge nodes
    If res_rangez.Cells(i, 1) = minifz And res_rangey.Cells(i, 1) = maxify And res_rangex.Cells(i, 1) <> maxifx And res_rangex.Cells(i, 1) <> minifx Then
        T_Ba(ss4, 2) = res_rangex.Cells(i, 1)
        T_Ba(ss4, 3) = res_rangey.Cells(i, 1)
        T_Ba(ss4, 4) = res_rangez.Cells(i, 1)
        T_Ba(ss4, 1) = nodes_range.Cells(i, 1)
        ss4 = ss4 + 1
    End If
Next
'calculate distances
For k = 1 To sx1
    For n = 1 To sx1
```

```
        dist_2(n) = ((F_Bo(k, 2) - T_Ba(n, 2)) ^ 2 + _
                (F_Bo(k, 3) - T_Ba(n, 3)) ^ 2 + (F_Bo(k, 4) - T_Ba(n, 4)) ^ 2) ^ 0.5
    Next
    'pair nodes
    dist_min = WorksheetFunction.Min(dist_2)
    r_min = WorksheetFunction.Match(dist_min, dist_2, 0)
    pairsht.Cells(4 + k, 19) = F_Bo(k, 1)
    pairsht.Cells(4 + k, 20) = T_Ba(r_min, 1)
    pairsht.Cells(4 + k, 21) = dist_min
    **********BEGINING of F_R Ba_L pairing**********
  For i = 1 To rows
    'F_R edge nodes
    If res_rangex.Cells(i, 1) = maxifx And res_rangez.Cells(i, 1) = maxifz And res_rangey.Cells(i, 1) <> maxify And res_rangey.Cells(i, 1) <> minify Then
      sy1 = sy1 + 1
    End If
    'Ba_L edge nodes
    If res_rangex.Cells(i, 1) = minifx And res_rangez.Cells(i, 1) = minifz And res_rangey.Cells(i, 1) <> maxify And res_rangey.Cells(i, 1) <> minify Then
      sy2 = sy2 + 1
    End If
  Next
  ReDim F_R(1 To sy1, 1 To 4)
  ReDim Ba_L(1 To sy1, 1 To 4)
  ReDim dist_3(1 To sy1)
  ss5 = 1
  ss6 = 1
  For i = 1 To rows
    'F_R edge nodes
```

```
    If res_rangex.Cells(i, 1) = maxifx And res_rangez.Cells(i, 1) = maxifz And
res_rangey.Cells(i, 1) <> maxify And res_rangey.Cells(i, 1) <> minify Then

        F_R(ss5, 2) = res_rangex.Cells(i, 1)

        F_R(ss5, 3) = res_rangey.Cells(i, 1)

        F_R(ss5, 4) = res_rangez.Cells(i, 1)

        F_R(ss5, 1) = nodes_range.Cells(i, 1)

        ss5 = ss5 + 1

    End If

    'Ba_L edge nodes

    If res_rangex.Cells(i, 1) = minifx And res_rangez.Cells(i, 1) = minifz And res_rangey.Cells(i,
1) <> maxify And res_rangey.Cells(i, 1) <> minify Then

        Ba_L(ss6, 2) = res_rangex.Cells(i, 1)

        Ba_L(ss6, 3) = res_rangey.Cells(i, 1)

        Ba_L(ss6, 4) = res_rangez.Cells(i, 1)

        Ba_L(ss6, 1) = nodes_range.Cells(i, 1)

        ss6 = ss6 + 1

    End If

  Next

  'calculate distances

  For q = 1 To sy1

    For h = 1 To sy1

      dist_3(h) = ((F_R(q, 2) - Ba_L(h, 2)) ^ 2 + _

            (F_R(q, 3) - Ba_L(h, 3)) ^ 2 + (F_R(q, 4) - Ba_L(h, 4)) ^ 2) ^ 0.5

    Next

    'pair nodes

    dist_min = WorksheetFunction.Min(dist_3)

    r_min = WorksheetFunction.Match(dist_min, dist_3, 0)

    pairsht.Cells(4 + q, 23) = F_R(q, 1)

    pairsht.Cells(4 + q, 24) = Ba_L(r_min, 1)
```

```
        pairsht.Cells(4 + q, 25) = dist_min
    Next
'***************END of F_R Ba_L pairing*****************************


'***************BEGINING of F_L R_Ba pairing************************
    For i = 1 To rows
        'F_L edge nodes
        If res_rangez.Cells(i, 1) = maxifz And res_rangex.Cells(i, 1) = minifx And res_rangey.Cells(i, 1) <> maxify And res_rangey.Cells(i, 1) <> minify Then
            sz11 = sz11 + 1
        End If
        'R_Ba edge nodes
        If res_rangez.Cells(i, 1) = minifz And res_rangex.Cells(i, 1) = maxifx And res_rangey.Cells(i, 1) <> maxify And res_rangey.Cells(i, 1) <> minify Then
            sz22 = sz22 + 1
        End If
    Next
    ReDim F_L(1 To sz11, 1 To 4)
    ReDim R_Ba(1 To sz11, 1 To 4)
    ReDim dist_4(1 To sz11)
    ss11 = 1
    ss22 = 1
    For i = 1 To rows
        'F_L edge nodes
        If res_rangez.Cells(i, 1) = maxifz And res_rangex.Cells(i, 1) = minifx And res_rangey.Cells(i, 1) <> maxify And res_rangey.Cells(i, 1) <> minify Then
            F_L(ss11, 2) = res_rangex.Cells(i, 1)
            F_L(ss11, 3) = res_rangey.Cells(i, 1)
            F_L(ss11, 4) = res_rangez.Cells(i, 1)
```

```
            F_L(ss11, 1) = nodes_range.Cells(i, 1)
            ss11 = ss11 + 1
        End If
        'R_Ba edge nodes
        If res_rangez.Cells(i, 1) = minifz And res_rangex.Cells(i, 1) = maxifx And res_rangey.Cells(i, 1) <> maxify And res_rangey.Cells(i, 1) <> minify Then
            R_Ba(ss22, 2) = res_rangex.Cells(i, 1)
            R_Ba(ss22, 3) = res_rangey.Cells(i, 1)
            R_Ba(ss22, 4) = res_rangez.Cells(i, 1)
            R_Ba(ss22, 1) = nodes_range.Cells(i, 1)
            ss22 = ss22 + 1
        End If
    Next
    'calculate distances
    For j = 1 To sz11
        For w = 1 To sz11
            dist_4(w) = ((F_L(j, 2) - R_Ba(w, 2)) ^ 2 + _
                    (F_L(j, 3) - R_Ba(w, 3)) ^ 2 + (F_L(j, 4) - R_Ba(w, 4)) ^ 2) ^ 0.5
        Next
        'pair nodes
        dist_min = WorksheetFunction.Min(dist_4)
        r_min = WorksheetFunction.Match(dist_min, dist_4, 0)
        pairsht.Cells(4 + j, 27) = F_L(j, 1)
        pairsht.Cells(4 + j, 28) = R_Ba(r_min, 1)
        pairsht.Cells(4 + j, 29) = dist_min
    Next
'********************END of F_L R_Ba pairing************************


'*************BEGINING of R_T L_Bo pairin*************************
```

```
For i = 1 To rows
    'R_T edge nodes
    If res_rangex.Cells(i, 1) = maxifx And res_rangey.Cells(i, 1) = maxify And res_rangez.Cells(i, 1) <> maxifz And res_rangez.Cells(i, 1) <> minifz Then
        sx11 = sx11 + 1
    End If
    'L_Bo edge nodes
    If res_rangex.Cells(i, 1) = minifx And res_rangey.Cells(i, 1) = minify And res_rangez.Cells(i, 1) <> maxifz And res_rangez.Cells(i, 1) <> minifz Then
        sx22 = sx22 + 1
    End If
Next
ReDim R_T(1 To sx11, 1 To 4)
ReDim L_Bo(1 To sx11, 1 To 4)
ReDim dist_5(1 To sx11)
ss33 = 1
ss44 = 1
For i = 1 To rows
    'R_T edge nodes
    If res_rangex.Cells(i, 1) = maxifx And res_rangey.Cells(i, 1) = maxify And res_rangez.Cells(i, 1) <> maxifz And res_rangez.Cells(i, 1) <> minifz Then
        R_T(ss33, 2) = res_rangex.Cells(i, 1)
        R_T(ss33, 3) = res_rangey.Cells(i, 1)
        R_T(ss33, 4) = res_rangez.Cells(i, 1)
        R_T(ss33, 1) = nodes_range.Cells(i, 1)
        ss33 = ss33 + 1
    End If
    'L_Bo edge nodes
    If res_rangex.Cells(i, 1) = minifx And res_rangey.Cells(i, 1) = minify And res_rangez.Cells(i, 1) <> maxifz And res_rangez.Cells(i, 1) <> minifz Then
```

```
            L_Bo(ss44, 2) = res_rangex.Cells(i, 1)
            L_Bo(ss44, 3) = res_rangey.Cells(i, 1)
            L_Bo(ss44, 4) = res_rangez.Cells(i, 1)
            L_Bo(ss44, 1) = nodes_range.Cells(i, 1)
            ss44 = ss44 + 1
        End If
    Next
    'calculate distances

    For k = 1 To sx11
        For n = 1 To sx11
            dist_5(n) = ((R_T(k, 2) - L_Bo(n, 2)) ^ 2 + _
                    (R_T(k, 3) - L_Bo(n, 3)) ^ 2 + (R_T(k, 4) - L_Bo(n, 4)) ^ 2) ^ 0.5
        Next
        'pair nodes
        dist_min = WorksheetFunction.Min(dist_5)
        r_min = WorksheetFunction.Match(dist_min, dist_5, 0)
        pairsht.Cells(4 + k, 31) = R_T(k, 1)
        pairsht.Cells(4 + k, 32) = L_Bo(r_min, 1)
        pairsht.Cells(4 + k, 33) = dist_min
    Next
'******************END of R_T L_Bo pairing**************************

'**************BEGINING of T_L R_Bo pairing**************************
    For i = 1 To rows
        'T_L edge nodes
        If res_rangex.Cells(i, 1) = minifx And res_rangey.Cells(i, 1) = maxify And
res_rangez.Cells(i, 1) <> maxifz And res_rangez.Cells(i, 1) <> minifz Then
            sy11 = sy11 + 1
```

```
        End If
        'R_Bo edge nodes
        If res_rangex.Cells(i, 1) = maxifx And res_rangey.Cells(i, 1) = minify And res_rangez.Cells(i, 1) <> maxifz And res_rangez.Cells(i, 1) <> minifz Then
            sy22 = sy22 + 1
        End If
    Next
    ReDim T_L(1 To sy11, 1 To 4)
    ReDim R_Bo(1 To sy11, 1 To 4)
    ReDim dist_6(1 To sy11)
    ss55 = 1
    ss66 = 1
    For i = 1 To rows
        'T_L edge nodes
        If res_rangex.Cells(i, 1) = minifx And res_rangey.Cells(i, 1) = maxify And res_rangez.Cells(i, 1) <> maxifz And res_rangez.Cells(i, 1) <> minifz Then
            T_L(ss55, 2) = res_rangex.Cells(i, 1)
            T_L(ss55, 3) = res_rangey.Cells(i, 1)
            T_L(ss55, 4) = res_rangez.Cells(i, 1)
            T_L(ss55, 1) = nodes_range.Cells(i, 1)
            ss55 = ss55 + 1
        End If
        'R_Bo edge nodes
        If res_rangex.Cells(i, 1) = maxifx And res_rangey.Cells(i, 1) = minify And res_rangez.Cells(i, 1) <> maxifz And res_rangez.Cells(i, 1) <> minifz Then
            R_Bo(ss66, 2) = res_rangex.Cells(i, 1)
            R_Bo(ss66, 3) = res_rangey.Cells(i, 1)
            R_Bo(ss66, 4) = res_rangez.Cells(i, 1)
            R_Bo(ss66, 1) = nodes_range.Cells(i, 1)
```

```
            ss66 = ss66 + 1
        End If
    Next
    'calculate distances
    For q = 1 To sy11
        For h = 1 To sy11
            dist_6(h) = ((T_L(q, 2) - R_Bo(h, 2)) ^ 2 + _
                    (T_L(q, 3) - R_Bo(h, 3)) ^ 2 + (T_L(q, 4) - R_Bo(h, 4)) ^ 2) ^ 0.5
        Next
        'pair nodes
        dist_min = WorksheetFunction.Min(dist_6)
        r_min = WorksheetFunction.Match(dist_min, dist_6, 0)
        pairsht.Cells(4 + q, 35) = T_L(q, 1)
        pairsht.Cells(4 + q, 36) = R_Bo(r_min, 1)
        pairsht.Cells(4 + q, 37) = dist_min
    Next
'*****************END of T_L R_Bo pairing**********************
End Sub
```

3. VBA Script micro_bdf_file ()

```
Option Explicit

Dim loop_incr, meso_lim_loop, micro_lim_loop As Integer

Sub micro_bdf_file(loop_incr)

Dim j, fail As Integer

Dim const_law_sht, MPC_EQ_DISP_sht As Worksheet

Set const_law_sht = Worksheets("MICRO_CONST_TANG")

Application.ScreenUpdating = False

'loop_incr = 1

For j = 1 To 6

'CHECK FOR FILE EXISTANCE AND CREATE ONE IF IT DOESN'T EXIST

'Open "C:\MsFEM_Project\GRAND_MASTER\REPAIR_INCR_" & CStr(loop_incr) & "\MESO_CRIT_" & CStr(loop_incr) & _

"\MESO_RVE_" & CStr(const_law_sht.Cells(9 + (4 * j), 4)) & ".bdf" For Output As #1

If Len(Dir("C:\MsFEM_Project\GRAND_MASTER\REPAIR_INCR_" & CStr(loop_incr) & "\MESO_CRIT_" & CStr(loop_incr) & _

   "\MICRO_CRIT_" & CStr(loop_incr), vbDirectory)) = 0 Then

   MkDir "C:\MsFEM_Project\GRAND_MASTER\REPAIR_INCR_" & CStr(loop_incr) & "\MESO_CRIT_" & CStr(loop_incr) & _

   "\MICRO_CRIT_" & CStr(loop_incr)

End If

Open "C:\MsFEM_Project\GRAND_MASTER\REPAIR_INCR_" & CStr(loop_incr) & "\MESO_CRIT_" & CStr(loop_incr) & _

   "\MICRO_CRIT_" & CStr(loop_incr) & "\MICRO_RVE_" & CStr(const_law_sht.Cells(9 + (4 * j), 4)) & ".bdf" For Output As #1

Print #1, "$ NASTRAN MICRO RVE input file created by Ikara MsFEM"

Print #1, "$ WITS Research Masters Project "

Print #1, "$ Direct Text Input for Nastran System Cell Section"
```

```
Print #1, "$ Linear Static Analysis, Database"
Print #1, "SOL 101"
Print #1, "CEND"
Print #1, "TITLE = MSC.Nastran MICRO RVE MsFEM job"
Print #1, "ECHO = NONE"
Print #1, "MPC = 4"
Print #1, "SUBCASE 1"
Print #1, "   SUBTITLE = Default"
Print #1, "   SPC = 2"
'Print #1, "   DISPLACEMENT(SORT1,REAL)=ALL"
Print #1, "   SPCFORCES(SORT1,REAL)=ALL"
Print #1, "   STRESS(SORT1,REAL,VONMISES,BILIN)=ALL"
Print #1, "   AUTOSPC(NOPRINT)=YES"
Print #1, "BEGIN BULK"
Print #1, "PARAM    POST    0"
Print #1, "PARAM   PRTMAXIM YES"
'ELEMENTS AND PROPERTIES

Dim elem_list_sht, node_list_sht As Variant
Dim elem_range_1, elem_range_2, elem_range_3, node_range As Variant
Dim elem_range_1f, elem_range_2f, elem_range_3f As Variant
Set elem_list_sht = Worksheets("MICRO_RVE_ELEM_PROPS")
   With elem_list_sht
      Set elem_range_1 = Range(.Cells(3, 14), .Cells(10754, 21))
      Set elem_range_2 = Range(.Cells(3, 3), .Cells(6914, 10))
      Set elem_range_3 = Range(.Cells(3, 25), .Cells(770, 32))
      Set elem_range_1f = Range(.Cells(3, 47), .Cells(10754, 54))
      Set elem_range_2f = Range(.Cells(3, 36), .Cells(6914, 43))
```

```
        Set elem_range_3f = Range(.Cells(3, 58), .Cells(770, 65))
    End With
Set node_list_sht = Worksheets("MICRO_RVE_NODES")
    With node_list_sht
        Set node_range = Range(.Cells(3, 3), .Cells(10985, 6))
    End With
Dim num_elem_1, num_elem_2, num_elem_3, i, num_nodes As Integer
Print #1, "$"
Print #1, "$ Elements and Element Properties for region : prop_fibre"
Print #1, "PSOLID,1,1,0"
Print #1, "$ Pset: ' prop_fibre' will be imported as: ' psolid.1'"
Print #1, "$"
Print #1, "$"
Print #1, "$ Elements and Element Properties for region : failed_prop_fibre"
Print #1, "PSOLID,11,11,0"
Print #1, "$ Pset: ' prop_fibre' will be imported as: ' psolid.11'"
Print #1, "$"
num_elem_1 = 10752
fail = 0
For i = 1 To num_elem_1 Step 2
    If elem_list_sht.Cells(6000 + 0.5 * (i - 1), 77 + loop_incr * 8) = "" Then
    Print #1, "CHEXA," + CStr(elem_range_1(i, 1)) + "," + CStr(elem_range_1(i, 2)) + "," + CStr(elem_range_1(i, 3)) + "," _
            + CStr(elem_range_1(i, 4)) + "," + CStr(elem_range_1(i, 5)) + "," + CStr(elem_range_1(i, 6)) + "," _
            + CStr(elem_range_1(i, 7)) + "," + CStr(elem_range_1(i, 8)) + ","
    Print #1, "+," + CStr(elem_range_1(i + 1, 1)) + "," + CStr(elem_range_1(i + 1, 2))
    Else
```

```
   Print #1, "CHEXA," + CStr(elem_range_1f(i, 1)) + "," + CStr(elem_range_1f(i, 2)) + "," + CStr(elem_range_1f(i, 3)) + "," _

            + CStr(elem_range_1f(i, 4)) + "," + CStr(elem_range_1f(i, 5)) + "," + CStr(elem_range_1f(i, 6)) + "," _

            + CStr(elem_range_1f(i, 7)) + "," + CStr(elem_range_1f(i, 8)) + ","

   Print #1, "+," + CStr(elem_range_1f(i + 1, 1)) + "," + CStr(elem_range_1f(i + 1, 2))

   End If

Next i

Print #1, "$"

Print #1, "$ Elements and Element Properties for region : prop_matrix"

Print #1, "PSOLID,2,2,0"

Print #1, "$ Pset: ' prop_matrix' will be imported as: ' psolid.2"'

Print #1, "$"

Print #1, "$"

Print #1, "$ Elements and Element Properties for region : failed_prop_matrix"

Print #1, "PSOLID,22,22,0"

Print #1, "$ Pset: ' prop_matrix' will be imported as: ' psolid.22"'

Print #1, "$"

num_elem_2 = 6912

For i = 1 To num_elem_2 Step 2

   If elem_list_sht.Cells(6000 + 0.5 * (i - 1), 75 + loop_incr * 8) = "" Then

   Print #1, "CHEXA," + CStr(elem_range_2(i, 1)) + "," + CStr(elem_range_2(i, 2)) + "," + CStr(elem_range_2(i, 3)) + "," _

            + CStr(elem_range_2(i, 4)) + "," + CStr(elem_range_2(i, 5)) + "," + CStr(elem_range_2(i, 6)) + "," _

            + CStr(elem_range_2(i, 7)) + "," + CStr(elem_range_2(i, 8)) + ","

   Print #1, "+," + CStr(elem_range_2(i + 1, 1)) + "," + CStr(elem_range_2(i + 1, 2))

   Else

    Print #1, "CHEXA," + CStr(elem_range_2f(i, 1)) + "," + CStr(elem_range_2f(i, 2)) + "," + CStr(elem_range_2f(i, 3)) + "," _
```

```
        + CStr(elem_range_2f(i, 4)) + "," + CStr(elem_range_2f(i, 5)) + "," + CStr(elem_range_2f(i, 6)) + "," _

        + CStr(elem_range_2f(i, 7)) + "," + CStr(elem_range_2f(i, 8)) + ","

    Print #1, "+," + CStr(elem_range_2f(i + 1, 1)) + "," + CStr(elem_range_2f(i + 1, 2))

    End If

Next i

Print #1, "$"

Print #1, "$ Elements and Element Properties for region : prop_interface"

Print #1, "PSOLID,3,3,0"

Print #1, "$ Pset: ' prop_interface' will be imported as: ' psolid.3'"

Print #1, "$"

Print #1, "$"

Print #1, "$ Elements and Element Properties for region : failed_prop_interface"

Print #1, "PSOLID,33,33,0"

Print #1, "$ Pset: ' prop_interface' will be imported as: ' psolid.33'"

Print #1, "$"

num_elem_3 = 768

For i = 1 To num_elem_3 Step 2

    If elem_list_sht.Cells(6000 + 0.5 * (i - 1), 73 + loop_incr * 8) = "" Then

    Print #1, "CHEXA," + CStr(elem_range_3(i, 1)) + "," + CStr(elem_range_3(i, 2)) + "," + CStr(elem_range_3(i, 3)) + "," _

        + CStr(elem_range_3(i, 4)) + "," + CStr(elem_range_3(i, 5)) + "," + CStr(elem_range_3(i, 6)) + "," _

        + CStr(elem_range_3(i, 7)) + "," + CStr(elem_range_3(i, 8)) + ","

    Print #1, "+," + CStr(elem_range_3(i + 1, 1)) + "," + CStr(elem_range_3(i + 1, 2))

    Else

    Print #1, "CHEXA," + CStr(elem_range_3f(i, 1)) + "," + CStr(elem_range_3f(i, 2)) + "," + CStr(elem_range_3f(i, 3)) + "," _

        + CStr(elem_range_3f(i, 4)) + "," + CStr(elem_range_3f(i, 5)) + "," + CStr(elem_range_3f(i, 6)) + "," _
```

```
            + CStr(elem_range_3f(i, 7)) + "," + CStr(elem_range_3f(i, 8)) + ","
    Print #1, "+," + CStr(elem_range_3f(i + 1, 1)) + "," + CStr(elem_range_3f(i + 1, 2))
    End If
Next i
'MATERIAL PROPERTIES
Print #1, "$"
Print #1, "$ Referenced Material Records"
Print #1, "$ Material Record : mat_fibre"
Print #1, "$ Description of Material :"
'Print #1, "MAT1,1,213.7,,.2"
Print #1, "MAT9,1,138,,, , , ,138,"
Print #1, "+,,,,,2130,"
Print #1, "+,113.5,,,113.5,,55"
Print #1, "$ Material Record : mat_matrix"
Print #1, "$ Description of Material :"
'Print #1, "MAT1,2,3.45,,.35"
Print #1, "MAT9,2,55.35,29.79,29.79, , , ,55.35,"
Print #1, "+,29.79,,,,55.35,"
Print #1, "+,13,,,13,,13"
Print #1, "$ Material Record : mat_interface"
Print #1, "$ Description of Material :"
'Print #1, "MAT1,3,3.45,,.35"
Print #1, "MAT9,3,55.35,29.79,29.79, , , ,55.35,"
Print #1, "+,29.79,,,,55.35,"
Print #1, "+,13,,,13,,13"
'Print #1, "$"
'Print #1, "$ VERIFIED AND TESTED FOR MATRIX AND INTERFACE"
'Print #1, "MAT9,1,55.35,29.79,29.79, , , ,55.35,"
```

```
'Print #1, "+,29.79,,,,55.35,"
'Print #1, "+,13,,,13,,13"
'Print #1, "MAT9,2,G11,G12,G13,,,,G22,"
'Print #1, "+,G23,,,,G33.,"
'Print #1, "+,G44,,,G55,,G66"
'Print #1, "$ VERIFIED AND TESTED FOR FIBRE NOT TO BE USED IN MsFEM"
'Print #1, "MAT9,3,550,400,1800, , , ,550,"
'Print #1, "+,1800,,,,8200,"
'Print #1, "+,113.5,,,113.5,,55"
'Print #1, "$ VERIFIED AND TESTED FOR FIBRE"
'Print #1, "MAT9,1,138,,, , , ,138,"
'Print #1, "+,,,,,2130,"
'Print #1, "+,113.5,,,113.5,,55"
'FAILED MATERIAL PROPERTIES
Print #1, "$ **********FAILED MATERIAL PROPERTIES**********"
Print #1, "$ Referenced Material Records"
Print #1, "$ Material Record : mat_fibre"
Print #1, "$ Description of Material :"
Print #1, "MAT9,11,0.138,,, , , ,0.138,"
Print #1, "+,,,,,2.130,"
Print #1, "+,0.1135,,,0.1135,,0.055"
Print #1, "$ Material Record : mat_matrix"
Print #1, "$ Description of Material :"
Print #1, "MAT9,22,0.05535,0.02979,0.02979, , , ,0.05535,"
Print #1, "+,0.02979,,,,0.05535,"
Print #1, "+,0.013,,,0.013,,0.013"
Print #1, "$ Material Record : mat_interface"
Print #1, "$ Description of Material :"
```

```
Print #1, "MAT9,33,0.05535,0.02979,0.02979, , , ,0.05535,"

Print #1, "+,0.02979,,,,0.05535,"

Print #1, "+,0.013,,,0.013,,0.013"

Print #1, "$"

'UPDATE CRITICAL STRAIN ENTRIES

Set MPC_EQ_DISP_sht = Worksheets("MICRO_EQUIV_DISP")

MPC_EQ_DISP_sht.Cells(3, 6) = const_law_sht.Cells(9 + (4 * j), 11) 'E11

MPC_EQ_DISP_sht.Cells(3, 7) = const_law_sht.Cells(9 + (4 * j), 12) 'E21

MPC_EQ_DISP_sht.Cells(3, 8) = const_law_sht.Cells(9 + (4 * j), 13) 'E31

MPC_EQ_DISP_sht.Cells(4, 7) = const_law_sht.Cells(10 + (4 * j), 12) 'E22

MPC_EQ_DISP_sht.Cells(4, 8) = const_law_sht.Cells(10 + (4 * j), 13) 'E32

MPC_EQ_DISP_sht.Cells(5, 8) = const_law_sht.Cells(11 + (4 * j), 13) 'E33

'INSERT MPC CONSTRAINTS

micro_mpc_pairs

'INSERT NODES

Print #1, "$ Nodes of the Entire Model"

num_nodes = 10983

For i = 1 To num_nodes

   Print #1, "GRID" + "," + CStr(node_range(i, 1)) + ",," + CStr(node_range(i, 2)) + "," + CStr(node_range(i, 3)) + "," + _

            CStr(node_range(i, 4))

Next i

Print #1, "$"

Print #1, "$"

Print #1, "$ Loads for Load Case : Default"

Print #1, "SPCADD,2,1"

Print #1, "$ Displacement Constraints of Load Set : fix_123456"

Print #1, "SPC1,1,123456,945"

Print #1, "$ Referenced Coordinate Frames"
```

```
Print #1, "ENDDATA 7b83e1dd"


Close #1
Next j
Application.ScreenUpdating = True
End Sub
```

4. VBA Script micro_mpc_pair ()

```
Sub micro_mpc_pairs()
Dim fso As Object
Dim oufile As Object
Dim i, num_mpc As Integer
Dim mpc_rows As String
Dim pairsht, datasht, mpc_pair As Variant
Dim res_range, res_rangex, res_rangey, res_rangez, nodes_range As Range
' initialise
Set pairsht = Worksheets("MICRO_PAIR")
Set datasht = Worksheets("MICRO_RVE_NODES")
Set mpc_pair = Worksheets("MICRO_EQUIV_DISP")
   With datasht
   Set res_range = Range(.Cells(3, 3), .Cells(10985, 6))
   End With
   Set res_rangex = res_range.Columns(2)
   Set res_rangey = res_range.Columns(3)
   Set res_rangez = res_range.Columns(4)
   Set nodes_range = res_range.Columns(1)
num_mpc = 5187
'Open "F:\MULTISCALE MODELLING\WITS_MATERIAL\realistic_micro\REAL_MICRO_E11.bdf" For Output As #1
Print #1, ("$MPC's for periodic boundary conditions")
Print #1, "$"
' print mpc block
Print #1, "$ Multipoint Constraints of the Entire Model"
Print #1, "MPCADD,4,1,2,3,4,5,6,7"
```

```
For i = 8 To num_mpc
   If (i Mod 8) = 7 Then
      mpc_rows = mpc_rows + "," + CStr(i)
      Print #1, mpc_rows
      mpc_rows = ""
   Else
      mpc_rows = mpc_rows + "," + CStr(i)
   End If
Next i
Print #1, "$ mpc_rows"
' print mpcs for vertices
   'print info for vertex 1
Dim FTR, BaBoL, FBoR, BaTL, FBoL, BaTR, FTL, BaBoR As Integer 'vertices node numbers
Dim disp_V1_dr1, disp_V1_dr2, disp_V1_dr3 As Double 'mpc displacements for vertex mpc_link_1
Print #1, "$vertices FTR-BaBoL"
FTR = 6970
BaBoL = 3429
disp_V1_dr1 = mpc_pair.Cells(9, 13).Value
disp_V1_dr2 = mpc_pair.Cells(10, 13).Value
disp_V1_dr3 = mpc_pair.Cells(11, 13).Value
'compare location components
comp1 = WorksheetFunction.Match(6970, nodes_range, 1)
comp2 = WorksheetFunction.Match(3429, nodes_range, 1)
xcomp1 = WorksheetFunction.Index(res_rangex, comp1)
xcomp2 = WorksheetFunction.Index(res_rangex, comp2)
ycomp1 = WorksheetFunction.Index(res_rangey, comp1)
ycomp2 = WorksheetFunction.Index(res_rangey, comp2)
zcomp1 = WorksheetFunction.Index(res_rangez, comp1)
```

```
zcomp2 = WorksheetFunction.Index(res_rangez, comp2)
If xcomp1 > xcomp2 Then
   Print #1, "MPCY,1," + CStr(FTR) + ",1,1.0," + CStr(disp_V1_dr1)
   Print #1, ",," + CStr(BaBoL) + ",1,-1.0"
Else
   Print #1, "MPCY,1," + CStr(BaBoL) + ",1,1.0," + CStr(disp_V1_dr1)
   Print #1, ",," + CStr(FTR) + ",1,-1.0"
End If
If ycomp1 > ycomp2 Then
   Print #1, "MPCY,2," + CStr(FTR) + ",2,1.0," + CStr(disp_V1_dr2)
   Print #1, ",," + CStr(BaBoL) + ",2,-1.0"
Else
   Print #1, "MPCY,2," + CStr(BaBoL) + ",2,1.0," + CStr(disp_V1_dr2)
   Print #1, ",," + CStr(FTR) + ",2,-1.0"
End If
If zcomp1 > zcomp2 Then
   Print #1, "MPCY,3," + CStr(FTR) + ",3,1.0," + CStr(disp_V1_dr3)
   Print #1, ",," + CStr(BaBoL) + ",3,-1.0"
Else
   Print #1, "MPCY,3," + CStr(BaBoL) + ",3,1.0," + CStr(disp_V1_dr3)
   Print #1, ",," + CStr(FTR) + ",3,-1.0"
End If
   'print info for vertex 2
Dim disp_V2_dr1, disp_V2_dr2, disp_V2_dr3 As Double 'mpc displacements for vertex mpc_link_2
Print #1, "$vertices FBoR-BaTL"
FBoR = 9917
BaTL = 91
disp_V2_dr1 = mpc_pair.Cells(27, 13).Value
```

```
disp_V2_dr2 = mpc_pair.Cells(28, 13).Value
disp_V2_dr3 = mpc_pair.Cells(29, 13).Value
'compare location components
comp1 = WorksheetFunction.Match(9917, nodes_range, 1)
comp2 = WorksheetFunction.Match(91, nodes_range, 1)
xcomp1 = WorksheetFunction.Index(res_rangex, comp1)
xcomp2 = WorksheetFunction.Index(res_rangex, comp2)
ycomp1 = WorksheetFunction.Index(res_rangey, comp1)
ycomp2 = WorksheetFunction.Index(res_rangey, comp2)
zcomp1 = WorksheetFunction.Index(res_rangez, comp1)
zcomp2 = WorksheetFunction.Index(res_rangez, comp2)
If xcomp1 > xcomp2 Then
Print #1, "MPCY,4," + CStr(FBoR) + ",1,1.0," + CStr(disp_V2_dr1)
Print #1, ",," + CStr(BaTL) + ",1,-1.0"
Else
Print #1, "MPCY,4," + CStr(BaTL) + ",1,1.0," + CStr(disp_V2_dr1)
Print #1, ",," + CStr(FBoR) + ",1,-1.0"
End If
If ycomp1 > ycomp2 Then
Print #1, "MPCY,5," + CStr(FBoR) + ",2,1.0," + CStr(disp_V2_dr2)
Print #1, ",," + CStr(BaTL) + ",2,-1.0"
Else
Print #1, "MPCY,5," + CStr(BaTL) + ",2,1.0," + CStr(disp_V2_dr2)
Print #1, ",," + CStr(FBoR) + ",2,-1.0"
End If
If zcomp1 > zcomp2 Then
Print #1, "MPCY,6," + CStr(FBoR) + ",3,1.0," + CStr(disp_V2_dr3)
Print #1, ",," + CStr(BaTL) + ",3,-1.0"
```

```
Else
Print #1, "MPCY,6," + CStr(BaTL) + ",3,1.0," + CStr(disp_V2_dr3)
Print #1, ",," + CStr(FBoR) + ",3,-1.0"
End If
   'print info for vertex 3
Dim disp_V3_dr1, disp_V3_dr2, disp_V3_dr3 As Double 'mpc displacements for vertex mpc_link_3
Print #1, "$vertices FBoL-BaTR"
FBoL = 4023
BaTR = 6376
disp_V3_dr1 = mpc_pair.Cells(15, 13).Value
disp_V3_dr2 = mpc_pair.Cells(16, 13).Value
disp_V3_dr3 = mpc_pair.Cells(17, 13).Value
'compare location components
comp1 = WorksheetFunction.Match(4023, nodes_range, 1)
comp2 = WorksheetFunction.Match(6376, nodes_range, 1)
xcomp1 = WorksheetFunction.Index(res_rangex, comp1)
xcomp2 = WorksheetFunction.Index(res_rangex, comp2)
ycomp1 = WorksheetFunction.Index(res_rangey, comp1)
ycomp2 = WorksheetFunction.Index(res_rangey, comp2)
zcomp1 = WorksheetFunction.Index(res_rangez, comp1)
zcomp2 = WorksheetFunction.Index(res_rangez, comp2)
If xcomp1 > xcomp2 Then
   Print #1, "MPCY,7," + CStr(FBoL) + ",1,1.0," + CStr(disp_V3_dr1)
   Print #1, ",," + CStr(BaTR) + ",1,-1.0"
Else
   Print #1, "MPCY,7," + CStr(BaTR) + ",1,1.0," + CStr(disp_V3_dr1)
   Print #1, ",," + CStr(FBoL) + ",1,-1.0"
End If
```

```
If ycomp1 > ycomp2 Then
   Print #1, "MPCY,8," + CStr(FBoL) + ",2,1.0," + CStr(disp_V3_dr2)
   Print #1, ",," + CStr(BaTR) + ",2,-1.0"
Else
   Print #1, "MPCY,8," + CStr(BaTR) + ",2,1.0," + CStr(disp_V3_dr2)
   Print #1, ",," + CStr(FBoL) + ",2,-1.0"
End If
If zcomp1 > zcomp2 Then
   Print #1, "MPCY,9," + CStr(FBoL) + ",3,1.0," + CStr(disp_V3_dr3)
   Print #1, ",," + CStr(BaTR) + ",3,-1.0"
Else
   Print #1, "MPCY,9," + CStr(BaTR) + ",3,1.0," + CStr(disp_V3_dr3)
   Print #1, ",," + CStr(FBoL) + ",3,-1.0"
End If
   'print info for vertex link 4
Dim disp_V4_dr1, disp_V4_dr2, disp_V4_dr3 As Double 'mpc displacements for vertex mpc_link_4
Print #1, "$vertices FTL-BaBoR"
FTL = 685
BaBoR = 9323
disp_V4_dr1 = mpc_pair.Cells(21, 13).Value
disp_V4_dr2 = mpc_pair.Cells(22, 13).Value
disp_V4_dr3 = mpc_pair.Cells(23, 13).Value
'compare location components
comp1 = WorksheetFunction.Match(685, nodes_range, 1)
comp2 = WorksheetFunction.Match(9323, nodes_range, 1)
xcomp1 = WorksheetFunction.Index(res_rangex, comp1)
xcomp2 = WorksheetFunction.Index(res_rangex, comp2)
ycomp1 = WorksheetFunction.Index(res_rangey, comp1)
```

```
ycomp2 = WorksheetFunction.Index(res_rangey, comp2)
zcomp1 = WorksheetFunction.Index(res_rangez, comp1)
zcomp2 = WorksheetFunction.Index(res_rangez, comp2)
If xcomp1 > xcomp2 Then
    Print #1, "MPCY,10," + CStr(FTL) + ",1,1.0," + CStr(disp_V4_dr1)
    Print #1, ",," + CStr(BaBoR) + ",1,-1.0"
Else
    Print #1, "MPCY,10," + CStr(BaBoR) + ",1,1.0," + CStr(disp_V4_dr1)
    Print #1, ",," + CStr(FTL) + ",1,-1.0"
End If
If ycomp1 > ycomp2 Then
    Print #1, "MPCY,11," + CStr(FTL) + ",2,1.0," + CStr(disp_V4_dr2)
    Print #1, ",," + CStr(BaBoR) + ",2,-1.0"
Else
    Print #1, "MPCY,11," + CStr(BaBoR) + ",2,1.0," + CStr(disp_V4_dr2)
    Print #1, ",," + CStr(FTL) + ",2,-1.0"
End If
If zcomp1 > zcomp2 Then
    Print #1, "MPCY,12," + CStr(FTL) + ",3,1.0," + CStr(disp_V4_dr3)
    Print #1, ",," + CStr(BaBoR) + ",3,-1.0"
Else
    Print #1, "MPCY,12," + CStr(BaBoR) + ",3,1.0," + CStr(disp_V4_dr3)
    Print #1, ",," + CStr(FTL) + ",3,-1.0"
End If
' print mpcs for edges
Print #1, "$ Edges FT_BaBo"
Dim numpts As Integer
Dim E_1_pts(1 To 15, 1 To 2) As Integer
```

```
Dim disp_E1_dr1, disp_E1_dr2, disp_E1_dr3, stp As Double ' Disp for edge 1
' Initialize
numpts = 15
For i = 1 To numpts
   E_1_pts(i, 1) = pairsht.Cells(4 + i, 15).Value
   E_1_pts(i, 2) = pairsht.Cells(4 + i, 16).Value
Next i
disp_E1_dr1 = mpc_pair.Cells(35, 13).Value
disp_E1_dr2 = mpc_pair.Cells(36, 13).Value
disp_E1_dr3 = mpc_pair.Cells(37, 13).Value
stp = 0
'compare location components
comp1 = WorksheetFunction.Match(686, nodes_range, 1)
comp2 = WorksheetFunction.Match(3430, nodes_range, 1)
xcomp1 = WorksheetFunction.Index(res_rangex, comp1)
xcomp2 = WorksheetFunction.Index(res_rangex, comp2)
ycomp1 = WorksheetFunction.Index(res_rangey, comp1)
ycomp2 = WorksheetFunction.Index(res_rangey, comp2)
zcomp1 = WorksheetFunction.Index(res_rangez, comp1)
zcomp2 = WorksheetFunction.Index(res_rangez, comp2)
For i = 1 To numpts
   If xcomp1 >= xcomp2 Then
   ' print x dir mpc
      Print #1, "MPCY," + CStr(12 + i + stp) + "," + CStr(E_1_pts(i, 1)) + ",1,1.0," _
            + CStr(disp_E1_dr1)
      Print #1, ",," + CStr(E_1_pts(i, 2)) + ",1,-1.0"
   Else
   ' print x dir mpc
```

```
    Print #1, "MPCY," + CStr(12 + i + stp) + "," + CStr(E_1_pts(i, 2)) + ",1,1.0," _
            + CStr(disp_E1_dr1)
    Print #1, ",," + CStr(E_1_pts(i, 1)) + ",1,-1.0"
  End If
  If ycomp1 >= ycomp2 Then
  ' print y dir mpc
    Print #1, "MPCY," + CStr(12 + i + 1 + stp) + "," + CStr(E_1_pts(i, 1)) + ",2,1.0," _
            + CStr(disp_E1_dr2)
    Print #1, ",," + CStr(E_1_pts(i, 2)) + ",2,-1.0"
  Else
  ' print y dir mpc
    Print #1, "MPCY," + CStr(12 + i + 1 + stp) + "," + CStr(E_1_pts(i, 2)) + ",2,1.0," _
            + CStr(disp_E1_dr2)
    Print #1, ",," + CStr(E_1_pts(i, 1)) + ",2,-1.0"
  End If
  If zcomp1 >= zcomp2 Then
  ' print z dir mpc
    Print #1, "MPCY," + CStr(12 + i + 2 + stp) + "," + CStr(E_1_pts(i, 1)) + ",3,1.0," _
            + CStr(disp_E1_dr3)
    Print #1, ",," + CStr(E_1_pts(i, 2)) + ",3,-1.0"
  Else
  ' print z dir mpc
    Print #1, "MPCY," + CStr(12 + i + 2 + stp) + "," + CStr(E_1_pts(i, 2)) + ",3,1.0," _
          + CStr(disp_E1_dr3)
    Print #1, ",," + CStr(E_1_pts(i, 1)) + ",3,-1.0"
  End If
  stp = stp + 2
Next i
```

```
Dim nxt_mpc As Integer
nxt_mpc = 12 + i + stp - 1
' print mpcs for edges
Print #1, "$ Edges FBo_TBa"
Dim E_2_pts(1 To 15, 1 To 2) As Integer
Dim disp_E2_dr1, disp_E2_dr2, disp_E2_dr3 As Double ' Disp for edge 2
' Initialize
numpts = 15
For i = 1 To numpts
    E_2_pts(i, 1) = pairsht.Cells(4 + i, 19).Value
    E_2_pts(i, 2) = pairsht.Cells(4 + i, 20).Value
Next i
disp_E2_dr1 = mpc_pair.Cells(40, 13).Value
disp_E2_dr2 = mpc_pair.Cells(41, 13).Value
disp_E2_dr3 = mpc_pair.Cells(42, 13).Value
stp = 0
'compare location components
comp1 = WorksheetFunction.Match(4025, nodes_range, 1)
comp2 = WorksheetFunction.Match(93, nodes_range, 1)
xcomp1 = WorksheetFunction.Index(res_rangex, comp1)
xcomp2 = WorksheetFunction.Index(res_rangex, comp2)
ycomp1 = WorksheetFunction.Index(res_rangey, comp1)
ycomp2 = WorksheetFunction.Index(res_rangey, comp2)
zcomp1 = WorksheetFunction.Index(res_rangez, comp1)
zcomp2 = WorksheetFunction.Index(res_rangez, comp2)
For i = 1 To numpts
    If xcomp1 > xcomp2 Then
    ' print x dir mpc
```

```
    Print #1, "MPCY," + CStr(nxt_mpc + i + stp) + "," + CStr(E_2_pts(i, 1)) + ",1,1.0," _
            + CStr(disp_E2_dr1)
    Print #1, ",," + CStr(E_2_pts(i, 2)) + ",1,-1.0"
Else
' print x dir mpc
    Print #1, "MPCY," + CStr(nxt_mpc + i + stp) + "," + CStr(E_2_pts(i, 2)) + ",1,1.0," _
            + CStr(disp_E2_dr1)
    Print #1, ",," + CStr(E_2_pts(i, 1)) + ",1,-1.0"
End If
If ycomp1 > ycomp2 Then
' print y dir mpc
    Print #1, "MPCY, " + CStr(nxt_mpc + i + 1 + stp) + "," + CStr(E_2_pts(i, 1)) + ",2,1.0," _
            + CStr(disp_E2_dr2)
    Print #1, ",," + CStr(E_2_pts(i, 2)) + ",2,-1.0"
Else
    ' print y dir mpc
    Print #1, "MPCY," + CStr(nxt_mpc + i + 1 + stp) + "," + CStr(E_2_pts(i, 2)) + ",2,1.0," _
            + CStr(disp_E2_dr2)
    Print #1, ",," + CStr(E_2_pts(i, 1)) + ",2,-1.0"
End If
If zcomp1 > zcomp2 Then
' print z dir mpc
    Print #1, "MPCY," + CStr(nxt_mpc + i + 2 + stp) + "," + CStr(E_2_pts(i, 1)) + ",3,1.0," _
            + CStr(disp_E2_dr3)
    Print #1, ",," + CStr(E_2_pts(i, 2)) + ",3,-1.0"
Else
 ' print z dir mpc
    Print #1, "MPCY," + CStr(nxt_mpc + i + 2 + stp) + "," + CStr(E_2_pts(i, 2)) + ",3,1.0," _
```

```
            + CStr(disp_E2_dr3)
        Print #1, ",," + CStr(E_2_pts(i, 1)) + ",3,-1.0"
    End If
    stp = stp + 2
Next i
nxt_mpc = nxt_mpc + i + stp - 1
' print mpcs for edges
Print #1, "$ Edges FR_BaL"
Dim E_3_pts(1 To 15, 1 To 2) As Integer
Dim disp_E3_dr1, disp_E3_dr2, disp_E3_dr3 As Double ' Disp for edge 3
' Initialize
numpts = 15
For i = 1 To numpts
    E_3_pts(i, 1) = pairsht.Cells(4 + i, 23).Value
    E_3_pts(i, 2) = pairsht.Cells(4 + i, 24).Value
Next i
disp_E3_dr1 = mpc_pair.Cells(45, 13).Value
disp_E3_dr2 = mpc_pair.Cells(46, 13).Value
disp_E3_dr3 = mpc_pair.Cells(47, 13).Value
stp = 0
'compare location components
comp1 = WorksheetFunction.Match(9226, nodes_range, 1)
comp2 = WorksheetFunction.Match(2738, nodes_range, 1)
xcomp1 = WorksheetFunction.Index(res_rangex, comp1)
xcomp2 = WorksheetFunction.Index(res_rangex, comp2)
ycomp1 = WorksheetFunction.Index(res_rangey, comp1)
ycomp2 = WorksheetFunction.Index(res_rangey, comp2)
zcomp1 = WorksheetFunction.Index(res_rangez, comp1)
```

```
zcomp2 = WorksheetFunction.Index(res_rangez, comp2)
For i = 1 To numpts
   If xcomp1 >= xcomp2 Then
   ' print x dir mpc
      Print #1, "MPCY," + CStr(nxt_mpc + i + stp) + "," + CStr(E_3_pts(i, 1)) + ",1,1.0," _
            + CStr(disp_E3_dr1)
      Print #1, ",," + CStr(E_3_pts(i, 2)) + ",1,-1.0"
   Else
   ' print x dir mpc
      Print #1, "MPCY," + CStr(nxt_mpc + i + stp) + "," + CStr(E_3_pts(i, 2)) + ",1,1.0," _
            + CStr(disp_E3_dr1)
      Print #1, ",," + CStr(E_3_pts(i, 1)) + ",1,-1.0"
   End If
   If ycomp1 >= ycomp2 Then
   ' print y dir mpc
      Print #1, "MPCY," + CStr(nxt_mpc + i + 1 + stp) + "," + CStr(E_3_pts(i, 1)) + ",2,1.0," _
            + CStr(disp_E3_dr2)
      Print #1, ",," + CStr(E_3_pts(i, 2)) + ",2,-1.0"
   Else
    ' print y dir mpc
      Print #1, "MPCY," + CStr(nxt_mpc + i + 1 + stp) + "," + CStr(E_3_pts(i, 2)) + ",2,1.0," _
            + CStr(disp_E3_dr2)
      Print #1, ",," + CStr(E_3_pts(i, 1)) + ",2,-1.0"
   End If
   If zcomp1 >= zcomp2 Then
   ' print z dir mpc
      Print #1, "MPCY," + CStr(nxt_mpc + i + 2 + stp) + "," + CStr(E_3_pts(i, 1)) + ",3,1.0," _
            + CStr(disp_E3_dr3)
```

```
        Print #1, ",," + CStr(E_3_pts(i, 2)) + ",3,-1.0"
    Else
    ' print z dir mpc
        Print #1, "MPCY," + CStr(nxt_mpc + i + 2 + stp) + "," + CStr(E_3_pts(i, 2)) + ",3,1.0," _
            + CStr(disp_E3_dr3)
        Print #1, ",," + CStr(E_3_pts(i, 1)) + ",3,-1.0"
    End If
    stp = stp + 2
Next i
nxt_mpc = nxt_mpc + i + stp - 1
' print mpcs for edges
Print #1, "$ Edges FL_RBa"
Dim E_4_pts(1 To 15, 1 To 2) As Integer
Dim disp_E4_dr1, disp_E4_dr2, disp_E4_dr3 As Double ' Disp for edge 4
' Initialize
numpts = 15
For i = 1 To numpts
    E_4_pts(i, 1) = pairsht.Cells(4 + i, 27).Value
    E_4_pts(i, 2) = pairsht.Cells(4 + i, 28).Value
Next i
disp_E4_dr1 = mpc_pair.Cells(50, 13).Value
disp_E4_dr2 = mpc_pair.Cells(51, 13).Value
disp_E4_dr3 = mpc_pair.Cells(52, 13).Value
stp = 0
'compare location components
comp1 = WorksheetFunction.Match(3332, nodes_range, 1)
comp2 = WorksheetFunction.Match(8746, nodes_range, 1)
xcomp1 = WorksheetFunction.Index(res_rangex, comp1)
```

```
xcomp2 = WorksheetFunction.Index(res_rangex, comp2)
ycomp1 = WorksheetFunction.Index(res_rangey, comp1)
ycomp2 = WorksheetFunction.Index(res_rangey, comp2)
zcomp1 = WorksheetFunction.Index(res_rangez, comp1)
zcomp2 = WorksheetFunction.Index(res_rangez, comp2)
For i = 1 To numpts
   If xcomp1 > xcomp2 Then
   ' print x dir mpc
      Print #1, "MPCY," + CStr(nxt_mpc + i + stp) + "," + CStr(E_4_pts(i, 1)) + ",1,1.0," _
            + CStr(disp_E4_dr1)
      Print #1, ",," + CStr(E_4_pts(i, 2)) + ",1,-1.0"
   Else
      ' print x dir mpc
      Print #1, "MPCY," + CStr(nxt_mpc + i + stp) + "," + CStr(E_4_pts(i, 2)) + ",1,1.0," _
            + CStr(disp_E4_dr1)
      Print #1, ",," + CStr(E_4_pts(i, 1)) + ",1,-1.0"
   End If
   If ycomp1 > ycomp2 Then
   ' print y dir mpc
      Print #1, "MPCY," + CStr(nxt_mpc + i + 1 + stp) + "," + CStr(E_4_pts(i, 1)) + ",2,1.0," _
            + CStr(disp_E4_dr2)
      Print #1, ",," + CStr(E_4_pts(i, 2)) + ",2,-1.0"
   Else
   ' print y dir mpc
      Print #1, "MPCY," + CStr(nxt_mpc + i + 1 + stp) + "," + CStr(E_4_pts(i, 2)) + ",2,1.0," _
            + CStr(disp_E4_dr2)
      Print #1, ",," + CStr(E_4_pts(i, 1)) + ",2,-1.0"
   End If
```

```
    If zcomp1 > zcomp2 Then
    ' print z dir mpc
        Print #1, "MPCY," + CStr(nxt_mpc + i + 2 + stp) + "," + CStr(E_4_pts(i, 1)) + ",3,1.0," _
            + CStr(disp_E4_dr3)
        Print #1, ",," + CStr(E_4_pts(i, 2)) + ",3,-1.0"
    Else
    ' print z dir mpc
        Print #1, "MPCY," + CStr(nxt_mpc + i + 2 + stp) + "," + CStr(E_4_pts(i, 2)) + ",3,1.0," _
            + CStr(disp_E4_dr3)
        Print #1, ",," + CStr(E_4_pts(i, 1)) + ",3,-1.0"
    End If
    stp = stp + 2
Next i
nxt_mpc = nxt_mpc + i + stp - 1
' print mpcs for edges
Print #1, "$ Edges RT_LBo"
Dim E_5_pts(1 To 5, 1 To 2) As Integer
Dim disp_E5_dr1, disp_E5_dr2, disp_E5_dr3 As Double ' Disp for edge 5
' Initialize
numpts = 5
For i = 1 To numpts
    E_5_pts(i, 1) = pairsht.Cells(4 + i, 31).Value
    E_5_pts(i, 2) = pairsht.Cells(4 + i, 32).Value
Next i
disp_E5_dr1 = mpc_pair.Cells(55, 13).Value
disp_E5_dr2 = mpc_pair.Cells(56, 13).Value
disp_E5_dr3 = mpc_pair.Cells(57, 13).Value
stp = 0
```

```
'numpts = 1
'compare location components
comp1 = WorksheetFunction.Match(6574, nodes_range, 1)
comp2 = WorksheetFunction.Match(3627, nodes_range, 1)
xcomp1 = WorksheetFunction.Index(res_rangex, comp1)
xcomp2 = WorksheetFunction.Index(res_rangex, comp2)
ycomp1 = WorksheetFunction.Index(res_rangey, comp1)
ycomp2 = WorksheetFunction.Index(res_rangey, comp2)
zcomp1 = WorksheetFunction.Index(res_rangez, comp1)
zcomp2 = WorksheetFunction.Index(res_rangez, comp2)
For i = 1 To numpts
   If xcomp1 >= xcomp2 Then
   ' print x dir mpc
      Print #1, "MPCY," + CStr(nxt_mpc + i + stp) + "," + CStr(E_5_pts(i, 1)) + ",1,1.0," _
            + CStr(disp_E5_dr1)
      Print #1, ",," + CStr(E_5_pts(i, 2)) + ",1,-1.0"
   Else
   ' print x dir mpc
      Print #1, "MPCY," + CStr(nxt_mpc + i + stp) + "," + CStr(E_5_pts(i, 2)) + ",1,1.0," _
            + CStr(disp_E5_dr1)
      Print #1, ",," + CStr(E_5_pts(i, 1)) + ",1,-1.0"
   End If
   If ycomp1 >= ycomp2 Then
   ' print y dir mpc
      Print #1, "MPCY," + CStr(nxt_mpc + i + 1 + stp) + "," + CStr(E_5_pts(i, 1)) + ",2,1.0," _
            + CStr(disp_E5_dr2)
      Print #1, ",," + CStr(E_5_pts(i, 2)) + ",2,-1.0"
   Else
```

```
    ' print y dir mpc
        Print #1, "MPCY," + CStr(nxt_mpc + i + 1 + stp) + "," + CStr(E_5_pts(i, 2)) + ",2,1.0," _
            + CStr(disp_E5_dr2)
        Print #1, ",," + CStr(E_5_pts(i, 1)) + ",2,-1.0"
    End If
    If zcomp1 >= zcomp2 Then
    ' print z dir mpc
        Print #1, "MPCY," + CStr(nxt_mpc + i + 2 + stp) + "," + CStr(E_5_pts(i, 1)) + ",3,1.0," _
            + CStr(disp_E5_dr3)
        Print #1, ",," + CStr(E_5_pts(i, 2)) + ",3,-1.0"
    Else
    ' print z dir mpc
        Print #1, "MPCY," + CStr(nxt_mpc + i + 2 + stp) + "," + CStr(E_5_pts(i, 2)) + ",3,1.0," _
            + CStr(disp_E5_dr3)
        Print #1, ",," + CStr(E_5_pts(i, 1)) + ",3,-1.0"
    End If
    stp = stp + 2
Next i
nxt_mpc = nxt_mpc + i + stp - 1
' print mpcs for edges
Print #1, "$ Edges TL_RBo"
Dim E_6_pts(1 To 5, 1 To 2) As Integer
Dim disp_E6_dr1, disp_E6_dr2, disp_E6_dr3 As Double ' Disp for edge 6
' Initialize
numpts = 5
For i = 1 To numpts
    E_6_pts(i, 1) = pairsht.Cells(4 + i, 35).Value
    E_6_pts(i, 2) = pairsht.Cells(4 + i, 36).Value
```

```
Next i
disp_E6_dr1 = mpc_pair.Cells(60, 13).Value
disp_E6_dr2 = mpc_pair.Cells(61, 13).Value
disp_E6_dr3 = mpc_pair.Cells(62, 13).Value
stp = 0
'numpts = 1
'compare location components
comp1 = WorksheetFunction.Match(289, nodes_range, 1)
comp2 = WorksheetFunction.Match(9521, nodes_range, 1)
xcomp1 = WorksheetFunction.Index(res_rangex, comp1)
xcomp2 = WorksheetFunction.Index(res_rangex, comp2)
ycomp1 = WorksheetFunction.Index(res_rangey, comp1)
ycomp2 = WorksheetFunction.Index(res_rangey, comp2)
zcomp1 = WorksheetFunction.Index(res_rangez, comp1)
zcomp2 = WorksheetFunction.Index(res_rangez, comp2)
For i = 1 To numpts
   If xcomp1 > xcomp2 Then
   ' print x dir mpc
      Print #1, "MPCY," + CStr(nxt_mpc + i + stp) + "," + CStr(E_6_pts(i, 1)) + ",1,1.0," _
            + CStr(disp_E6_dr1)
      Print #1, ",," + CStr(E_6_pts(i, 2)) + ",1,-1.0"
   Else
   ' print x dir mpc
      Print #1, "MPCY," + CStr(nxt_mpc + i + stp) + "," + CStr(E_6_pts(i, 2)) + ",1,1.0," _
            + CStr(disp_E6_dr1)
      Print #1, ",," + CStr(E_6_pts(i, 1)) + ",1,-1.0"
   End If
   If ycomp1 > ycomp2 Then
```

```
    ' print y dir mpc
        Print #1, "MPCY," + CStr(nxt_mpc + i + 1 + stp) + "," + CStr(E_6_pts(i, 1)) + ",2,1.0," _
            + CStr(disp_E6_dr2)
        Print #1, ",," + CStr(E_6_pts(i, 2)) + ",2,-1.0"
    Else
    ' print y dir mpc
        Print #1, "MPCY," + CStr(nxt_mpc + i + 1 + stp) + "," + CStr(E_6_pts(i, 2)) + ",2,1.0," _
            + CStr(disp_E6_dr2)
        Print #1, ",," + CStr(E_6_pts(i, 1)) + ",2,-1.0"
    End If
    If zcomp1 > zcomp2 Then
    ' print z dir mpc
        Print #1, "MPCY," + CStr(nxt_mpc + i + 2 + stp) + "," + CStr(E_6_pts(i, 1)) + ",3,1.0," _
            + CStr(disp_E6_dr3)
        Print #1, ",," + CStr(E_6_pts(i, 2)) + ",3,-1.0"
    Else
    ' print z dir mpc
        Print #1, "MPCY," + CStr(nxt_mpc + i + 2 + stp) + "," + CStr(E_6_pts(i, 2)) + ",3,1.0," _
            + CStr(disp_E6_dr3)
        Print #1, ",," + CStr(E_6_pts(i, 1)) + ",3,-1.0"
    End If
    stp = stp + 2
Next i
nxt_mpc = nxt_mpc + i + stp - 1
' print mpcs for surfaces
Print #1, "$surfaces T_B"
Dim S_1_pts(1 To 75, 1 To 2) As Integer
Dim disp_S1_dr1, disp_S1_dr2, disp_S1_dr3 As Double ' Disp for surface T_B
```

```
' Initialize
numpts = 75
For i = 1 To numpts
    S_1_pts(i, 1) = pairsht.Cells(4 + i, 11).Value
    S_1_pts(i, 2) = pairsht.Cells(4 + i, 12).Value
Next i
disp_S1_dr1 = mpc_pair.Cells(68, 13).Value
disp_S1_dr2 = mpc_pair.Cells(69, 13).Value
disp_S1_dr3 = mpc_pair.Cells(70, 13).Value
stp = 0
'numpts = 1
For i = 1 To numpts
    'compare location components
    comp1 = WorksheetFunction.Match(S_1_pts(i, 1), nodes_range, 1)
    comp2 = WorksheetFunction.Match(S_1_pts(i, 2), nodes_range, 1)
    xcomp1 = WorksheetFunction.Index(res_rangex, comp1)
    xcomp2 = WorksheetFunction.Index(res_rangex, comp2)
    ycomp1 = WorksheetFunction.Index(res_rangey, comp1)
    ycomp2 = WorksheetFunction.Index(res_rangey, comp2)
    zcomp1 = WorksheetFunction.Index(res_rangez, comp1)
    zcomp2 = WorksheetFunction.Index(res_rangez, comp2)
    If xcomp1 > xcomp2 Then
    ' print x dir mpc
        Print #1, "MPCY," + CStr(nxt_mpc + i + stp) + "," + CStr(S_1_pts(i, 1)) + ",1,1.0," _
              + CStr(disp_S1_dr1)
        Print #1, ",," + CStr(S_1_pts(i, 2)) + ",1,-1.0"
    Else
    ' print x dir mpc
```

```
    Print #1, "MPCY," + CStr(nxt_mpc + i + stp) + "," + CStr(S_1_pts(i, 2)) + ",1,1.0," _
        + CStr(disp_S1_dr1)
    Print #1, ",," + CStr(S_1_pts(i, 1)) + ",1,-1.0"
  End If
  If ycomp1 > ycomp2 Then
  ' print y dir mpc
    Print #1, "MPCY," + CStr(nxt_mpc + i + 1 + stp) + "," + CStr(S_1_pts(i, 1)) + ",2,1.0," _
        + CStr(disp_S1_dr2)
    Print #1, ",," + CStr(S_1_pts(i, 2)) + ",2,-1.0"
  Else
  ' print y dir mpc
    Print #1, "MPCY," + CStr(nxt_mpc + i + 1 + stp) + "," + CStr(S_1_pts(i, 2)) + ",2,1.0," _
        + CStr(disp_S1_dr2)
    Print #1, ",," + CStr(S_1_pts(i, 1)) + ",2,-1.0"
  End If
  If zcomp1 > zcomp2 Then
  ' print z dir mpc
    Print #1, "MPCY," + CStr(nxt_mpc + i + 2 + stp) + "," + CStr(S_1_pts(i, 1)) + ",3,1.0," _
        + CStr(disp_S1_dr3)
    Print #1, ",," + CStr(S_1_pts(i, 2)) + ",3,-1.0"
  Else
  ' print z dir mpc
    Print #1, "MPCY," + CStr(nxt_mpc + i + 2 + stp) + "," + CStr(S_1_pts(i, 2)) + ",3,1.0," _
        + CStr(disp_S1_dr3)
    Print #1, ",," + CStr(S_1_pts(i, 1)) + ",3,-1.0"
  End If
  stp = stp + 2
Next i
```

```
nxt_mpc = nxt_mpc + i + stp - 1
' print mpcs for surfaces
Print #1, "$surfaces L_R"
Dim S_2_pts(1 To 75, 1 To 2) As Integer
Dim disp_S2_dr1, disp_S2_dr2, disp_S2_dr3 As Double ' Disp for surface L_R
' Initialize
numpts = 75
For i = 1 To numpts
    S_2_pts(i, 1) = pairsht.Cells(4 + i, 7).Value

    S_2_pts(i, 2) = pairsht.Cells(4 + i, 8).Value
Next i
disp_S2_dr1 = mpc_pair.Cells(73, 13).Value
disp_S2_dr2 = mpc_pair.Cells(74, 13).Value
disp_S2_dr3 = mpc_pair.Cells(75, 13).Value
stp = 0
'numpts = 1
For i = 1 To numpts
    'compare location components
    comp1 = WorksheetFunction.Match(S_2_pts(i, 1), nodes_range, 1)
    comp2 = WorksheetFunction.Match(S_2_pts(i, 2), nodes_range, 1)
    xcomp1 = WorksheetFunction.Index(res_rangex, comp1)
    xcomp2 = WorksheetFunction.Index(res_rangex, comp2)
    ycomp1 = WorksheetFunction.Index(res_rangey, comp1)
    ycomp2 = WorksheetFunction.Index(res_rangey, comp2)
    zcomp1 = WorksheetFunction.Index(res_rangez, comp1)
    zcomp2 = WorksheetFunction.Index(res_rangez, comp2)
    If xcomp1 > xcomp2 Then
```

```
' print x dir mpc
   Print #1, "MPCY," + CStr(nxt_mpc + i + stp) + "," + CStr(S_2_pts(i, 1)) + ",1,1.0," _
         + CStr(disp_S2_dr1)
   Print #1, ",," + CStr(S_2_pts(i, 2)) + ",1,-1.0"
Else
' print x dir mpc
   Print #1, "MPCY," + CStr(nxt_mpc + i + stp) + "," + CStr(S_2_pts(i, 2)) + ",1,1.0," _
         + CStr(disp_S2_dr1)
   Print #1, ",," + CStr(S_2_pts(i, 1)) + ",1,-1.0"
End If
If ycomp1 > ycomp2 Then
' print y dir mpc
   Print #1, "MPCY," + CStr(nxt_mpc + i + 1 + stp) + "," + CStr(S_2_pts(i, 1)) + ",2,1.0," _
         + CStr(disp_S2_dr2)
   Print #1, ",," + CStr(S_2_pts(i, 2)) + ",2,-1.0"
Else
' print y dir mpc
   Print #1, "MPCY," + CStr(nxt_mpc + i + 1 + stp) + "," + CStr(S_2_pts(i, 2)) + ",2,1.0," _
         + CStr(disp_S2_dr2)
   Print #1, ",," + CStr(S_2_pts(i, 1)) + ",2,-1.0"
End If
If zcomp1 > zcomp2 Then
' print z dir mpc
   Print #1, "MPCY," + CStr(nxt_mpc + i + 2 + stp) + "," + CStr(S_2_pts(i, 1)) + ",3,1.0," _
         + CStr(disp_S2_dr3)
   Print #1, ",," + CStr(S_2_pts(i, 2)) + ",3,-1.0"
Else
' print z dir mpc
```

```
        Print #1, "MPCY," + CStr(nxt_mpc + i + 2 + stp) + "," + CStr(S_2_pts(i, 2)) + ",3,1.0," _
            + CStr(disp_S2_dr3)
        Print #1, ",," + CStr(S_2_pts(i, 1)) + ",3,-1.0"
    End If
    stp = stp + 2
Next i
nxt_mpc = nxt_mpc + i + stp - 1
' print mpcs for surfaces
Print #1, "$surfaces F_B"
Dim S_3_pts(1 To 1505, 1 To 2) As Integer
Dim disp_S3_dr1, disp_S3_dr2, disp_S3_dr3 As Double ' Disp for surface F_B
' Initialize
numpts = 1505
For i = 1 To numpts
    S_3_pts(i, 1) = pairsht.Cells(4 + i, 3).Value
    S_3_pts(i, 2) = pairsht.Cells(4 + i, 4).Value
Next i
disp_S3_dr1 = mpc_pair.Cells(78, 13).Value
disp_S3_dr2 = mpc_pair.Cells(79, 13).Value
disp_S3_dr3 = mpc_pair.Cells(80, 13).Value
stp = 0
'compare location components
comp1 = WorksheetFunction.Match(S_3_pts(1, 1), nodes_range, 1)
comp2 = WorksheetFunction.Match(S_3_pts(1, 2), nodes_range, 1)
xcomp1 = WorksheetFunction.Index(res_rangex, comp1)
xcomp2 = WorksheetFunction.Index(res_rangex, comp2)
ycomp1 = WorksheetFunction.Index(res_rangey, comp1)
ycomp2 = WorksheetFunction.Index(res_rangey, comp2)
```

```
zcomp1 = WorksheetFunction.Index(res_rangez, comp1)
zcomp2 = WorksheetFunction.Index(res_rangez, comp2)
For i = 1 To numpts
   If xcomp1 > xcomp2 Then
   ' print x dir mpc
      Print #1, "MPCY," + CStr(nxt_mpc + i + stp) + "," + CStr(S_3_pts(i, 1)) + ",1,--1.0," _
            + CStr(disp_S3_dr1)
      Print #1, ",," + CStr(S_3_pts(i, 2)) + ",1,-1.0"
   Else
   ' print x dir mpc
      Print #1, "MPCY," + CStr(nxt_mpc + i + stp) + "," + CStr(S_3_pts(i, 2)) + ",1,1.0," _
            + CStr(disp_S3_dr1)
      Print #1, ",," + CStr(S_3_pts(i, 1)) + ",1,-1.0"
   End If
   If ycomp1 > ycomp2 Then
   ' print y dir mpc
      Print #1, "MPCY," + CStr(nxt_mpc + i + 1 + stp) + "," + CStr(S_3_pts(i, 1)) + ",2,1.0," _
            + CStr(disp_S3_dr2)
      Print #1, ",," + CStr(S_3_pts(i, 2)) + ",2,-1.0"
   Else
      ' print y dir mpc
      Print #1, "MPCY," + CStr(nxt_mpc + i + 1 + stp) + "," + CStr(S_3_pts(i, 2)) + ",2,1.0," _
            + CStr(disp_S3_dr2)
      Print #1, ",," + CStr(S_3_pts(i, 1)) + ",2,-1.0"
   End If
   If zcomp1 > zcomp2 Then
   ' print z dir mpc
      Print #1, "MPCY," + CStr(nxt_mpc + i + 2 + stp) + "," + CStr(S_3_pts(i, 1)) + ",3,1.0," _
```

```
            + CStr(disp_S3_dr3)
        Print #1, ",," + CStr(S_3_pts(i, 2)) + ",3,-1.0"
    Else
      ' print z dir mpc
        Print #1, "MPCY," + CStr(nxt_mpc + i + 2 + stp) + "," + CStr(S_3_pts(i, 2)) + ",3,1.0," _
            + CStr(disp_S3_dr3)
        Print #1, ",," + CStr(S_3_pts(i, 1)) + ",3,-1.0"
    End If
    stp = stp + 2
Next i
'Close #1
End Sub
```

5. **VBA Script micro_con_tangent ()**

```
Option Base 1
Sub micro_con_tangent(loop_incr)
'READ F06 FILE
Dim file1 As String
Dim volumebook, stressbook As Workbooks
Dim volumesht, stresssht, resultsht As Worksheet
Dim volume_sum_rep, volume_mult, r As Variant
Dim norm_x, norm_y, norm_z, shr_xy, shr_yz, shr_zx, von_mises As Double
Dim numelem, elem, j, k As Integer
Dim i As Double
numelem = 9216
'loop_incr = 1
Dim stress_matrix(9216, 8) As Variant
Dim range_row() As Variant
Dim range_colm() As Variant
Set resultsht = Worksheets("MICRO_CONST_TANG")
Application.ScreenUpdating = False
For j = 1 To 6 ' loop through results files
file1 = "C:\MsFEM_Project\GRAND_MASTER\REPAIR_INCR_" & CStr(loop_incr) & "\MESO_CRIT_" & CStr(loop_incr) & _
   "\MICRO_CRIT_" & CStr(loop_incr) & "\MICRO_RVE_" & CStr(resultsht.Cells(10 + (4 * j), 4)) & ".f06"
Workbooks.OpenText FileName:= _
      file1 _
      , Origin:=437, StartRow:=1, DataType:=xlDelimited, TextQualifier:= _
      xlDoubleQuote, ConsecutiveDelimiter:=True, Tab:=True, Semicolon:=False, _
      Comma:=False, Space:=True, Other:=False, FieldInfo:=Array(Array(1, 1), _
```

```
        Array(2, 1), Array(3, 1), Array(4, 1), Array(5, 1), Array(6, 1), Array(7, 1), Array(8, 1), _
        Array(9, 1), Array(10, 1), Array(11, 1), Array(12, 1), Array(13, 1)), _
        TrailingMinusNumbers:=True
Set stresssht = Workbooks("MICRO_RVE_" & CStr(resultsht.Cells(10 + (4 * j), 4))). _
Worksheets("MICRO_RVE_" & CStr(resultsht.Cells(10 + (4 * j), 4)))
k = 1
For i = 1 To 340000
    If stresssht.Cells(i, 2).Value = "CENTER" Then
        'elem
            stress_matrix(k, 1) = stresssht.Cells((i - 1), 2)
                If stresssht.Cells(i - 1, 2).Value = "ELEMENT-ID" Then
            'elem
                    stress_matrix(k, 1) = stresssht.Cells(i - 8, 2)
                Else
                End If
        'norm_x
            stress_matrix(k, 2) = stresssht.Cells(i, 4)
        'norm_y
            stress_matrix(k, 3) = stresssht.Cells(i + 1, 3)
        'norm_z
            stress_matrix(k, 4) = stresssht.Cells(i + 2, 3)
        'shr_xy
            stress_matrix(k, 5) = stresssht.Cells(i, 6)
        'shr_yz
            stress_matrix(k, 6) = stresssht.Cells(i + 1, 5)
        'shr_zx
            stress_matrix(k, 7) = stresssht.Cells(i + 2, 5)
        'von_mises
```

```
            stress_matrix(k, 8) = stresssht.Cells(i, 13)
               If stresssht.Cells(i, 13).Value = "" Then
            'von_mises
                  stress_matrix(k, 8) = stresssht.Cells(i, 12)

               ElseIf stresssht.Cells(i, 12).Value = "" Then
                  stress_matrix(k, 8) = stresssht.Cells(i, 11)
               Else
               End If
            k = k + 1
         Else
         End If
Next i
Workbooks("MICRO_RVE_" & CStr(resultsht.Cells(10 + (4 * j), 4)) & ".f06").Close SaveChanges:=False
'CALCULATE COLUMNS
Set volumesht = Worksheets("MICRO_ELEM_VOLUME")
volume_sum_rep = 1 / (WorksheetFunction.Sum(Range(volumesht.Cells(3, 3), volumesht.Cells(2 + numelem, 3)).Value))
Dim Fn As Object
Set Fn = Application.WorksheetFunction
ReDim range_row(1 To 1, 1 To 9216)
range_row = (Range(volumesht.Cells(3, 3), volumesht.Cells(2 + numelem, 3)))
For r = 1 To 6
range_colm = Fn.Index(stress_matrix, , 1 + r)
volume_mult = Fn.MMult(Fn.Transpose(range_row), range_colm)
resultsht.Cells(3 + r, 13 + j) = Fn.MMult(volume_mult, volume_sum_rep)
Next r
Next j
```

```
Application.ScreenUpdating = True
End Sub
```

6. VBA Script meso_mpc_pairs ()

```
Sub meso_mpc_pairs()
Dim fso As Object
Dim oufile As Object
Dim i, num_mpc As Integer
Dim mpc_rows As String
Dim pairsht, datasht, mpc_pair As Variant
Dim res_range, res_rangex, res_rangey, res_rangez, nodes_range As Range
' initialise
Set pairsht = Worksheets("MESO_PAIR")
Set datasht = Worksheets("MESO_RVE_NODES")
Set mpc_pair = Worksheets("MESO_EQUIV_DISP")
   With datasht
   Set res_range = Range(.Cells(3, 3), .Cells(44423, 6))
   End With
   Set res_rangex = res_range.Columns(2)
   Set res_rangey = res_range.Columns(3)
   Set res_rangez = res_range.Columns(4)
   Set nodes_range = res_range.Columns(1)
num_mpc = 12903
'Open "J:\MULTISCALE MODELLING\WITS_MATERIAL\realistic_meso\MESO_MPC_E11.txt" For Output As #1
Print #1, ("$MPC's for periodic boundary conditions")
Print #1, "$"
' print mpc block
Print #1, "$ Multipoint Constraints of the Entire Model"
Print #1, "MPCADD,4,1,2,3,4,5,6,7"
```

```
For i = 8 To num_mpc
   If (i Mod 8) = 7 Then
      mpc_rows = mpc_rows + "," + CStr(i)
      Print #1, mpc_rows
      mpc_rows = ""
   Else
      mpc_rows = mpc_rows + "," + CStr(i)
   End If
Next i
Print #1, "$"
Print #1, "$ mpc_rows"
' print mpcs for vertices
   'print info for vertex 1
Dim FTR, BaBoL, FBoR, BaTL, FBoL, BaTR, FTL, BaBoR As Integer 'vertices node numbers
Dim disp_V1_dr1, disp_V1_dr2, disp_V1_dr3 As Double 'mpc displacements for vertex mpc_link_1
Print #1, "$vertices FTR-BaBoL"
FTR = 61401
BaBoL = 5610
disp_V1_dr1 = mpc_pair.Cells(9, 13).Value
disp_V1_dr2 = mpc_pair.Cells(10, 13).Value
disp_V1_dr3 = mpc_pair.Cells(11, 13).Value
'compare location components
comp1 = WorksheetFunction.Match(61401, nodes_range, 1)
comp2 = WorksheetFunction.Match(5610, nodes_range, 1)
xcomp1 = WorksheetFunction.Index(res_rangex, comp1)
xcomp2 = WorksheetFunction.Index(res_rangex, comp2)
ycomp1 = WorksheetFunction.Index(res_rangey, comp1)
ycomp2 = WorksheetFunction.Index(res_rangey, comp2)
```

```
zcomp1 = WorksheetFunction.Index(res_rangez, comp1)

zcomp2 = WorksheetFunction.Index(res_rangez, comp2)

If xcomp1 > xcomp2 Then

   Print #1, "MPCY,1," + CStr(FTR) + ",1,1.0," + CStr(disp_V1_dr1)

   Print #1, ",," + CStr(BaBoL) + ",1,-1.0"

Else

   Print #1, "MPCY,1," + CStr(BaBoL) + ",1,1.0," + CStr(disp_V1_dr1)

   Print #1, ",," + CStr(FTR) + ",1,-1.0"

End If

If ycomp1 > ycomp2 Then

   Print #1, "MPCY,2," + CStr(FTR) + ",2,1.0," + CStr(disp_V1_dr2)

   Print #1, ",," + CStr(BaBoL) + ",2,-1.0"

Else

   Print #1, "MPCY,2," + CStr(BaBoL) + ",2,1.0," + CStr(disp_V1_dr2)

   Print #1, ",," + CStr(FTR) + ",2,-1.0"

End If

If zcomp1 > zcomp2 Then

   Print #1, "MPCY,3," + CStr(FTR) + ",3,1.0," + CStr(disp_V1_dr3)

   Print #1, ",," + CStr(BaBoL) + ",3,-1.0"

Else

   Print #1, "MPCY,3," + CStr(BaBoL) + ",3,1.0," + CStr(disp_V1_dr3)

   Print #1, ",," + CStr(FTR) + ",3,-1.0"

End If

   'print info for vertex 2

Dim disp_V2_dr1, disp_V2_dr2, disp_V2_dr3 As Double 'mpc displacements for vertex mpc_link_2

Print #1, "$vertices FBoR-BaTL"

FBoR = 19552

BaTL = 6479
```

```
disp_V2_dr1 = mpc_pair.Cells(27, 13).Value
disp_V2_dr2 = mpc_pair.Cells(28, 13).Value
disp_V2_dr3 = mpc_pair.Cells(29, 13).Value
'compare location components
comp1 = WorksheetFunction.Match(19552, nodes_range, 1)
comp2 = WorksheetFunction.Match(6479, nodes_range, 1)
xcomp1 = WorksheetFunction.Index(res_rangex, comp1)
xcomp2 = WorksheetFunction.Index(res_rangex, comp2)
ycomp1 = WorksheetFunction.Index(res_rangey, comp1)
ycomp2 = WorksheetFunction.Index(res_rangey, comp2)
zcomp1 = WorksheetFunction.Index(res_rangez, comp1)
zcomp2 = WorksheetFunction.Index(res_rangez, comp2)
If xcomp1 > xcomp2 Then
Print #1, "MPCY,4," + CStr(FBoR) + ",1,1.0," + CStr(disp_V2_dr1)
Print #1, ",," + CStr(BaTL) + ",1,-1.0"
Else
Print #1, "MPCY,4," + CStr(BaTL) + ",1,1.0," + CStr(disp_V2_dr1)
Print #1, ",," + CStr(FBoR) + ",1,-1.0"
End If
If ycomp1 > ycomp2 Then
Print #1, "MPCY,5," + CStr(FBoR) + ",2,1.0," + CStr(disp_V2_dr2)
Print #1, ",," + CStr(BaTL) + ",2,-1.0"
Else
Print #1, "MPCY,5," + CStr(BaTL) + ",2,1.0," + CStr(disp_V2_dr2)
Print #1, ",," + CStr(FBoR) + ",2,-1.0"
End I
If zcomp1 > zcomp2 Then
Print #1, "MPCY,6," + CStr(FBoR) + ",3,1.0," + CStr(disp_V2_dr3)
```

```
Print #1, ",," + CStr(BaTL) + ",3,-1.0"

Else

Print #1, "MPCY,6," + CStr(BaTL) + ",3,1.0," + CStr(disp_V2_dr3)

Print #1, ",," + CStr(FBoR) + ",3,-1.0"

End If

   'print info for vertex 3

Dim disp_V3_dr1, disp_V3_dr2, disp_V3_dr3 As Double 'mpc displacements for vertex mpc_link_3

Print #1, "$vertices FBoL-BaTR"

FBoL = 18688

BaTR = 59673

disp_V3_dr1 = mpc_pair.Cells(15, 13).Value

disp_V3_dr2 = mpc_pair.Cells(16, 13).Value

disp_V3_dr3 = mpc_pair.Cells(17, 13).Value

'compare location components

comp1 = WorksheetFunction.Match(18688, nodes_range, 1)

comp2 = WorksheetFunction.Match(59673, nodes_range, 1)

xcomp1 = WorksheetFunction.Index(res_rangex, comp1)

xcomp2 = WorksheetFunction.Index(res_rangex, comp2)

ycomp1 = WorksheetFunction.Index(res_rangey, comp1)

ycomp2 = WorksheetFunction.Index(res_rangey, comp2)

zcomp1 = WorksheetFunction.Index(res_rangez, comp1)

zcomp2 = WorksheetFunction.Index(res_rangez, comp2)

If xcomp1 > xcomp2 Then

   Print #1, "MPCY,7," + CStr(FBoL) + ",1,1.0," + CStr(disp_V3_dr1)

   Print #1, ",," + CStr(BaTR) + ",1,-1.0"

Else

   Print #1, "MPCY,7," + CStr(BaTR) + ",1,1.0," + CStr(disp_V3_dr1)

   Print #1, ",," + CStr(FBoL) + ",1,-1.0"
```

```
End If
If ycomp1 > ycomp2 Then
   Print #1, "MPCY,8," + CStr(FBoL) + ",2,1.0," + CStr(disp_V3_dr2)
   Print #1, ",," + CStr(BaTR) + ",2,-1.0"
Else
   Print #1, "MPCY,8," + CStr(BaTR) + ",2,1.0," + CStr(disp_V3_dr2)
   Print #1, ",," + CStr(FBoL) + ",2,-1.0"
End If
If zcomp1 > zcomp2 Then
   Print #1, "MPCY,9," + CStr(FBoL) + ",3,1.0," + CStr(disp_V3_dr3)
   Print #1, ",," + CStr(BaTR) + ",3,-1.0"
Else
   Print #1, "MPCY,9," + CStr(BaTR) + ",3,1.0," + CStr(disp_V3_dr3)
   Print #1, ",," + CStr(FBoL) + ",3,-1.0"
End If
   'print info for vertex link 4
Dim disp_V4_dr1, disp_V4_dr2, disp_V4_dr3 As Double 'mpc displacements for vertex mpc_link_4
Print #1, "$vertices FTL-BaBoR"
FTL = 60537
BaBoR = 17824
disp_V4_dr1 = mpc_pair.Cells(21, 13).Value
disp_V4_dr2 = mpc_pair.Cells(22, 13).Value
disp_V4_dr3 = mpc_pair.Cells(23, 13).Value
'compare location components
comp1 = WorksheetFunction.Match(60537, nodes_range, 1)
comp2 = WorksheetFunction.Match(17824, nodes_range, 1)
xcomp1 = WorksheetFunction.Index(res_rangex, comp1)
xcomp2 = WorksheetFunction.Index(res_rangex, comp2)
```

```
ycomp1 = WorksheetFunction.Index(res_rangey, comp1)
ycomp2 = WorksheetFunction.Index(res_rangey, comp2)
zcomp1 = WorksheetFunction.Index(res_rangez, comp1)
zcomp2 = WorksheetFunction.Index(res_rangez, comp2)
If xcomp1 > xcomp2 Then
   Print #1, "MPCY,10," + CStr(FTL) + ",1,1.0," + CStr(disp_V4_dr1)
   Print #1, ",," + CStr(BaBoR) + ",1,-1.0"
Else
   Print #1, "MPCY,10," + CStr(BaBoR) + ",1,1.0," + CStr(disp_V4_dr1)
   Print #1, ",," + CStr(FTL) + ",1,-1.0"
End If
If ycomp1 > ycomp2 Then
   Print #1, "MPCY,11," + CStr(FTL) + ",2,1.0," + CStr(disp_V4_dr2)
   Print #1, ",," + CStr(BaBoR) + ",2,-1.0"
Else
   Print #1, "MPCY,11," + CStr(BaBoR) + ",2,1.0," + CStr(disp_V4_dr2)
   Print #1, ",," + CStr(FTL) + ",2,-1.0"
End If
If zcomp1 > zcomp2 Then
   Print #1, "MPCY,12," + CStr(FTL) + ",3,1.0," + CStr(disp_V4_dr3)
   Print #1, ",," + CStr(BaBoR) + ",3,-1.0"
Else
   Print #1, "MPCY,12," + CStr(BaBoR) + ",3,1.0," + CStr(disp_V4_dr3)
   Print #1, ",," + CStr(FTL) + ",3,-1.0"
End If
' print mpcs for edges
Print #1, "$ Edges FT_BaBo"
Dim numpts As Integer
```

```
Dim E_1_pts(1 To 49, 1 To 2) As Double
Dim disp_E1_dr1, disp_E1_dr2, disp_E1_dr3, stp As Double ' Disp for edge 1
' Initialize
numpts = 49
For i = 1 To numpts
    E_1_pts(i, 1) = pairsht.Cells(4 + i, 15).Value
    E_1_pts(i, 2) = pairsht.Cells(4 + i, 16).Value
Next i
disp_E1_dr1 = mpc_pair.Cells(35, 13).Value
disp_E1_dr2 = mpc_pair.Cells(36, 13).Value
disp_E1_dr3 = mpc_pair.Cells(37, 13).Value
stp = 0
'compare location components
comp1 = WorksheetFunction.Match(43291, nodes_range, 1)
comp2 = WorksheetFunction.Match(6258, nodes_range, 1)
xcomp1 = WorksheetFunction.Index(res_rangex, comp1)
xcomp2 = WorksheetFunction.Index(res_rangex, comp2)
ycomp1 = WorksheetFunction.Index(res_rangey, comp1)
ycomp2 = WorksheetFunction.Index(res_rangey, comp2)
zcomp1 = WorksheetFunction.Index(res_rangez, comp1)
zcomp2 = WorksheetFunction.Index(res_rangez, comp2)
For i = 1 To numpts
    If xcomp1 >= xcomp2 Then
    ' print x dir mpc
        Print #1, "MPCY," + CStr(12 + i + stp) + "," + CStr(E_1_pts(i, 1)) + ",1,1.0," _
              + CStr(disp_E1_dr1)
        Print #1, ",," + CStr(E_1_pts(i, 2)) + ",1,-1.0"
    Else
```

```
' print x dir mpc
    Print #1, "MPCY," + CStr(12 + i + stp) + "," + CStr(E_1_pts(i, 2)) + ",1,1.0," _
            + CStr(disp_E1_dr1)
    Print #1, ",," + CStr(E_1_pts(i, 1)) + ",1,-1.0"
End If
If ycomp1 >= ycomp2 Then
' print y dir mpc
    Print #1, "MPCY," + CStr(12 + i + 1 + stp) + "," + CStr(E_1_pts(i, 1)) + ",2,1.0," _
            + CStr(disp_E1_dr2)
    Print #1, ",," + CStr(E_1_pts(i, 2)) + ",2,-1.0"
Else
' print y dir mpc
    Print #1, "MPCY," + CStr(12 + i + 1 + stp) + "," + CStr(E_1_pts(i, 2)) + ",2,1.0," _
            + CStr(disp_E1_dr2)
    Print #1, ",," + CStr(E_1_pts(i, 1)) + ",2,-1.0"
End If
If zcomp1 >= zcomp2 Then
' print z dir mpc
    Print #1, "MPCY," + CStr(12 + i + 2 + stp) + "," + CStr(E_1_pts(i, 1)) + ",3,1.0," _
            + CStr(disp_E1_dr3)
    Print #1, ",," + CStr(E_1_pts(i, 2)) + ",3,-1.0"
Else
' print z dir mpc
    Print #1, "MPCY," + CStr(12 + i + 2 + stp) + "," + CStr(E_1_pts(i, 2)) + ",3,1.0," _
        + CStr(disp_E1_dr3)
    Print #1, ",," + CStr(E_1_pts(i, 1)) + ",3,-1.0"
End If
stp = stp + 2
```

```
Next i
Dim nxt_mpc As Integer
nxt_mpc = 12 + i + stp - 1
' print mpcs for edges
Print #1, "$ Edges FBo_TBa"
Dim E_2_pts(1 To 49, 1 To 2) As Double
Dim disp_E2_dr1, disp_E2_dr2, disp_E2_dr3 As Double ' Disp for edge 2
' Initialize
numpts = 49
For i = 1 To numpts
    E_2_pts(i, 1) = pairsht.Cells(4 + i, 19).Value
    E_2_pts(i, 2) = pairsht.Cells(4 + i, 20).Value
Next i
disp_E2_dr1 = mpc_pair.Cells(40, 13).Value
disp_E2_dr2 = mpc_pair.Cells(41, 13).Value
disp_E2_dr3 = mpc_pair.Cells(42, 13).Value
stp = 0
'compare location components
comp1 = WorksheetFunction.Match(18664, nodes_range, 1)
comp2 = WorksheetFunction.Match(6455, nodes_range, 1)
xcomp1 = WorksheetFunction.Index(res_rangex, comp1)
xcomp2 = WorksheetFunction.Index(res_rangex, comp2)
ycomp1 = WorksheetFunction.Index(res_rangey, comp1)
ycomp2 = WorksheetFunction.Index(res_rangey, comp2)
zcomp1 = WorksheetFunction.Index(res_rangez, comp1)
zcomp2 = WorksheetFunction.Index(res_rangez, comp2)
For i = 1 To numpts
    If xcomp1 > xcomp2 Then
```

```
' print x dir mpc
    Print #1, "MPCY," + CStr(nxt_mpc + i + stp) + "," + CStr(E_2_pts(i, 1)) + ",1,1.0," _
            + CStr(disp_E2_dr1)
    Print #1, ",," + CStr(E_2_pts(i, 2)) + ",1,-1.0"
Else
' print x dir mpc
    Print #1, "MPCY," + CStr(nxt_mpc + i + stp) + "," + CStr(E_2_pts(i, 2)) + ",1,1.0," _
            + CStr(disp_E2_dr1)
    Print #1, ",," + CStr(E_2_pts(i, 1)) + ",1,-1.0"
End If
If ycomp1 > ycomp2 Then
' print y dir mpc
    Print #1, "MPCY, " + CStr(nxt_mpc + i + 1 + stp) + "," + CStr(E_2_pts(i, 1)) + ",2,1.0," _
            + CStr(disp_E2_dr2)
    Print #1, ",," + CStr(E_2_pts(i, 2)) + ",2,-1.0"
Else
    ' print y dir mpc
    Print #1, "MPCY," + CStr(nxt_mpc + i + 1 + stp) + "," + CStr(E_2_pts(i, 2)) + ",2,1.0," _
            + CStr(disp_E2_dr2)
    Print #1, ",," + CStr(E_2_pts(i, 1)) + ",2,-1.0"
End If
If zcomp1 > zcomp2 Then
' print z dir mpc
    Print #1, "MPCY," + CStr(nxt_mpc + i + 2 + stp) + "," + CStr(E_2_pts(i, 1)) + ",3,1.0," _
            + CStr(disp_E2_dr3)
    Print #1, ",," + CStr(E_2_pts(i, 2)) + ",3,-1.0"
Else
 ' print z dir mpc
```

```
        Print #1, "MPCY," + CStr(nxt_mpc + i + 2 + stp) + "," + CStr(E_2_pts(i, 2)) + ",3,1.0," _
            + CStr(disp_E2_dr3)
        Print #1, ",," + CStr(E_2_pts(i, 1)) + ",3,-1.0"
    End If
    stp = stp + 2
Next i
nxt_mpc = nxt_mpc + i + stp - 1
' print mpcs for edges
Print #1, "$ Edges FR_BaL"
Dim E_3_pts(1 To 19, 1 To 2) As Double
Dim disp_E3_dr1, disp_E3_dr2, disp_E3_dr3 As Double ' Disp for edge 3
' Initialize
numpts = 19
For i = 1 To numpts
    E_3_pts(i, 1) = pairsht.Cells(4 + i, 23).Value
    E_3_pts(i, 2) = pairsht.Cells(4 + i, 24).Value
Next i
disp_E3_dr1 = mpc_pair.Cells(45, 13).Value
disp_E3_dr2 = mpc_pair.Cells(46, 13).Value
disp_E3_dr3 = mpc_pair.Cells(47, 13).Value
stp = 0
'compare location components
comp1 = WorksheetFunction.Match(19412, nodes_range, 1)
comp2 = WorksheetFunction.Match(5466, nodes_range, 1)
xcomp1 = WorksheetFunction.Index(res_rangex, comp1)
xcomp2 = WorksheetFunction.Index(res_rangex, comp2)
ycomp1 = WorksheetFunction.Index(res_rangey, comp1)
ycomp2 = WorksheetFunction.Index(res_rangey, comp2)
```

```
zcomp1 = WorksheetFunction.Index(res_rangez, comp1)
zcomp2 = WorksheetFunction.Index(res_rangez, comp2)

For i = 1 To numpts
   If xcomp1 >= xcomp2 Then
   ' print x dir mpc
      Print #1, "MPCY," + CStr(nxt_mpc + i + stp) + "," + CStr(E_3_pts(i, 1)) + ",1,1.0," _
            + CStr(disp_E3_dr1)
      Print #1, ",," + CStr(E_3_pts(i, 2)) + ",1,-1.0"
   Else
   ' print x dir mpc
      Print #1, "MPCY," + CStr(nxt_mpc + i + stp) + "," + CStr(E_3_pts(i, 2)) + ",1,1.0," _
            + CStr(disp_E3_dr1)
      Print #1, ",," + CStr(E_3_pts(i, 1)) + ",1,-1.0"
   End If
   If ycomp1 >= ycomp2 Then
   ' print y dir mpc
      Print #1, "MPCY," + CStr(nxt_mpc + i + 1 + stp) + "," + CStr(E_3_pts(i, 1)) + ",2,1.0," _
            + CStr(disp_E3_dr2)
      Print #1, ",," + CStr(E_3_pts(i, 2)) + ",2,-1.0"
   Else
    ' print y dir mpc
      Print #1, "MPCY," + CStr(nxt_mpc + i + 1 + stp) + "," + CStr(E_3_pts(i, 2)) + ",2,1.0," _
            + CStr(disp_E3_dr2)
      Print #1, ",," + CStr(E_3_pts(i, 1)) + ",2,-1.0"
   End If
   If zcomp1 >= zcomp2 Then
   ' print z dir mpc
```

```
        Print #1, "MPCY," + CStr(nxt_mpc + i + 2 + stp) + "," + CStr(E_3_pts(i, 1)) + ",3,1.0," _
            + CStr(disp_E3_dr3)
        Print #1, ",," + CStr(E_3_pts(i, 2)) + ",3,-1.0"
    Else
    ' print z dir mpc
        Print #1, "MPCY," + CStr(nxt_mpc + i + 2 + stp) + "," + CStr(E_3_pts(i, 2)) + ",3,1.0," _
            + CStr(disp_E3_dr3)
        Print #1, ",," + CStr(E_3_pts(i, 1)) + ",3,-1.0"
    End If
    stp = stp + 2
Next i
nxt_mpc = nxt_mpc + i + stp - 1
' print mpcs for edges
Print #1, "$ Edges FL_RBa"
Dim E_4_pts(1 To 19, 1 To 2) As Double
Dim disp_E4_dr1, disp_E4_dr2, disp_E4_dr3 As Double ' Disp for edge 4
' Initialize
numpts = 19
For i = 1 To numpts
    E_4_pts(i, 1) = pairsht.Cells(4 + i, 27).Value
    E_4_pts(i, 2) = pairsht.Cells(4 + i, 28).Value
Next i
disp_E4_dr1 = mpc_pair.Cells(50, 13).Value
disp_E4_dr2 = mpc_pair.Cells(51, 13).Value
disp_E4_dr3 = mpc_pair.Cells(52, 13).Value
stp = 0
'compare location components
comp1 = WorksheetFunction.Match(18548, nodes_range, 1)
```

```
comp2 = WorksheetFunction.Match(17684, nodes_range, 1)
xcomp1 = WorksheetFunction.Index(res_rangex, comp1)
xcomp2 = WorksheetFunction.Index(res_rangex, comp2)
ycomp1 = WorksheetFunction.Index(res_rangey, comp1)
ycomp2 = WorksheetFunction.Index(res_rangey, comp2)
zcomp1 = WorksheetFunction.Index(res_rangez, comp1)
zcomp2 = WorksheetFunction.Index(res_rangez, comp2)
For i = 1 To numpts
   If xcomp1 > xcomp2 Then
   ' print x dir mpc
      Print #1, "MPCY," + CStr(nxt_mpc + i + stp) + "," + CStr(E_4_pts(i, 1)) + ",1,1.0," _
            + CStr(disp_E4_dr1)
      Print #1, ",," + CStr(E_4_pts(i, 2)) + ",1,-1.0"
   Else
      ' print x dir mpc
      Print #1, "MPCY," + CStr(nxt_mpc + i + stp) + "," + CStr(E_4_pts(i, 2)) + ",1,1.0," _
            + CStr(disp_E4_dr1)
      Print #1, ",," + CStr(E_4_pts(i, 1)) + ",1,-1.0"
   End If
   If ycomp1 > ycomp2 Then
   ' print y dir mpc
      Print #1, "MPCY," + CStr(nxt_mpc + i + 1 + stp) + "," + CStr(E_4_pts(i, 1)) + ",2,1.0," _
            + CStr(disp_E4_dr2)
      Print #1, ",," + CStr(E_4_pts(i, 2)) + ",2,-1.0"
   Else
   ' print y dir mpc
      Print #1, "MPCY," + CStr(nxt_mpc + i + 1 + stp) + "," + CStr(E_4_pts(i, 2)) + ",2,1.0," _
            + CStr(disp_E4_dr2)
```

```
        Print #1, ",," + CStr(E_4_pts(i, 1)) + ",2,-1.0"
    End If
    If zcomp1 > zcomp2 Then
    ' print z dir mpc
        Print #1, "MPCY," + CStr(nxt_mpc + i + 2 + stp) + "," + CStr(E_4_pts(i, 1)) + ",3,1.0," _
            + CStr(disp_E4_dr3)
        Print #1, ",," + CStr(E_4_pts(i, 2)) + ",3,-1.0"
    Else
    ' print z dir mpc
        Print #1, "MPCY," + CStr(nxt_mpc + i + 2 + stp) + "," + CStr(E_4_pts(i, 2)) + ",3,1.0," _
            + CStr(disp_E4_dr3)
        Print #1, ",," + CStr(E_4_pts(i, 1)) + ",3,-1.0"
    End If
    stp = stp + 2
Next i
nxt_mpc = nxt_mpc + i + stp - 1
' print mpcs for edges
Print #1, "$ Edges RT_LBo"
Dim E_5_pts(1 To 49, 1 To 2) As Double
Dim disp_E5_dr1, disp_E5_dr2, disp_E5_dr3 As Double ' Disp for edge 5
' Initialize
numpts = 49
For i = 1 To numpts
    E_5_pts(i, 1) = pairsht.Cells(4 + i, 31).Value
    E_5_pts(i, 2) = pairsht.Cells(4 + i, 32).Value
Next i
disp_E5_dr1 = mpc_pair.Cells(55, 13).Value
disp_E5_dr2 = mpc_pair.Cells(56, 13).Value
```

```
disp_E5_dr3 = mpc_pair.Cells(57, 13).Value
stp = 0
'numpts = 1
'compare location components
comp1 = WorksheetFunction.Match(37890, nodes_range, 1)
comp2 = WorksheetFunction.Match(5801, nodes_range, 1)
xcomp1 = WorksheetFunction.Index(res_rangex, comp1)
xcomp2 = WorksheetFunction.Index(res_rangex, comp2)
ycomp1 = WorksheetFunction.Index(res_rangey, comp1)
ycomp2 = WorksheetFunction.Index(res_rangey, comp2)
zcomp1 = WorksheetFunction.Index(res_rangez, comp1)
zcomp2 = WorksheetFunction.Index(res_rangez, comp2)
For i = 1 To numpts
   If xcomp1 >= xcomp2 Then
   ' print x dir mpc
      Print #1, "MPCY," + CStr(nxt_mpc + i + stp) + "," + CStr(E_5_pts(i, 1)) + ",1,1.0," _
            + CStr(disp_E5_dr1)
      Print #1, ",," + CStr(E_5_pts(i, 2)) + ",1,-1.0"
   Else
   ' print x dir mpc
      Print #1, "MPCY," + CStr(nxt_mpc + i + stp) + "," + CStr(E_5_pts(i, 2)) + ",1,1.0," _
            + CStr(disp_E5_dr1)
      Print #1, ",," + CStr(E_5_pts(i, 1)) + ",1,-1.0"
   End If
   If ycomp1 >= ycomp2 Then
   ' print y dir mpc
      Print #1, "MPCY," + CStr(nxt_mpc + i + 1 + stp) + "," + CStr(E_5_pts(i, 1)) + ",2,1.0," _
            + CStr(disp_E5_dr2)
```

```
        Print #1, ",," + CStr(E_5_pts(i, 2)) + ",2,-1.0"
    Else
    ' print y dir mpc
        Print #1, "MPCY," + CStr(nxt_mpc + i + 1 + stp) + "," + CStr(E_5_pts(i, 2)) + ",2,1.0," _
            + CStr(disp_E5_dr2)
        Print #1, ",," + CStr(E_5_pts(i, 1)) + ",2,-1.0"
    End If
    If zcomp1 >= zcomp2 Then
    ' print z dir mpc
        Print #1, "MPCY," + CStr(nxt_mpc + i + 2 + stp) + "," + CStr(E_5_pts(i, 1)) + ",3,1.0," _
            + CStr(disp_E5_dr3)
        Print #1, ",," + CStr(E_5_pts(i, 2)) + ",3,-1.0"
    Else
    ' print z dir mpc
        Print #1, "MPCY," + CStr(nxt_mpc + i + 2 + stp) + "," + CStr(E_5_pts(i, 2)) + ",3,1.0," _
            + CStr(disp_E5_dr3)
        Print #1, ",," + CStr(E_5_pts(i, 1)) + ",3,-1.0"
    End If
    stp = stp + 2
Next i
nxt_mpc = nxt_mpc + i + stp - 1
' print mpcs for edges
Print #1, "$ Edges TL_RBo"
Dim E_6_pts(1 To 49, 1 To 2) As Double
Dim disp_E6_dr1, disp_E6_dr2, disp_E6_dr3 As Double ' Disp for edge 6
' Initialize
numpts = 49
For i = 1 To numpts
```

```
    E_6_pts(i, 1) = pairsht.Cells(4 + i, 35).Value
    E_6_pts(i, 2) = pairsht.Cells(4 + i, 36).Value
Next i
disp_E6_dr1 = mpc_pair.Cells(60, 13).Value
disp_E6_dr2 = mpc_pair.Cells(61, 13).Value
disp_E6_dr3 = mpc_pair.Cells(62, 13).Value
stp = 0
'numpts = 1
'compare location components
comp1 = WorksheetFunction.Match(6475, nodes_range, 1)
comp2 = WorksheetFunction.Match(17828, nodes_range, 1)
xcomp1 = WorksheetFunction.Index(res_rangex, comp1)
xcomp2 = WorksheetFunction.Index(res_rangex, comp2)
ycomp1 = WorksheetFunction.Index(res_rangey, comp1)
ycomp2 = WorksheetFunction.Index(res_rangey, comp2)
zcomp1 = WorksheetFunction.Index(res_rangez, comp1)
zcomp2 = WorksheetFunction.Index(res_rangez, comp2)
For i = 1 To numpts
    If xcomp1 > xcomp2 Then
    ' print x dir mpc
        Print #1, "MPCY," + CStr(nxt_mpc + i + stp) + "," + CStr(E_6_pts(i, 1)) + ",1,1.0," _
              + CStr(disp_E6_dr1)
        Print #1, ",," + CStr(E_6_pts(i, 2)) + ",1,-1.0"
    Else
    ' print x dir mpc
        Print #1, "MPCY," + CStr(nxt_mpc + i + stp) + "," + CStr(E_6_pts(i, 2)) + ",1,1.0," _
              + CStr(disp_E6_dr1)
        Print #1, ",," + CStr(E_6_pts(i, 1)) + ",1,-1.0"
```

```
    End If
    If ycomp1 > ycomp2 Then
    ' print y dir mpc
        Print #1, "MPCY," + CStr(nxt_mpc + i + 1 + stp) + "," + CStr(E_6_pts(i, 1)) + ",2,1.0," _
            + CStr(disp_E6_dr2)
        Print #1, ",," + CStr(E_6_pts(i, 2)) + ",2,-1.0"
    Else
    ' print y dir mpc
        Print #1, "MPCY," + CStr(nxt_mpc + i + 1 + stp) + "," + CStr(E_6_pts(i, 2)) + ",2,1.0," _
            + CStr(disp_E6_dr2)
        Print #1, ",," + CStr(E_6_pts(i, 1)) + ",2,-1.0"
    End If
    If zcomp1 > zcomp2 Then
    ' print z dir mpc
        Print #1, "MPCY," + CStr(nxt_mpc + i + 2 + stp) + "," + CStr(E_6_pts(i, 1)) + ",3,1.0," _
            + CStr(disp_E6_dr3)
        Print #1, ",," + CStr(E_6_pts(i, 2)) + ",3,-1.0"
    Else
    ' print z dir mpc
        Print #1, "MPCY," + CStr(nxt_mpc + i + 2 + stp) + "," + CStr(E_6_pts(i, 2)) + ",3,1.0," _
            + CStr(disp_E6_dr3)
        Print #1, ",," + CStr(E_6_pts(i, 1)) + ",3,-1.0"
    End If
    stp = stp + 2
Next i
nxt_mpc = nxt_mpc + i + stp - 1
' print mpcs for surfaces
Print #1, "$surfaces T_B"
```

```
Dim S_1_pts(1 To 2401, 1 To 2) As Double
Dim disp_S1_dr1, disp_S1_dr2, disp_S1_dr3 As Double ' Disp for surface T_B
' Initialize
numpts = 2401
For i = 1 To numpts
    S_1_pts(i, 1) = pairsht.Cells(4 + i, 11).Value
    S_1_pts(i, 2) = pairsht.Cells(4 + i, 12).Value
Next i
disp_S1_dr1 = mpc_pair.Cells(68, 13).Value
disp_S1_dr2 = mpc_pair.Cells(69, 13).Value
disp_S1_dr3 = mpc_pair.Cells(70, 13).Value
stp = 0
'numpts = 1
For i = 1 To numpts
    'compare location components
    comp1 = WorksheetFunction.Match(S_1_pts(i, 1), nodes_range, 1)
    comp2 = WorksheetFunction.Match(S_1_pts(i, 2), nodes_range, 1)
    xcomp1 = WorksheetFunction.Index(res_rangex, comp1)
    xcomp2 = WorksheetFunction.Index(res_rangex, comp2)
    ycomp1 = WorksheetFunction.Index(res_rangey, comp1)
    ycomp2 = WorksheetFunction.Index(res_rangey, comp2)
    zcomp1 = WorksheetFunction.Index(res_rangez, comp1)
    zcomp2 = WorksheetFunction.Index(res_rangez, comp2)

    If xcomp1 > xcomp2 Then
    ' print x dir mpc
        Print #1, "MPCY," + CStr(nxt_mpc + i + stp) + "," + CStr(S_1_pts(i, 1)) + ",1,1.0," _
            + CStr(disp_S1_dr1)
```

```
    Print #1, ",," + CStr(S_1_pts(i, 2)) + ",1,-1.0"
Else
' print x dir mpc
    Print #1, "MPCY," + CStr(nxt_mpc + i + stp) + "," + CStr(S_1_pts(i, 2)) + ",1,1.0," _
        + CStr(disp_S1_dr1)
    Print #1, ",," + CStr(S_1_pts(i, 1)) + ",1,-1.0"
End If
If ycomp1 > ycomp2 Then
' print y dir mpc
    Print #1, "MPCY," + CStr(nxt_mpc + i + 1 + stp) + "," + CStr(S_1_pts(i, 1)) + ",2,1.0," _
        + CStr(disp_S1_dr2)
    Print #1, ",," + CStr(S_1_pts(i, 2)) + ",2,-1.0"
Else
' print y dir mpc
    Print #1, "MPCY," + CStr(nxt_mpc + i + 1 + stp) + "," + CStr(S_1_pts(i, 2)) + ",2,1.0," _
        + CStr(disp_S1_dr2)
    Print #1, ",," + CStr(S_1_pts(i, 1)) + ",2,-1.0"
End If
If zcomp1 > zcomp2 Then
' print z dir mpc
    Print #1, "MPCY," + CStr(nxt_mpc + i + 2 + stp) + "," + CStr(S_1_pts(i, 1)) + ",3,1.0," _
        + CStr(disp_S1_dr3)
    Print #1, ",," + CStr(S_1_pts(i, 2)) + ",3,-1.0"
Else
' print z dir mpc
    Print #1, "MPCY," + CStr(nxt_mpc + i + 2 + stp) + "," + CStr(S_1_pts(i, 2)) + ",3,1.0," _
        + CStr(disp_S1_dr3)
    Print #1, ",," + CStr(S_1_pts(i, 1)) + ",3,-1.0"
```

```
    End If
    stp = stp + 2
Next i
nxt_mpc = nxt_mpc + i + stp - 1
' print mpcs for surfaces
Print #1, "$Surfaces L_R"
Dim S_2_pts(1 To 831, 1 To 2) As Double
Dim disp_S2_dr1, disp_S2_dr2, disp_S2_dr3 As Double ' Disp for surface L_R
' Initialize
numpts = 831
For i = 1 To numpts
    S_2_pts(i, 1) = pairsht.Cells(4 + i, 7).Value

    S_2_pts(i, 2) = pairsht.Cells(4 + i, 8).Value
Next i
disp_S2_dr1 = mpc_pair.Cells(73, 13).Value
disp_S2_dr2 = mpc_pair.Cells(74, 13).Value
disp_S2_dr3 = mpc_pair.Cells(75, 13).Value
stp = 0
'numpts = 1
For i = 1 To numpts
    'compare location components
    comp1 = WorksheetFunction.Match(S_2_pts(3, 1), nodes_range, 1)
    comp2 = WorksheetFunction.Match(S_2_pts(3, 2), nodes_range, 1)
    xcomp1 = WorksheetFunction.Index(res_rangex, comp1)
    xcomp2 = WorksheetFunction.Index(res_rangex, comp2)
    ycomp1 = WorksheetFunction.Index(res_rangey, comp1)
    ycomp2 = WorksheetFunction.Index(res_rangey, comp2)
```

```
zcomp1 = WorksheetFunction.Index(res_rangez, comp1)
zcomp2 = WorksheetFunction.Index(res_rangez, comp2)
If xcomp1 > xcomp2 Then
' print x dir mpc
   Print #1, "MPCY," + CStr(nxt_mpc + i + stp) + "," + CStr(S_2_pts(i, 1)) + ",1,1.0," _
         + CStr(disp_S2_dr1)
   Print #1, ",," + CStr(S_2_pts(i, 2)) + ",1,-1.0"
Else
' print x dir mpc
   Print #1, "MPCY," + CStr(nxt_mpc + i + stp) + "," + CStr(S_2_pts(i, 2)) + ",1,1.0," _
         + CStr(disp_S2_dr1)
   Print #1, ",," + CStr(S_2_pts(i, 1)) + ",1,-1.0"
End If
If ycomp1 > ycomp2 Then
' print y dir mpc
   Print #1, "MPCY," + CStr(nxt_mpc + i + 1 + stp) + "," + CStr(S_2_pts(i, 1)) + ",2,1.0," _
         + CStr(disp_S2_dr2)
   Print #1, ",," + CStr(S_2_pts(i, 2)) + ",2,-1.0"
Else
' print y dir mpc
   Print #1, "MPCY," + CStr(nxt_mpc + i + 1 + stp) + "," + CStr(S_2_pts(i, 2)) + ",2,1.0," _
         + CStr(disp_S2_dr2)
   Print #1, ",," + CStr(S_2_pts(i, 1)) + ",2,-1.0"
End If
If zcomp1 > zcomp2 Then
' print z dir mpc
   Print #1, "MPCY," + CStr(nxt_mpc + i + 2 + stp) + "," + CStr(S_2_pts(i, 1)) + ",3,1.0," _
         + CStr(disp_S2_dr3)
```

```
        Print #1, ",," + CStr(S_2_pts(i, 2)) + ",3,-1.0"
    Else
    ' print z dir mpc
        Print #1, "MPCY," + CStr(nxt_mpc + i + 2 + stp) + "," + CStr(S_2_pts(i, 2)) + ",3,1.0," _
            + CStr(disp_S2_dr3)
        Print #1, ",," + CStr(S_2_pts(i, 1)) + ",3,-1.0"
    End If
    stp = stp + 2
Next i
nxt_mpc = nxt_mpc + i + stp - 1
' print mpcs for surfaces
Print #1, "$Surfaces F_B"
Dim S_3_pts(1 To 831, 1 To 2) As Double
Dim disp_S3_dr1, disp_S3_dr2, disp_S3_dr3 As Double ' Disp for surface F_B
' Initialize
numpts = 831
For i = 1 To numpts
    S_3_pts(i, 1) = pairsht.Cells(4 + i, 3).Value
    S_3_pts(i, 2) = pairsht.Cells(4 + i, 4).Value
Next i
disp_S3_dr1 = mpc_pair.Cells(78, 13).Value
disp_S3_dr2 = mpc_pair.Cells(79, 13).Value
disp_S3_dr3 = mpc_pair.Cells(80, 13).Value
stp = 0
'compare location components
comp1 = WorksheetFunction.Match(S_3_pts(3, 1), nodes_range, 1)
comp2 = WorksheetFunction.Match(S_3_pts(3, 2), nodes_range, 1)
xcomp1 = WorksheetFunction.Index(res_rangex, comp1)
```

```
xcomp2 = WorksheetFunction.Index(res_rangex, comp2)
ycomp1 = WorksheetFunction.Index(res_rangey, comp1)
ycomp2 = WorksheetFunction.Index(res_rangey, comp2)
zcomp1 = WorksheetFunction.Index(res_rangez, comp1)
zcomp2 = WorksheetFunction.Index(res_rangez, comp2)
For i = 1 To numpts
   If xcomp1 > xcomp2 Then
   ' print x dir mpc
      Print #1, "MPCY," + CStr(nxt_mpc + i + stp) + "," + CStr(S_3_pts(i, 1)) + ",1,--1.0," _
            + CStr(disp_S3_dr1)
      Print #1, ",," + CStr(S_3_pts(i, 2)) + ",1,-1.0"
   Else
   ' print x dir mpc
      Print #1, "MPCY," + CStr(nxt_mpc + i + stp) + "," + CStr(S_3_pts(i, 2)) + ",1,1.0," _
            + CStr(disp_S3_dr1)
      Print #1, ",," + CStr(S_3_pts(i, 1)) + ",1,-1.0"
   End If
   If ycomp1 > ycomp2 Then
   ' print y dir mpc
      Print #1, "MPCY," + CStr(nxt_mpc + i + 1 + stp) + "," + CStr(S_3_pts(i, 1)) + ",2,1.0," _
            + CStr(disp_S3_dr2)
      Print #1, ",," + CStr(S_3_pts(i, 2)) + ",2,-1.0"
   Else
      ' print y dir mpc
      Print #1, "MPCY," + CStr(nxt_mpc + i + 1 + stp) + "," + CStr(S_3_pts(i, 2)) + ",2,1.0," _
            + CStr(disp_S3_dr2)
      Print #1, ",," + CStr(S_3_pts(i, 1)) + ",2,-1.0"
   End If
```

```
    If zcomp1 > zcomp2 Then
    ' print z dir mpc
       Print #1, "MPCY," + CStr(nxt_mpc + i + 2 + stp) + "," + CStr(S_3_pts(i, 1)) + ",3,1.0," _
            + CStr(disp_S3_dr3)
       Print #1, ",," + CStr(S_3_pts(i, 2)) + ",3,-1.0"
    Else
      ' print z dir mpc
       Print #1, "MPCY," + CStr(nxt_mpc + i + 2 + stp) + "," + CStr(S_3_pts(i, 2)) + ",3,1.0," _
            + CStr(disp_S3_dr3)
       Print #1, ",," + CStr(S_3_pts(i, 1)) + ",3,-1.0"
    End If
    stp = stp + 2
Next i
'Close #1
End Sub
```

7. VBA Script meso_bdf_file ()

```
Option Explicit

Dim loop_incr, meso_lim_loop, micro_lim_loop As Integer

Sub meso_bdf_file(loop_incr)

Dim j As Integer

Dim const_law_sht, MPC_EQ_DISP_sht, micro_mat_sht As Worksheet

Dim mat9_wefttow, mat9_warptow, mat9_weftramp, mat9_warpramp, id_mat As Variant

Dim mat9_wefttow_damg, mat9_warptow_damg, mat9_weftramp_damg, mat9_warpramp_damg As Variant

Set const_law_sht = Worksheets("MESO_CONST_TANG")

Application.ScreenUpdating = False

Workbooks("GRAND_MASTER_1.xlsm").Activate

Application.Worksheets("MICRO_CONST_TANG").Activate

Set micro_mat_sht = Worksheets("MICRO_CONST_TANG")

    With micro_mat_sht

        Set mat9_wefttow = Range(Cells(38, 30), Cells(43, 35))

        Set mat9_warptow = Range(Cells(38, 39), Cells(43, 44))

        Set mat9_weftramp = Range(Cells(38, 48), Cells(43, 53))

        Set mat9_warpramp = Range(Cells(38, 57), Cells(43, 62))

        Set id_mat = Range(Cells(29, 14), Cells(34, 19))

    End With

    'damaged properties

    mat9_wefttow_damg = Application.WorksheetFunction.MMult(id_mat, mat9_wefttow)

    mat9_warptow_damg = Application.WorksheetFunction.MMult(id_mat, mat9_warptow)

    mat9_weftramp_damg = Application.WorksheetFunction.MMult(id_mat, mat9_weftramp)

    mat9_warpramp_damg = Application.WorksheetFunction.MMult(id_mat, mat9_warpramp)

'loop_incr = 1
```

```
For j = 1 To 6

If Len(Dir("C:\MsFEM_Project\GRAND_MASTER\REPAIR_INCR_" & CStr(loop_incr) & "\MESO_CRIT_" & CStr(loop_incr) _

   , vbDirectory)) = 0 Then

   MkDir "C:\MsFEM_Project\GRAND_MASTER\REPAIR_INCR_" & CStr(loop_incr) & "\MESO_CRIT_" & CStr(loop_incr)

End If

Open "C:\MsFEM_Project\GRAND_MASTER\REPAIR_INCR_" & CStr(loop_incr) & "\MESO_CRIT_" & CStr(loop_incr) & _

"\MESO_RVE_" & CStr(const_law_sht.Cells(9 + (4 * j), 4)) & ".bdf" For Output As #1

'C:\MsFEM_Project\GRAND_MASTER\REPAIR_INCR_1\MESO_CRIT_1

Print #1, "$ NASTRAN MESO RVE input file created by Ikara MsFEM"

Print #1, "$ WITS Research Masters Project "

Print #1, "$ Direct Text Input for Nastran System Cell Section"

Print #1, "$ Linear Static Analysis, Database"

Print #1, "SOL 101"

Print #1, "CEND"

Print #1, "TITLE = MSC.Nastran MESO RVE MsFEM job"

Print #1, "ECHO = NONE"

Print #1, "MPC = 4"

Print #1, "SUBCASE 1"

Print #1, "   SUBTITLE = Default"

Print #1, "   SPC = 2"

'Print #1, "   DISPLACEMENT(SORT1,REAL)=ALL"

Print #1, "   SPCFORCES(SORT1,REAL)=ALL"

Print #1, "   STRESS(SORT1,PUNCH,REAL,VONMISES,CENTER)=ALL"

Print #1, "   AUTOSPC(NOPRINT)=YES" 'Remove singularity table

Print #1, "BEGIN BULK"

Print #1, "PARAM    POST    0"
```

```
Print #1, "PARAM   PRTMAXIM YES"
'ELEMENTS AND PROPERTIES
Dim elem_list_sht, node_list_sht As Variant
Dim elem_range_1, elem_range_2, elem_range_3, elem_range_4, node_range As Variant
Dim elem_range_5, elem_range_6, elem_range_7, elem_range_11 As Variant
Set elem_list_sht = Worksheets("MESO_RVE_ELEM_PROPS")
   With elem_list_sht
      Set elem_range_1 = Range(.Cells(3, 3), .Cells(10002, 10))
      Set elem_range_11 = Range(.Cells(3, 14), .Cells(50002, 21))
      Set elem_range_2 = Range(.Cells(3, 25), .Cells(4002, 32))
      Set elem_range_3 = Range(.Cells(3, 36), .Cells(3002, 42))
      Set elem_range_4 = Range(.Cells(3, 47), .Cells(3002, 54))
      Set elem_range_5 = Range(.Cells(3, 58), .Cells(4002, 65))
      Set elem_range_6 = Range(.Cells(3, 69), .Cells(3002, 76))
      Set elem_range_7 = Range(.Cells(3, 80), .Cells(3002, 87))
   End With
Set node_list_sht = Worksheets("MESO_RVE_NODES")
   With node_list_sht
      Set node_range = Range(.Cells(3, 3), .Cells(44423, 6))
   End With
Dim num_elem_1, num_elem_11, num_elem_2, num_elem_3, i, num_nodes As Double
Dim num_elem_4, num_elem_5, num_elem_6, num_elem_7 As Double
Print #1, "$"
Print #1, "$ Elements and Element Properties for region : matrix_prop"
Print #1, "PSOLID,1,1,0"
Print #1, "$ Pset: ' matrix_prop' will be imported as: ' psolid.1'"
Print #1, "$"
Print #1, "$"
```

```
Print #1, "$ Elements and Element Properties for region : failed_matrix_prop"

Print #1, "PSOLID,11,11,0"

Print #1, "$ Pset: ' matrix_prop' will be imported as: ' psolid.11'"

Print #1, "$"


num_elem_1 = 10000

For i = 1 To num_elem_1

   If elem_list_sht.Cells(59999 + 1 * (i - 1), 108 + loop_incr * 18) = 0 Then

   Print #1, "CPENTA," + CStr(elem_range_1(i, 1)) + "," + CStr(elem_range_1(i, 2)) + "," + CStr(elem_range_1(i, 3)) + "," _

            + CStr(elem_range_1(i, 4)) + "," + CStr(elem_range_1(i, 5)) + "," + CStr(elem_range_1(i, 6)) + "," _

            + CStr(elem_range_1(i, 7)) + "," + CStr(elem_range_1(i, 8))

   Else

   Print #1, "CPENTA," + CStr(elem_range_1(i, 1)) + ",11," + CStr(elem_range_1(i, 3)) + "," _

            + CStr(elem_range_1(i, 4)) + "," + CStr(elem_range_1(i, 5)) + "," + CStr(elem_range_1(i, 6)) + "," _

            + CStr(elem_range_1(i, 7)) + "," + CStr(elem_range_1(i, 8))

   End If

Next i

num_elem_11 = 50000

For i = 1 To num_elem_11 Step 2

   If elem_list_sht.Cells(59999 + 0.5 * (i - 1), 110 + loop_incr * 18) = 0 Then

   Print #1, "CHEXA," + CStr(elem_range_11(i, 1)) + "," + CStr(elem_range_11(i, 2)) + "," + CStr(elem_range_11(i, 3)) + "," _

            + CStr(elem_range_11(i, 4)) + "," + CStr(elem_range_11(i, 5)) + "," + CStr(elem_range_11(i, 6)) + "," _

            + CStr(elem_range_11(i, 7)) + "," + CStr(elem_range_11(i, 8)) + ","

   Print #1, "+," + CStr(elem_range_11(i + 1, 1)) + "," + CStr(elem_range_11(i + 1, 2))

   Else
```

```
   Print #1, "CHEXA," + CStr(elem_range_11(i, 1)) + ",11," + CStr(elem_range_11(i, 3)) + "," _
            + CStr(elem_range_11(i, 4)) + "," + CStr(elem_range_11(i, 5)) + "," + CStr(elem_range_11(i, 6)) + "," _
            + CStr(elem_range_11(i, 7)) + "," + CStr(elem_range_11(i, 8)) + ","
   Print #1, "+," + CStr(elem_range_11(i + 1, 1)) + "," + CStr(elem_range_11(i + 1, 2))
   End If
Next i
Print #1, "$"
Print #1, "$ Elements and Element Properties for region : prop_weft_tow"
Print #1, "PSOLID,2,2,0"
Print #1, "$ Pset: ' prop_weft_tow' will be imported as: ' psolid.2'"
Print #1, "$"
Print #1, "$"
Print #1, "$ Elements and Element Properties for region : failed_prop_weft_tow"
Print #1, "PSOLID,22,22,0"
Print #1, "$ Pset: ' prop_weft_tow' will be imported as: ' psolid.22'"
Print #1, "$"

num_elem_2 = 4000
For i = 1 To num_elem_2 Step 2
   If elem_list_sht.Cells(59999 + 0.5 * (i - 1), 112 + loop_incr * 18) = 0 Then
   Print #1, "CHEXA," + CStr(elem_range_2(i, 1)) + "," + CStr(elem_range_2(i, 2)) + "," + CStr(elem_range_2(i, 3)) + "," _
            + CStr(elem_range_2(i, 4)) + "," + CStr(elem_range_2(i, 5)) + "," + CStr(elem_range_2(i, 6)) + "," _
            + CStr(elem_range_2(i, 7)) + "," + CStr(elem_range_2(i, 8)) + ","
   Print #1, "+," + CStr(elem_range_2(i + 1, 1)) + "," + CStr(elem_range_2(i + 1, 2))
   Else
   Print #1, "CHEXA," + CStr(elem_range_2(i, 1)) + ",22," + CStr(elem_range_2(i, 3)) + "," _
```

```
        + CStr(elem_range_2(i, 4)) + "," + CStr(elem_range_2(i, 5)) + "," + CStr(elem_range_2(i, 6)) + "," _

        + CStr(elem_range_2(i, 7)) + "," + CStr(elem_range_2(i, 8)) + ","

   Print #1, "+," + CStr(elem_range_2(i + 1, 1)) + "," + CStr(elem_range_2(i + 1, 2))

   End If

Next i

Print #1, "$"

Print #1, "$ Elements and Element Properties for region : prop_weft_ramp_tow_1"

Print #1, "PSOLID,3,4,0"

Print #1, "$ Pset: ' prop_weft_ramp_tow_1' will be imported as: ' psolid.3'"

Print #1, "$"

Print #1, "$"

Print #1, "$ Elements and Element Properties for region : prop_weft_ramp_tow_1"

Print #1, "PSOLID,33,44,0"

Print #1, "$ Pset: ' prop_weft_ramp_tow_1' will be imported as: ' psolid.3'"

Print #1, "$"

num_elem_3 = 3000

For i = 1 To num_elem_3 Step 2

   If elem_list_sht.Cells(59999 + 0.5 * (i - 1), 114 + loop_incr * 18) = 0 Then

   Print #1, "CHEXA," + CStr(elem_range_3(i, 1)) + "," + CStr(elem_range_3(i, 2)) + "," + CStr(elem_range_3(i, 3)) + "," _

        + CStr(elem_range_3(i, 4)) + "," + CStr(elem_range_3(i, 5)) + "," + CStr(elem_range_3(i, 6)) + "," _

        + CStr(elem_range_3(i, 7)) + "," + CStr(elem_range_3(i, 8)) + ","

   Print #1, "+," + CStr(elem_range_3(i + 1, 1)) + "," + CStr(elem_range_3(i + 1, 2))

   Else

   Print #1, "CHEXA," + CStr(elem_range_3(i, 1)) + ",33," + CStr(elem_range_3(i, 3)) + "," _

        + CStr(elem_range_3(i, 4)) + "," + CStr(elem_range_3(i, 5)) + "," + CStr(elem_range_3(i, 6)) + "," _

        + CStr(elem_range_3(i, 7)) + "," + CStr(elem_range_3(i, 8)) + ","
```

```
    Print #1, "+," + CStr(elem_range_3(i + 1, 1)) + "," + CStr(elem_range_3(i + 1, 2))
    End If
Next i
Print #1, "$"
Print #1, "$ Elements and Element Properties for region : prop_weft_ramp_tow_2"
Print #1, "PSOLID,4,4,0"
Print #1, "$ Pset: ' prop_weft_ramp_tow_2' will be imported as: ' psolid.4'"
Print #1, "$"
Print #1, "$"
Print #1, "$ Elements and Element Properties for region : prop_weft_ramp_tow_2"
Print #1, "PSOLID,44,44,0"
Print #1, "$ Pset: ' prop_weft_ramp_tow_2' will be imported as: ' psolid.4'"
Print #1, "$"
num_elem_4 = 3000
For i = 1 To num_elem_4 Step 2
    If elem_list_sht.Cells(59999 + 0.5 * (i - 1), 116 + loop_incr * 18) = 0 Then
    Print #1, "CHEXA," + CStr(elem_range_4(i, 1)) + "," + CStr(elem_range_4(i, 2)) + "," + CStr(elem_range_4(i, 3)) + "," _
            + CStr(elem_range_4(i, 4)) + "," + CStr(elem_range_4(i, 5)) + "," + CStr(elem_range_4(i, 6)) + "," _
            + CStr(elem_range_4(i, 7)) + "," + CStr(elem_range_4(i, 8)) + ","
    Print #1, "+," + CStr(elem_range_4(i + 1, 1)) + "," + CStr(elem_range_4(i + 1, 2))
    Else
    Print #1, "CHEXA," + CStr(elem_range_4(i, 1)) + ",44," + CStr(elem_range_4(i, 3)) + "," _
            + CStr(elem_range_4(i, 4)) + "," + CStr(elem_range_4(i, 5)) + "," + CStr(elem_range_4(i, 6)) + "," _
            + CStr(elem_range_4(i, 7)) + "," + CStr(elem_range_4(i, 8)) + ","
    Print #1, "+," + CStr(elem_range_4(i + 1, 1)) + "," + CStr(elem_range_4(i + 1, 2))
    End If
```

```
Next i

'888888888888888888888888888888888888888888888888888888888888888

Print #1, "$"

Print #1, "$ Elements and Element Properties for region : prop_warp_tow"

Print #1, "PSOLID,5,3,0"

Print #1, "$ Pset: ' prop_warp_tow' will be imported as: ' psolid.5'"

Print #1, "$"

Print #1, "$"

Print #1, "$ Elements and Element Properties for region : prop_warp_tow"

Print #1, "PSOLID,55,33,0"

Print #1, "$ Pset: ' prop_warp_tow' will be imported as: ' psolid.5'"

Print #1, "$"

num_elem_5 = 4000

For i = 1 To num_elem_5 Step 2

   If elem_list_sht.Cells(59999 + 0.5 * (i - 1), 118 + loop_incr * 18) = 0 Then

   Print #1, "CHEXA," + CStr(elem_range_5(i, 1)) + "," + CStr(elem_range_5(i, 2)) + "," + CStr(elem_range_5(i, 3)) + "," _

            + CStr(elem_range_5(i, 4)) + "," + CStr(elem_range_5(i, 5)) + "," + CStr(elem_range_5(i, 6)) + "," _

            + CStr(elem_range_5(i, 7)) + "," + CStr(elem_range_5(i, 8)) + ","

   Print #1, "+," + CStr(elem_range_5(i + 1, 1)) + "," + CStr(elem_range_5(i + 1, 2))

   Else

   Print #1, "CHEXA," + CStr(elem_range_5(i, 1)) + ",55," + CStr(elem_range_5(i, 3)) + "," _

            + CStr(elem_range_5(i, 4)) + "," + CStr(elem_range_5(i, 5)) + "," + CStr(elem_range_5(i, 6)) + "," _

            + CStr(elem_range_5(i, 7)) + "," + CStr(elem_range_5(i, 8)) + ","

   Print #1, "+," + CStr(elem_range_5(i + 1, 1)) + "," + CStr(elem_range_5(i + 1, 2))

   End If

Next i
```

```
Print #1, "$"

Print #1, "$ Elements and Element Properties for region : prop_warp_ramp_tow_1"

Print #1, "PSOLID,6,5,0"

Print #1, "$ Pset: ' prop_warp_ramp_tow_1' will be imported as: ' psolid.6'"

Print #1, "$"

Print #1, "$"

Print #1, "$ Elements and Element Properties for region : prop_warp_ramp_tow_1"

Print #1, "PSOLID,66,55,0"

Print #1, "$ Pset: ' prop_warp_ramp_tow_1' will be imported as: ' psolid.6'"

Print #1, "$"

num_elem_6 = 3000

For i = 1 To num_elem_6 Step 2

   If elem_list_sht.Cells(59999 + 0.5 * (i - 1), 120 + loop_incr * 18) = 0 Then

   Print #1, "CHEXA," + CStr(elem_range_6(i, 1)) + "," + CStr(elem_range_6(i, 2)) + "," + CStr(elem_range_6(i, 3)) + "," _

            + CStr(elem_range_6(i, 4)) + "," + CStr(elem_range_6(i, 5)) + "," + CStr(elem_range_6(i, 6)) + "," _

            + CStr(elem_range_6(i, 7)) + "," + CStr(elem_range_6(i, 8)) + ","

   Print #1, "+," + CStr(elem_range_6(i + 1, 1)) + "," + CStr(elem_range_6(i + 1, 2))

   Else

   Print #1, "CHEXA," + CStr(elem_range_6(i, 1)) + ",66," + CStr(elem_range_6(i, 3)) + "," _

            + CStr(elem_range_6(i, 4)) + "," + CStr(elem_range_6(i, 5)) + "," + CStr(elem_range_6(i, 6)) + "," _

            + CStr(elem_range_6(i, 7)) + "," + CStr(elem_range_6(i, 8)) + ","

   Print #1, "+," + CStr(elem_range_6(i + 1, 1)) + "," + CStr(elem_range_6(i + 1, 2))

   End If

Next i

Print #1, "$"

Print #1, "$ Elements and Element Properties for region : prop_warp_ramp_tow_2"
```

```
Print #1, "PSOLID,7,5,0"

Print #1, "$ Pset: ' prop_warp_ramp_tow_2' will be imported as: ' psolid.7'"

Print #1, "$"

Print #1, "$"

Print #1, "$ Elements and Element Properties for region : prop_warp_ramp_tow_2"

Print #1, "PSOLID,77,55,0"

Print #1, "$ Pset: ' prop_warp_ramp_tow_2' will be imported as: ' psolid.7'"

Print #1, "$"

num_elem_7 = 3000

For i = 1 To num_elem_7 Step 2

   If elem_list_sht.Cells(59999 + 0.5 * (i - 1), 122 + loop_incr * 18) = 0 Then

   Print #1, "CHEXA," + CStr(elem_range_7(i, 1)) + "," + CStr(elem_range_7(i, 2)) + "," + CStr(elem_range_7(i, 3)) + "," _

            + CStr(elem_range_7(i, 4)) + "," + CStr(elem_range_7(i, 5)) + "," + CStr(elem_range_7(i, 6)) + "," _

            + CStr(elem_range_7(i, 7)) + "," + CStr(elem_range_7(i, 8)) + ","

   Print #1, "+," + CStr(elem_range_7(i + 1, 1)) + "," + CStr(elem_range_7(i + 1, 2))

   Else

   Print #1, "CHEXA," + CStr(elem_range_7(i, 1)) + ",77," + CStr(elem_range_7(i, 3)) + "," _

            + CStr(elem_range_7(i, 4)) + "," + CStr(elem_range_7(i, 5)) + "," + CStr(elem_range_7(i, 6)) + "," _

            + CStr(elem_range_7(i, 7)) + "," + CStr(elem_range_7(i, 8)) + ","

   Print #1, "+," + CStr(elem_range_7(i + 1, 1)) + "," + CStr(elem_range_7(i + 1, 2))


   End If

Next i

'MATERIAL PROPERTIES

Print #1, "$"

Print #1, "$"
```

```
Print #1, "$ Referenced Material Records"
Print #1, "$ Material Record : mat_matrix"
Print #1, "$ Description of Material :"
Print #1, "MAT9,1,55.35,29.79,29.79, , , ,55.35,"
Print #1, "+,29.79,,,,55.35,"
Print #1, "+,13,,,13,,13"
Print #1, "$"
Print #1, "$"
Print #1, "$  mat9_wefttow layer properties"
Print #1, "$"
Print #1, "$"
Print #1, "MAT9,2," + CStr(mat9_wefttow(1, 1) * 100) + "," + CStr(mat9_wefttow(1, 2) * 0) + "," _
+ CStr(mat9_wefttow(1, 3) * 0) + ", , , ," + CStr(mat9_wefttow(2, 2) * 100) + ","
Print #1, "+," + CStr(mat9_wefttow(2, 3) * 0) + ", , , ," + CStr(mat9_wefttow(3, 3) * 100) + ","
Print #1, "+," + CStr(mat9_wefttow(4, 4) * 100) + ", , ," + CStr(mat9_wefttow(5, 5) * 100) + ", ," _
+ CStr(mat9_wefttow(6, 6) * 100)
Print #1, "$"
Print #1, "$"
Print #1, "$ mat9_warptow layer properties"
Print #1, "$"
Print #1, "$"
Print #1, "MAT9,3," + CStr(mat9_warptow(1, 1) * 100) + "," + CStr(mat9_warptow(1, 2) * 0) + "," _
+ CStr(mat9_warptow(1, 3) * 0) + ", , , ," + CStr(mat9_warptow(2, 2) * 100) + ","
Print #1, "+," + CStr(mat9_warptow(2, 3) * 0) + ", , , ," + CStr(mat9_warptow(3, 3) * 100) + ","
Print #1, "+," + CStr(mat9_warptow(4, 4) * 100) + ", , ," + CStr(mat9_warptow(5, 5) * 100) + ", ," _
+ CStr(mat9_warptow(6, 6) * 100)
```

```
Print #1, "$"

Print #1, "$"

Print #1, "$  mat9_weftramp layer properties"

Print #1, "$"

Print #1, "$"

Print #1, "MAT9,4," + CStr(mat9_weftramp(1, 1) * 100) + "," + CStr(mat9_weftramp(1, 2) * 0) + "," _

+ CStr(mat9_weftramp(1, 3) * 0) + ", , , ," + CStr(mat9_weftramp(2, 2) * 100) + ","

Print #1, "+," + CStr(mat9_weftramp(2, 3) * 0) + ", , , ," + CStr(mat9_weftramp(3, 3) * 100) + ","

Print #1, "+," + CStr(mat9_weftramp(4, 4) * 100) + ", , ," + CStr(mat9_weftramp(5, 5) * 100) + ",," _

+ CStr(mat9_weftramp(6, 6) * 100)

Print #1, "$"

Print #1, "$"

Print #1, "$  mat9_warpramp layer properties"

Print #1, "$"

Print #1, "$"

Print #1, "MAT9,5," + CStr(mat9_warpramp(1, 1) * 100) + "," + CStr(mat9_warpramp(1, 2) * 0) + "," _

+ CStr(mat9_warpramp(1, 3) * 0) + ",0 ,0 ,0 ," + CStr(mat9_warpramp(2, 2) * 100) + ","

Print #1, "+," + CStr(mat9_warpramp(2, 3) * 0) + ",0 ,0 ,0 ," + CStr(mat9_warpramp(3, 3) * 100) + ","

Print #1, "+," + CStr(mat9_warpramp(4, 4) * 100) + ",0 ,0 ," + CStr(mat9_warpramp(5, 5) * 100) + ",0 ," _

+ CStr(mat9_warpramp(6, 6) * 100)

'FAILED MATERIAL PROPERTIES

Print #1, "$"

Print #1, "$"

Print #1, "$FAILED MATERIAL PROPERTIES"

Print #1, "$"
```

```
Print #1, "$"

Print #1, "$ Referenced Material Records"

Print #1, "$ Material Record : mat_matrix_damg"

Print #1, "MAT9,11,0.138,,, , , ,0.138,"

Print #1, "+,,,,,2.130,"

Print #1, "+,0.1135,,,0.1135,,0.055"

Print #1, "$"

Print #1, "$"

Print #1, "$  mat9_wefttow_damg layer properties"

Print #1, "$"

Print #1, "$"

Print #1, "MAT9,22," + CStr(mat9_wefttow_damg(1, 1) * 0.1) + "," + CStr(mat9_wefttow_damg(1, 2) * 0.1) + "," _

+ CStr(mat9_wefttow_damg(1, 3) * 0.1) + ",0,0,0," + CStr(mat9_wefttow_damg(2, 2) * 0.1) + ","

Print #1, "+," + CStr(mat9_wefttow_damg(2, 3) * 0.1) + ",0,0,0," + CStr(mat9_wefttow_damg(3, 3) * 0.1) + ","

Print #1, "+," + CStr(mat9_wefttow_damg(4, 4) * 0.1) + ",0,0," + CStr(mat9_wefttow_damg(5, 5) * 0.1) + ",0," _

+ CStr(mat9_wefttow_damg(6, 6) * 0.1)

Print #1, "$"

Print #1, "$"

Print #1, "$ mat9_warptow_damg layer properties"

Print #1, "$"

Print #1, "$"

Print #1, "MAT9,33," + CStr(mat9_warptow_damg(1, 1) * 0.1) + "," + CStr(mat9_warptow_damg(1, 2) * 0.1) + "," _

+ CStr(mat9_warptow_damg(1, 3) * 0.1) + ",0,0,0," + CStr(mat9_warptow_damg(2, 2) * 0.1) + ","

Print #1, "+," + CStr(mat9_warptow_damg(2, 3) * 0.1) + ",0,0,0," + CStr(mat9_warptow_damg(3, 3) * 0.1) + ","
```

```
Print #1, "+," + CStr(mat9_warptow_damg(4, 4) * 0.1) + ",0,0," + CStr(mat9_warptow_damg(5, 5) * 0.1) + ",0," _

+ CStr(mat9_warptow_damg(6, 6) * 0.1)

Print #1, "$"

Print #1, "$"

Print #1, "$  mat9_weftramp_damg layer properties"

Print #1, "$"

Print #1, "$"

Print #1, "MAT9,44," + CStr(mat9_weftramp_damg(1, 1) * 0.1) + "," + CStr(mat9_weftramp_damg(1, 2) * 0.1) + "," _

+ CStr(mat9_weftramp_damg(1, 3) * 0.1) + ",0,0,0," + CStr(mat9_weftramp_damg(2, 2) * 0.1) + ","

Print #1, "+," + CStr(mat9_weftramp_damg(2, 3) * 0.1) + ",0,0,0," + CStr(mat9_weftramp_damg(3, 3) * 0.1) + ","

Print #1, "+," + CStr(mat9_weftramp_damg(4, 4) * 0.1) + ",0,0," + CStr(mat9_weftramp_damg(5, 5) * 0.1) + ",0," _

+ CStr(mat9_weftramp_damg(6, 6) * 0.1)

Print #1, "$"

Print #1, "$"

Print #1, "$  mat9_warpramp_damg layer properties"

Print #1, "$"

Print #1, "$"

Print #1, "MAT9,55," + CStr(mat9_warpramp_damg(1, 1) * 0.1) + "," + CStr(mat9_warpramp_damg(1, 2) * 0.1) + "," _

+ CStr(mat9_warpramp_damg(1, 3) * 0.1) + ",0,0,0," + CStr(mat9_warpramp_damg(2, 2) * 0.1) + ","

Print #1, "+," + CStr(mat9_warpramp_damg(2, 3) * 0.1) + ",0,0,0," + CStr(mat9_warpramp_damg(3, 3) * 0.1) + ","

Print #1, "+," + CStr(mat9_warpramp_damg(4, 4) * 0.1) + ",0,0," + CStr(mat9_warpramp_damg(5, 5) * 0.1) + ",0," _

+ CStr(mat9_warpramp_damg(6, 6) * 0.1)
```

```
'UPDATE STRAIN ENTRIES

Set MPC_EQ_DISP_sht = Worksheets("MESO_EQUIV_DISP")

MPC_EQ_DISP_sht.Cells(3, 6) = const_law_sht.Cells(9 + (4 * j), 11) 'E11

MPC_EQ_DISP_sht.Cells(3, 7) = const_law_sht.Cells(9 + (4 * j), 12) 'E21

MPC_EQ_DISP_sht.Cells(3, 8) = const_law_sht.Cells(9 + (4 * j), 13) 'E31

MPC_EQ_DISP_sht.Cells(4, 7) = const_law_sht.Cells(10 + (4 * j), 12) 'E22

MPC_EQ_DISP_sht.Cells(4, 8) = const_law_sht.Cells(10 + (4 * j), 13) 'E32

MPC_EQ_DISP_sht.Cells(5, 8) = const_law_sht.Cells(11 + (4 * j), 13) 'E33

'INSERT MPC CONSTRAINTS

meso_mpc_pairs

'INSERT NODES

Print #1, "$ Nodes of the Entire Model"

num_nodes = 44421

For i = 1 To num_nodes

   Print #1, "GRID" + "," + CStr(node_range(i, 1)) + ",," + CStr(node_range(i, 2)) + "," + CStr(node_range(i, 3)) + "," + _

            CStr(node_range(i, 4))

Next i

Print #1, "$"

Print #1, "$"

Print #1, "$ Loads for Load Case : Default"

Print #1, "SPCADD,2,1"

Print #1, "$ Displacement Constraints of Load Set : complete_fix"

Print #1, "SPC1,1,123456,7770"

Print #1, "$ Referenced Coordinate Frames"

Print #1, "CORD2R   6              4.      .30001055.3     4.0000156.9468155.299968"

Print #1, "        4.000008.2999788-1.34680"

Print #1, "CORD2R   7              1.100002.4      1.8000041.0999982.5470951.800004"

Print #1, "        -1.04709.39999621.800004"
```

```
Print #1, "CORD2R   8              1.986667.14666671.8     1.3354982.7513421.8"
Print #1, "         4.591342.79783571.8"
Print #1, "CORD2R   10             2.146667.11333332.2     2.8926773.0973732.2"
Print #1, "         -.837372.85934332.2"
Print #1, "CORD2R   11             1.8     .15333331.9866671.8     2.75837 2.637926"
Print #1, "         1.8     .8045927-.618369"
Print #1, "CORD2R   12             2.2     .16     2.04    2.2     3.0748221.311294"
Print #1, "         2.2     .88870564.954822"
Print #1, "$"
Print #1, "ENDDATA 7b83e1dd"
Close #1
Next j
```

End Sub

8. VBA Script meso_con_tangent ()

```
Option Explicit
Dim loop_incr, meso_lim_loop, micro_lim_loop As Integer
Option Base 1
Sub meso_con_tangent(loop_incr)
'READ F06 FILE
Dim file1 As String
Dim volumebook, stressbook As Workbooks
Dim volumesht, stresssht, resultsht As Worksheet
Dim volume_sum_rep, volume_mult As Variant
Dim norm_x, norm_y, norm_z, shr_xy, shr_yz, shr_zx, von_mises As Double
Dim elem, i, j, k As Single
Dim fNameAndPath As Variant
Dim TextLine, fName
Dim myCount, counter, a_elm, a_11, a_22, a_33, a_44, A_ii As Long
Dim counterp, a_elmp, a_11p, a_22p, a_33p, a_44p As Long
Dim numelem As Double
Dim outputFile
Dim range_row(1, 45000) As Variant
Dim stress_matrix(45000, 8) As Variant
Dim r As Integer
numelem = 45000
'loop_incr = 1

Set resultsht = Worksheets("MESO_CONST_TANG")
For j = 1 To 6 ' loop through results files
```

```
'file1 = "C:\MsFEM_Project\GRAND_MASTER\REPAIR_INCR_" & CStr(loop_incr) & "\MESO_CRIT_" & CStr(loop_incr) & _

"\MESO_RVE_" & CStr(resultsht.Cells(9 + (4 * j), 4)) & ".f06"

    myCount = 1

    counter = 1

    counterp = 1

    a_elm = 8

    a_11 = 9

    a_22 = 11

    a_33 = 12

    a_44 = 14

    a_elmp = 2240015

    a_11p = 2240016

    a_22p = 2240018

    a_33p = 2240019

    a_44p = 2240021

    numelem = 45000

    'working input folder location

    fNameAndPath = "C:\MsFEM_Project\GRAND_MASTER\REPAIR_INCR_" & CStr(loop_incr) & "\MESO_CRIT_" & CStr(loop_incr) & _

    "\MESO_RVE_" & CStr(resultsht.Cells(10 + (4 * j), 4)) & ".pch"

    'working output folder location

    'outputFile = "C:\MsFEM_Project\GRAND_MASTER\REPAIR_INCR_" & CStr(loop_incr) & "\MESO_CRIT_" & CStr(loop_incr) & _

    "\MESO_RVE_E11_cut.txt"

    'Open outputFile For Output Shared As #2 ' Open file.

    Open fNameAndPath For Input As #1    ' Open file.


    Do While Not EOF(1)    ' Loop until end of file.
```

```
Line Input #1, TextLine    ' Read line into variable.

'HEXA ELEMENT RESULTS

A_ii = a_elm + (counter - 1) * 64

If A_ii = myCount And Left(TextLine, 3) = "   " Then

   'Print #2, TextLine & " ELEMENT NO " & CStr(myCount) & "   " & CStr(counter) & "   " & Mid(TextLine, 6, 5) ' write variable to output file.

   'counter = counter + 1

   'elem

      stress_matrix(counter, 1) = CLng((Mid(TextLine, 6, 5)))

End If

A_ii = a_11 + (counter - 1) * 64

If A_ii = myCount And Left(TextLine, 6) = "-CONT-" Then

   'Print #2, TextLine & " X and XY  " & CStr(myCount) & "   " & CStr(counter) & " X " & Mid(TextLine, 42, 13) & " XY " & Mid(TextLine, 60, 13) ' write variable to output file.

   'counter = counter + 1

   'norm_x

      stress_matrix(counter, 2) = CSng(Mid(TextLine, 42, 13))

   'shr_xy

      stress_matrix(counter, 5) = CSng(Mid(TextLine, 60, 13))

End If

A_ii = a_22 + (counter - 1) * 64

If A_ii = myCount And Left(TextLine, 6) = "-CONT-" Then

   'Print #2, TextLine & " VM  " & CStr(myCount) & " VM " & Mid(TextLine, 60, 13) ' write variable to output file.

   'counter = counter + 1

   'von_mises

      stress_matrix(counter, 8) = CSng(Mid(TextLine, 60, 13))

End If

A_ii = a_33 + (counter - 1) * 64
```

```
If A_ii = myCount And Left(TextLine, 6) = "-CONT-" Then

   'Print #2, TextLine & " Y and YZ  " & CStr(myCount) & "   " & CStr(counter) & " Y " & Mid(TextLine, 24, 13) & " YZ " & Mid(TextLine, 42, 13) ' write variable to output file.

   'counter = counter + 1

   'norm_y

      stress_matrix(counter, 3) = CSng(Mid(TextLine, 24, 13))

   'shr_yz

      stress_matrix(counter, 6) = CSng(Mid(TextLine, 42, 13))

End If

A_ii = a_44 + (counter - 1) * 64

If A_ii = myCount And Left(TextLine, 6) = "-CONT-" Then

   'Print #2, TextLine & " Z and ZX  " & CStr(myCount) & "   " & CStr(counter) & " Z " & Mid(TextLine, 24, 13) & " ZX " & Mid(TextLine, 42, 13) ' write variable to output file.

   'norm_z

      stress_matrix(counter, 4) = CSng(Mid(TextLine, 24, 13))

   'shr_zx

      stress_matrix(counter, 7) = CSng(Mid(TextLine, 42, 13))

   counter = counter + 1

End If

'PENTA ELEMENT RESULTS

A_ii = a_elmp + (counterp - 1) * 50

If A_ii = myCount And Left(TextLine, 3) = "   " Then

   'Print #2, TextLine & " ELEMENT NO " & CStr(myCount) & "   " & CStr(counterp + counter - 1) & "   " & Mid(TextLine, 6, 5) ' write variable to output file.

   'counter = counter + 1

   'elem

   stress_matrix(counterp + counter - 1, 1) = CLng((Mid(TextLine, 6, 5)))

End If

A_ii = a_11p + (counterp - 1) * 50
```

```
If A_ii = myCount And Left(TextLine, 6) = "-CONT-" Then

    'Print #2, TextLine & " X and XY  " & CStr(myCount) & "   " & CStr(counterp + counter - 1) & " X " & Mid(TextLine, 42, 13) & " XY " & Mid(TextLine, 60, 13) ' write variable to output file.

    'counter = counter + 1

    'norm_x

       stress_matrix(counterp + counter - 1, 2) = CSng(Mid(TextLine, 42, 13))

    'shr_xy

       stress_matrix(counterp + counter - 1, 5) = CSng(Mid(TextLine, 60, 13))

End If

A_ii = a_22p + (counterp - 1) * 50

If A_ii = myCount And Left(TextLine, 6) = "-CONT-" Then

    'Print #2, TextLine & " VM  " & CStr(myCount) & " VM " & Mid(TextLine, 60, 13)  ' write variable to output file.

    'counter = counter + 1

    'von_mises

       stress_matrix(counter + counterp - 1, 8) = CSng(Mid(TextLine, 60, 13))

End If

A_ii = a_33p + (counterp - 1) * 50

If A_ii = myCount And Left(TextLine, 6) = "-CONT-" Then

    'Print #2, TextLine & " Y and YZ  " & CStr(myCount) & "   " & CStr(counterp + counter - 1) & " Y " & Mid(TextLine, 24, 13) & " YZ " & Mid(TextLine, 42, 13) ' write variable to output file.

    'counter = counter + 1

    'norm_y

       stress_matrix(counter + counterp - 1, 3) = CSng(Mid(TextLine, 24, 13))

    'shr_yz

       stress_matrix(counter + counterp - 1, 6) = CSng(Mid(TextLine, 42, 13))

End If

A_ii = a_44p + (counterp - 1) * 50
```

```
        If A_ii = myCount And Left(TextLine, 6) = "-CONT-" Then

            'Print #2, TextLine & " Z and ZX  " & CStr(myCount) & "    " & CStr(counterp + counter - 1) & " Z " & Mid(TextLine, 24, 13) & " ZX " & Mid(TextLine, 42, 13) ' write variable to output file.

            'norm_z

                stress_matrix(counter + counterp - 1, 4) = CSng(Mid(TextLine, 24, 13))

            'shr_zx

                stress_matrix(counter + counterp - 1, 7) = CSng(Mid(TextLine, 42, 13))

            counterp = counterp + 1

        End If

        myCount = myCount + 1

    Loop

    'Close #2

    Close #1    ' Close file.

    'BubbleSort stress_matrix, 1

    'CALCULATE COLUMNS

    Set volumesht = Worksheets("MESO_ELEMENT_VOLUMES")

    Set resultsht = Worksheets("MESO_CONST_TANG")

    volume_sum_rep = 1 / (WorksheetFunction.Sum(Range(volumesht.Cells(3, 3), volumesht.Cells(2 + numelem, 3)).Value))

    Dim Fn As Object

    Set Fn = Application.WorksheetFunction

    'range_row = Fn.Transpose(Range(volumesht.Cells(3, 3), volumesht.Cells(2 + numelem, 3)))

    For k = 1 To numelem

        range_row(1, k) = Fn.Index(Range(volumesht.Cells(3, 3), volumesht.Cells(2 + numelem, 3)), stress_matrix(k, 1))

    Next

    For r = 1 To 6

        volume_mult = Fn.MMult(range_row, Fn.Index(stress_matrix, 0, 1 + r))

        resultsht.Cells(3 + r, 13 + j) = Fn.MMult(volume_mult, volume_sum_rep)
```

```
    Next r
Next j
End Sub
```

9. VBA Script repair_bdf_file ()

```
Option Explicit

Option Base 1

Dim E_crit_repair, E_crit_micro, C_equi_meso, C_equi_micro As Variant

Dim loop_incr, meso_lim_loop, micro_lim_loop As Integer

Dim test As Integer

'Sub repair_bdf_file(loop_incr)

Sub repair_bdf_file()

'Declare variables

Dim elem_list_sht, node_list_sht, load_sht, crit_strns_sht, micro_mat_sht As Worksheet

'Dim meso_mat_sht As Worksheet

Dim elem_range_1, elem_range_2, elem_range_3, elem_range_4, elem_range_5, node_range As Variant

Dim elem_range_6, elem_range_7, elem_range_8, elem_range_9, elem_range_10 As Variant

Dim elem_range_11, elem_range_12, elem_range_13, elem_range_14, elem_range_15 As Variant

Dim elem_range_16, elem_range_17, elem_range_18, elem_range_19, elem_range_20 As Variant

Dim mat_core, mat9_0_90_deg, mat9_45_deg, mat_matrix As Variant

Dim mat9_0_deg_damg, mat9_90_deg_damg, mat9_45_deg_damg As Variant

Dim fail_mat_sht, meso_mat_sht As Variant

Dim loop_incr As Integer

Dim distrib_load As Double

Dim s_f_elem As Integer

'Initialise variables

Set meso_mat_sht = Worksheets("MESO_PROPERTIES")

   With meso_mat_sht

      Set mat9_45_deg = Range(.Cells(6, 17), .Cells(11, 22))

      Set mat9_0_90_deg = Range(.Cells(6, 3), .Cells(11, 8))
```

```
    End With

'INITIALISE ELEMENTS AND NODES

Set elem_list_sht = Worksheets("REPAIR_ELEM_PROPS")

    With elem_list_sht

        Set elem_range_1 = Range(.Cells(3, 3), .Cells(1874, 10)) ' CORE_ALUMINIUM ELEMENTS

        Set elem_range_2 = Range(.Cells(3, 14), .Cells(602, 21)) ' CORE_NOMEX ELEMENTS

        Set elem_range_3 = Range(.Cells(3, 14), .Cells(11554, 21)) ' FILLER_0_90_PLY_1 ELEMENTS

        Set elem_range_4 = Range(.Cells(3, 36), .Cells(194, 43)) ' FILLER_45_PLY_2 ELEMENTS

        Set elem_range_5 = Range(.Cells(3, 102), .Cells(386, 109)) ' FILLER_0_90_PLY_3 ELEMENTS

        Set elem_range_6 = Range(.Cells(3, 25), .Cells(1266, 32)) ' FIRST_STAGE_0_90_PLY_1 or *_PLY_3

        Set elem_range_7 = Range(.Cells(3, 47), .Cells(634, 54)) ' FIRST_STAGE_45_PLY_2

        'Set elem_range_8 = Range(.Cells(3, 36), .Cells(26, 43)) ' FIRST_STAGE_0_90_PLY_3

        Set elem_range_9 = Range(.Cells(3, 113), .Cells(26, 120)) ' FOAM

        Set elem_range_10 = Range(.Cells(3, 124), .Cells(26, 131)) ' MATRIX

        Set elem_range_11 = Range(.Cells(3, 58), .Cells(882, 65)) ' REPAIR_0_90_PLY_1 or *_PLY_3

        Set elem_range_12 = Range(.Cells(3, 69), .Cells(442, 76)) ' REPAIR_45_PLY_2

        'Set elem_range_13 = Range(.Cells(3, 36), .Cells(26, 43)) ' REPAIR_0_90_PLY_3

        Set elem_range_14 = Range(.Cells(3, 80), .Cells(3362, 87)) ' SEC_STAGE_0_90_PLY_1 or *_PLY_#

        'Set elem_range_15 = Range(.Cells(3, 36), .Cells(26, 43)) ' SEC_STAGE_0_90_PLY_2

        Set elem_range_16 = Range(.Cells(3, 91), .Cells(1682, 98)) ' SEC_STAGE_45_PLY_3

        'Set elem_range_17 = Range(.Cells(3, 36), .Cells(26, 43)) ' SEC_STAGE_45_PLY_4

        'Set elem_range_18 = Range(.Cells(3, 36), .Cells(26, 43)) ' SEC_STAGE_0_90_PLY_5

        'Set elem_range_19 = Range(.Cells(3, 36), .Cells(26, 43)) ' SEC_STAGE_0_90_PLY_6

    End With
```

```
Set node_list_sht = Worksheets("REPAIR_NODES")
   With node_list_sht
      Set node_range = Range(.Cells(3, 3), .Cells(15500, 6))
   End With
Set load_sht = Worksheets("LOAD_CONTROL")
'LOOP THROUGH LOADS INCREMENTS
For loop_incr = 2 To 2
'loop_incr = 1
'CHECK FOR FILE EXISTANCE AND CREATE ONE IF IT DOESN'T EXIST
If Len(Dir("C:\MsFEM_Project\GRAND_MASTER_2\REPAIR_INCR_" & CStr(loop_incr), vbDirectory)) = 0 Then
   MkDir "C:\MsFEM_Project\GRAND_MASTER_2\REPAIR_INCR_" & CStr(loop_incr)
End If
Open "C:\MsFEM_Project\GRAND_MASTER_2\REPAIR_INCR_" & CStr(loop_incr) & "\repair_" & CStr(loop_incr) & ".bdf" For Output As #1
Print #1, "$ NASTRAN Stepped Repair input file created by Ikara MsFEM"
Print #1, "$ WITS Research Masters Project"
Print #1, "$ Direct Text Input for Nastran System Cell Section"
Print #1, "$ Linear Static Analysis, Database"
Print #1, "SOL 101"
Print #1, "CEND"
Print #1, "TITLE = MSC.Nastran repair MsFEM job"
Print #1, "ECHO = NONE"
Print #1, ""
Print #1, "SUBCASE 1"
Print #1, "   SUBTITLE = select"
Print #1, "   SPC = 2"
Print #1, "   LOAD = 2"
Print #1, "   DISPLACEMENT(SORT1,REAL)=ALL"
```

```
Print #1, "   SPCFORCES(SORT1,REAL)=ALL"

Print #1, "   STRAIN=ALL" 'selected zones

Print #1, "   STRESS=ALL" 'selected zones

Print #1, "   GPFORCE=ALL" 'selected zones

Print #1, "   AUTOSPC(NOPRINT)=YES" 'Remove singularity table

Print #1, "BEGIN BULK"

Print #1, "$ Direct Text Input for Bulk Data"

Print #1, "PARAM    POST    0"

Print #1, "PARAM   PRTMAXIM YES"

'ELEMENTS AND PROPERTIES

Dim num_elem_1, num_elem_2, num_elem_3, num_elem_4, num_elem_5, i, num_nodes As Integer

Dim num_elem_6, num_elem_7, num_elem_8, num_elem_9, num_elem_10 As Integer

Dim num_elem_11, num_elem_12, num_elem_13, num_elem_14, num_elem_15 As Integer

Dim num_elem_16, num_elem_17, num_elem_18, num_elem_19, num_elem_20 As Integer

'CORE 1 PROPERTIES HERE

Print #1, "$"

Print #1, "$ Elements and Element Properties for region : prop_core_alumi"

Print #1, "PSOLID,1,1,0"

Print #1, "$ Pset: ' prop_core_alumi' will be imported as: ' psolid.1'"

Print #1, "$"

num_elem_1 = 1872

For i = 1 To num_elem_1 Step 2

   If i < 0 Then 'CHECKS FOR FAILURE IN CORE MATERIAL

   Else

      Print #1, "CHEXA," + CStr(elem_range_1(i, 1)) + "," + CStr(elem_range_1(i, 2)) + "," + CStr(elem_range_1(i, 3)) + "," _

          + CStr(elem_range_1(i, 4)) + "," + CStr(elem_range_1(i, 5)) + "," + CStr(elem_range_1(i, 6)) + "," _
```

```
            + CStr(elem_range_1(i, 7)) + "," + CStr(elem_range_1(i, 8)) + ","
        Print #1, "+," + CStr(elem_range_1(i + 1, 1)) + "," + CStr(elem_range_1(i + 1, 2))
    End If
Next i
'CORE 2 PROPERTIES HERE
Print #1, "$"
Print #1, "$ Elements and Element Properties for region : prop_core_nomex"
Print #1, "PSOLID,2,2,0"
Print #1, "$ Pset: ' prop_core_nomex' will be imported as: ' psolid.2'"
Print #1, "$"
num_elem_2 = 600
For i = 1 To num_elem_2 Step 2
    If i < 0 Then 'CHECKS FOR FAILURE IN CORE MATERIAL
    Else
        Print #1, "CHEXA," + CStr(elem_range_2(i, 1)) + "," + CStr(elem_range_2(i, 2)) + "," + CStr(elem_range_2(i, 3)) + "," _
            + CStr(elem_range_2(i, 4)) + "," + CStr(elem_range_2(i, 5)) + "," + CStr(elem_range_2(i, 6)) + "," _
            + CStr(elem_range_2(i, 7)) + "," + CStr(elem_range_2(i, 8)) + ","
        Print #1, "+," + CStr(elem_range_2(i + 1, 1)) + "," + CStr(elem_range_2(i + 1, 2))
    End If
Next i
'FABRIC 0_90 PROPERTIES HERE
Print #1, "$"
Print #1, "$ Elements and Element Properties for region : prop_fst_Stg_0_90"
Print #1, "PSOLID,3,3,0"
Print #1, "$ Pset: ' prop_fst_Stg_0_90' will be imported as: ' psolid.3'"
Print #1, "$"
num_elem_3 = 1264
```

```
For i = 1 To num_elem_3 Step 2

   's_f_elem = elem_range_3(i, 1)

   'If s_f_elem = 11101 Or s_f_elem = 11105 Or s_f_elem = 11109 Or s_f_elem = 11097 Or s_f_elem = 11093 Or s_f_elem = 11089 Or s_f_elem = 11113 Or s_f_elem = 11117 Then 'CHECKS FOR FAILURE IN 0_90_PLY MATERIAL

   If i < 0 Then

      Print #1, "CHEXA," + CStr(elem_range_6(i, 1)) + "," + CStr(33) + "," + CStr(elem_range_6(i, 3)) + "," _

            + CStr(elem_range_6(i, 4)) + "," + CStr(elem_range_6(i, 5)) + "," + CStr(elem_range_6(i, 6)) + "," _

            + CStr(elem_range_6(i, 7)) + "," + CStr(elem_range_6(i, 8)) + ","

      Print #1, "+," + CStr(elem_range_6(i + 1, 1)) + "," + CStr(elem_range_6(i + 1, 2))

   Else

      Print #1, "CHEXA," + CStr(elem_range_6(i, 1)) + "," + CStr(elem_range_6(i, 2)) + "," + CStr(elem_range_6(i, 3)) + "," _

            + CStr(elem_range_6(i, 4)) + "," + CStr(elem_range_6(i, 5)) + "," + CStr(elem_range_6(i, 6)) + "," _

            + CStr(elem_range_6(i, 7)) + "," + CStr(elem_range_6(i, 8)) + ","

      Print #1, "+," + CStr(elem_range_6(i + 1, 1)) + "," + CStr(elem_range_6(i + 1, 2))

   End If

Next i

'FILL +-45 PROPERTIES HERE

Print #1, "$"

Print #1, "$ Elements and Element Properties for region : prop_fill_45"

Print #1, "PSOLID,4,4,0"

Print #1, "$ Pset: ' prop_fill_45' will be imported as: ' psolid.4'"

Print #1, "$"

num_elem_4 = 192

For i = 1 To num_elem_4 Step 2

   If i < 0 Then 'CHECKS FOR FAILURE IN FILL +-45_PLY MATERIAL
```

```
   Else
      Print #1, "CHEXA," + CStr(elem_range_4(i, 1)) + "," + CStr(elem_range_4(i, 2)) + "," + CStr(elem_range_4(i, 3)) + "," _
            + CStr(elem_range_4(i, 4)) + "," + CStr(elem_range_4(i, 5)) + "," + CStr(elem_range_4(i, 6)) + "," _
            + CStr(elem_range_4(i, 7)) + "," + CStr(elem_range_4(i, 8)) + ","
      Print #1, "+," + CStr(elem_range_4(i + 1, 1)) + "," + CStr(elem_range_4(i + 1, 2))
   End If
Next i
'FIRST STAGE -+45 PLY PROPERTIES HERE
Print #1, "$"
Print #1, "$ Elements and Element Properties for region : prop_fst_Stg_45"
Print #1, "PSOLID,5,4,0"
Print #1, "$ Pset: ' prop_fst_Stg_45' will be imported as: ' psolid.5'"
Print #1, "$"
num_elem_5 = 632
For i = 1 To num_elem_5 Step 2
   If i < 0 Then 'CHECKS FOR FAILURE IN FILL elem_range_4+-45_PLY MATERIAL
   Else
      Print #1, "CHEXA," + CStr(elem_range_7(i, 1)) + "," + CStr(elem_range_7(i, 2)) + "," + CStr(elem_range_7(i, 3)) + "," _
            + CStr(elem_range_7(i, 4)) + "," + CStr(elem_range_7(i, 5)) + "," + CStr(elem_range_7(i, 6)) + "," _
            + CStr(elem_range_7(i, 7)) + "," + CStr(elem_range_7(i, 8)) + ","
      Print #1, "+," + CStr(elem_range_7(i + 1, 1)) + "," + CStr(elem_range_7(i + 1, 2))
   End If
Next i
'FABRIC 0_90 PROPERTIES HERE
Print #1, "$"
Print #1, "$ Elements and Element Properties for region : prop_repair_0_90"
```

```
Print #1, "PSOLID,6,3,0"

Print #1, "$ Pset: ' pprop_repair_0_90' will be imported as: ' psolid.5'"

Print #1, "$"

num_elem_6 = 880

For i = 1 To num_elem_6 Step 2

   If i < 0 Then 'CHECKS FOR FAILURE IN FILL elem_range_4+-45_PLY MATERIAL

   Else

      Print #1, "CHEXA," + CStr(elem_range_11(i, 1)) + "," + CStr(elem_range_11(i, 2)) + "," + CStr(elem_range_11(i, 3)) + "," _

            + CStr(elem_range_11(i, 4)) + "," + CStr(elem_range_11(i, 5)) + "," + CStr(elem_range_11(i, 6)) + "," _

            + CStr(elem_range_11(i, 7)) + "," + CStr(elem_range_11(i, 8)) + ","

      Print #1, "+," + CStr(elem_range_11(i + 1, 1)) + "," + CStr(elem_range_11(i + 1, 2))

   End If

Next i

'FIRST STAGE -+45 PLY PROPERTIES HERE

Print #1, "$"

Print #1, "$ Elements and Element Properties for region : prop_repair_45"

Print #1, "PSOLID,7,4,0"

Print #1, "$ Pset: ' prop_repair_45' will be imported as: ' psolid.5'"

Print #1, "$"

num_elem_7 = 440

For i = 1 To num_elem_7 Step 2


   If i < 0 Then 'CHECKS FOR FAILURE IN FILL elem_range_4+-45_PLY MATERIAL

   Else

      Print #1, "CHEXA," + CStr(elem_range_12(i, 1)) + "," + CStr(elem_range_12(i, 2)) + "," + CStr(elem_range_12(i, 3)) + "," _

            + CStr(elem_range_12(i, 4)) + "," + CStr(elem_range_12(i, 5)) + "," + CStr(elem_range_12(i, 6)) + "," _
```

```
            + CStr(elem_range_12(i, 7)) + "," + CStr(elem_range_12(i, 8)) + ","
        Print #1, "+," + CStr(elem_range_12(i + 1, 1)) + "," + CStr(elem_range_12(i + 1, 2))
    End If
Next i
'FABRIC 0_90 PROPERTIES HERE
Print #1, "$"
Print #1, "$ Elements and Element Properties for region : prop_sec_stg_0_90"
Print #1, "PSOLID,8,3,0"
Print #1, "$ Pset: ' prop_sec_stg_0_90' will be imported as: ' psolid.5'"
Print #1, "$"
num_elem_8 = 3360
For i = 1 To num_elem_8 Step 2
    If i < 0 Then 'CHECKS FOR FAILURE IN FILL elem_range_4+-45_PLY MATERIAL


'       Print #1, "+," + CStr(elem_range_14(i + 1, 1)) + "," + CStr(elem_range_14(i + 1, 2))
    Else
        Print #1, "CHEXA," + CStr(elem_range_14(i, 1)) + "," + CStr(elem_range_14(i, 2)) + "," +
CStr(elem_range_14(i, 3)) + "," _
            + CStr(elem_range_14(i, 4)) + "," + CStr(elem_range_14(i, 5)) + "," +
CStr(elem_range_14(i, 6)) + "," _
            + CStr(elem_range_14(i, 7)) + "," + CStr(elem_range_14(i, 8)) + ","
        Print #1, "+," + CStr(elem_range_14(i + 1, 1)) + "," + CStr(elem_range_14(i + 1, 2))
    End If
Next i
'FIRST STAGE -+45 PLY PROPERTIES HERE
Print #1, "$"
Print #1, "$ Elements and Element Properties for region : prop_sec_stg_45"
Print #1, "PSOLID,9,4,0"
Print #1, "$ Pset: ' prop_sec_stg_45' will be imported as: ' psolid.5'"
```

```
Print #1, "$"

num_elem_9 = 1680

For i = 1 To num_elem_9 Step 2

   If i < 0 Then 'CHECKS FOR FAILURE IN FILL +-45_PLY MATERIAL

   Else

      Print #1, "CHEXA," + CStr(elem_range_16(i, 1)) + "," + CStr(elem_range_16(i, 2)) + "," + CStr(elem_range_16(i, 3)) + "," _

            + CStr(elem_range_16(i, 4)) + "," + CStr(elem_range_16(i, 5)) + "," + CStr(elem_range_16(i, 6)) + "," _

            + CStr(elem_range_16(i, 7)) + "," + CStr(elem_range_16(i, 8)) + ","

      Print #1, "+," + CStr(elem_range_16(i + 1, 1)) + "," + CStr(elem_range_16(i + 1, 2))

   End If

Next i

'FABRIC 0_90 PROPERTIES HERE

Print #1, "$"

Print #1, "$ Elements and Element Properties for region : prop_fill_0_90"

Print #1, "PSOLID,10,3,0"

Print #1, "$ Pset: ' prop_fill_0_90' will be imported as: ' psolid.5"'

Print #1, "$"

num_elem_10 = 384

For i = 1 To num_elem_10 Step 2


   If i < 0 Then 'CHECKS FOR FAILURE IN FILL elem_range_4+-45_PLY MATERIAL

    Else

      Print #1, "CHEXA," + CStr(elem_range_5(i, 1)) + "," + CStr(elem_range_5(i, 2)) + "," + CStr(elem_range_5(i, 3)) + "," _

            + CStr(elem_range_5(i, 4)) + "," + CStr(elem_range_5(i, 5)) + "," + CStr(elem_range_5(i, 6)) + "," _

            + CStr(elem_range_5(i, 7)) + "," + CStr(elem_range_5(i, 8)) + ","

      Print #1, "+," + CStr(elem_range_5(i + 1, 1)) + "," + CStr(elem_range_5(i + 1, 2))
```

```
    End If
Next i
'FOAM PROPERTIES HERE
Print #1, "$"
Print #1, "$ Elements and Element Properties for region : prop_foam"
Print #1, "PSOLID,11,5,0"
Print #1, "$ Pset: ' prop_foam' will be imported as: ' psolid.5'"
Print #1, "$"
'Print #1, "$"
'Print #1, "$ Elements and Element Properties for region : prop_foam_damaged"
'Print #1, "PSOLID,1111,55,0"
'Print #1, "$ Pset: ' prop_foam_damaged' will be imported as: ' psolid.55'"
'Print #1, "$"
num_elem_11 = 48
For i = 1 To num_elem_11 Step 2

    If i < 0 Then 'CHECKS FOR FAILURE IN FOAM MATERIAL
    Else
        Print #1, "CHEXA," + CStr(elem_range_9(i, 1)) + "," + CStr(elem_range_9(i, 2)) + "," + CStr(elem_range_9(i, 3)) + "," _
            + CStr(elem_range_9(i, 4)) + "," + CStr(elem_range_9(i, 5)) + "," + CStr(elem_range_9(i, 6)) + "," _
            + CStr(elem_range_9(i, 7)) + "," + CStr(elem_range_9(i, 8)) + ","
        Print #1, "+," + CStr(elem_range_9(i + 1, 1)) + "," + CStr(elem_range_9(i + 1, 2))
    End If
Next i
'CHEXA MATRIX PROPERTIES HERE
Print #1, "$"
Print #1, "$ Elements and Element Properties for region : prop_matrix"
```

```
Print #1, "PSOLID,12,6,0"

Print #1, "$ Pset: ' prop_matrix' will be imported as: ' psolid.5'"

Print #1, "$"

'Print #1, "$"

'Print #1, "$ Elements and Element Properties for region : prop_matrix_damaged"

'Print #1, "PSOLID,1212,66,0"

'Print #1, "$ Pset: ' prop_matrix_damaged' will be imported as: ' psolid.55'"

'Print #1, "$"

num_elem_12 = 48

For i = 1 To num_elem_12 Step 2

   If i < 0 Then 'CHECKS FOR FAILURE IN MATRIX MATERIAL

   Else

      Print #1, "CHEXA," + CStr(elem_range_10(i, 1)) + "," + CStr(elem_range_10(i, 2)) + "," + CStr(elem_range_10(i, 3)) + "," _

            + CStr(elem_range_10(i, 4)) + "," + CStr(elem_range_10(i, 5)) + "," + CStr(elem_range_10(i, 6)) + "," _

            + CStr(elem_range_10(i, 7)) + "," + CStr(elem_range_10(i, 8)) + ","

      Print #1, "+," + CStr(elem_range_10(i + 1, 1)) + "," + CStr(elem_range_10(i + 1, 2))

   End If

Next i

'CPENTA MATRIX PROPERTIES HERE

For i = 49 To 60

   If i < 0 Then 'CHECKS FOR FAILURE IN MATRIX MATERIAL

   Else

      Print #1, "CPENTA," + CStr(elem_range_10(i, 1)) + "," + CStr(elem_range_10(i, 2)) + "," + CStr(elem_range_10(i, 3)) + "," _

            + CStr(elem_range_10(i, 4)) + "," + CStr(elem_range_10(i, 5)) + "," + CStr(elem_range_10(i, 6)) + "," _

            + CStr(elem_range_10(i, 7)) + "," + CStr(elem_range_10(i, 8))

   End If
```

```
Next i
'MATERIAL PROPERTIES


Print #1, "$"
Print #1, "$ Referenced Material Records"
Print #1, "$"
Print #1, "$"
Print #1, "$ Material Record : core_alumi"
Print #1, "$ Description of Material :"
Print #1, "$ BASED ON MATRIX E11 OF 140 MPa, POISSONS OF 0.3 "
'Print #1, "MAT1,1,70000,,.3"
Print #1, "MAT9,1,30,20,20, , , ,30,"
Print #1, "+,20,,,,160,"
Print #1, "+,100,,,100,,100"
Print #1, "$"
Print #1, "$"
Print #1, "$ Material Record : core_test"
Print #1, "$ Description of Material :"
Print #1, "$ BASED ON MATRIX E11 OF 90 MPa, POISSONS OF 0.3 "
'Print #1, "MAT1,2,120,,.3"
Print #1, "MAT9,2,19.3,12.8,12.8, , , ,19.3,"
Print #1, "+,12.8,,,,103,"
Print #1, "+,64.3,,,64.3,,64.3"
Print #1, "$"
Print #1, "$"
Print #1, "$ Material Record : ply_0_90"
Print #1, "$ Description of Material :"
```

```
Print #1, "MAT9,3," + CStr(mat9_0_90_deg(1, 1) * 10) + "," + CStr(mat9_0_90_deg(1, 2) * 10) + "," _
+ CStr(mat9_0_90_deg(1, 3) * 10) + ", , , ," + CStr(mat9_0_90_deg(2, 2) * 10) + ","
Print #1, "+," + CStr(mat9_0_90_deg(2, 3) * 10) + ", , , ," + CStr(mat9_0_90_deg(3, 3) * 10) + ","
Print #1, "+," + CStr(mat9_0_90_deg(4, 4) * 10) + ", , ," + CStr(mat9_0_90_deg(5, 5) * 10) + ", ," _
+ CStr(mat9_0_90_deg(6, 6) * 10)
Print #1, "$"
Print #1, "$"
Print #1, "$ Material Record : ply_-+45"
Print #1, "$ Description of Material :"
Print #1, "MAT9,4," + CStr(mat9_45_deg(1, 1) * 10) + "," + CStr(mat9_45_deg(1, 2) * 10) + "," _
+ CStr(mat9_45_deg(1, 3) * 10) + ", , , ," + CStr(mat9_45_deg(2, 2) * 10) + ","
Print #1, "+," + CStr(mat9_45_deg(2, 3) * 10) + ", , , ," + CStr(mat9_45_deg(3, 3) * 10) + ","
Print #1, "+," + CStr(mat9_45_deg(4, 4) * 10) + ", , ," + CStr(mat9_45_deg(5, 5) * 10) + ", ," _
+ CStr(mat9_45_deg(6, 6) * 10)
Print #1, "$"
Print #1, "$"
Print #1, "$ Material Record : Matrix"
Print #1, "$ Description of Material :"
Print #1, "$ BASED ON MATRIX E11 OF 3500 MPa, POISSONS OF 0.35 "
Print #1, "MAT9,6,553.5,297.9,297.9, , , ,553.5,"
Print #1, "+,297.9,,,,553.5,"
Print #1, "+,130.0,,,130.0,,130.0"
Print #1, "$"
Print #1, "$"
Print #1, "$ Material Record : foam"
Print #1, "$ Description of Material :"
```

```
Print #1, "MAT1,5,140,,.3"

'Print #1, "MAT9,2,G11,,, , , ,G22,"

'Print #1, "+,,,,,G33.,"

'Print #1, "+,G44,,,G55,,G66"


'DAMAGED MATERIALS

Set fail_mat_sht = Worksheets("MESO_CONST_TANG")

With fail_mat_sht

Print #1, "$ DAMAGED 0_90 PLY PROPERTIES"

Print #1, "MAT9,33," + CStr((.Cells(37, 14))) + ",,, , , ," + CStr((.Cells(38, 15))) + " ,"

Print #1, "+,,,,," + CStr((.Cells(39, 16))) + ","

Print #1, "+," + CStr((.Cells(40, 17))) + ",,," + CStr((.Cells(41, 18))) + ",," + CStr((.Cells(42, 19)))

End With

'INSERT NODES

Print #1, "$ Nodes of the Entire Model"

num_nodes = 7730

For i = 1 To num_nodes

   Print #1, "GRID" + "," + CStr(node_range(i, 1)) + ",," + CStr(node_range(i, 2)) + "," + CStr(node_range(i, 3)) + "," + _

            CStr(node_range(i, 4))

Next i

'ADD LOADS AND BOUNDARY CONDITIONS

Print #1, "$ Loads for Load Case : Default"

Print #1, "SPCADD,2,1,3,4"

Print #1, "LOAD,2,1,1,1,1,3,1,4"

Print #1, "+,1,5"

Print #1, "$ Displacement Constraints of Load Set : left_1_3"

Print #1, "SPC1,1,13,12631,12632,12633,12634,12635"

Print #1, "$ Displacement Constraints of Load Set : left_2"
```

```
Print #1, "SPC1,3,2,6486,6487,6488,6489,6490,8626"

Print #1, "+,8627,8628,8629,8630"

Print #1, "$ Displacement Constraints of Load Set : right_3"

Print #1, "SPC1,4,3,11031,11032,11033,11034,11035"

Print #1, "$"

Print #1, "$ Nodal Forces of Load Set : centre_left_force"

Print #1, "FORCE    3       5641    0       " & CStr(load_sht.Cells(6 + loop_incr, 4)) & "      0.      0
-1."

Print #1, "FORCE    3       5682    0       " & CStr(load_sht.Cells(6 + loop_incr, 4)) & "      0.      0
-1."

Print #1, "FORCE    3       5723    0       " & CStr(load_sht.Cells(6 + loop_incr, 4)) & "      0.      0
-1."

Print #1, "FORCE    3       5764    0       " & CStr(load_sht.Cells(6 + loop_incr, 4)) & "      0.      0
-1."

Print #1, "FORCE    3       5805    0       " & CStr(load_sht.Cells(6 + loop_incr, 4)) & "      0.      0
-1."

Print #1, "$ Nodal Forces of Load Set : left_force"

Print #1, "FORCE    1       5640    0       " & CStr(load_sht.Cells(6 + loop_incr, 5)) & "      0.      0
-1."

Print #1, "FORCE    1       5642    0       " & CStr(load_sht.Cells(6 + loop_incr, 5)) & "      0.      0
-1."

Print #1, "FORCE    1       5681    0       " & CStr(load_sht.Cells(6 + loop_incr, 5)) & "      0.      0
-1."

Print #1, "FORCE    1       5683    0       " & CStr(load_sht.Cells(6 + loop_incr, 5)) & "      0.      0
-1."

Print #1, "FORCE    1       5722    0       " & CStr(load_sht.Cells(6 + loop_incr, 5)) & "      0.      0
-1."

Print #1, "FORCE    1       5724    0       " & CStr(load_sht.Cells(6 + loop_incr, 5)) & "      0.      0
-1."

Print #1, "FORCE    1       5763    0       " & CStr(load_sht.Cells(6 + loop_incr, 5)) & "      0.      0
-1."

Print #1, "FORCE    1       5765    0       " & CStr(load_sht.Cells(6 + loop_incr, 5)) & "      0.      0
-1."
```

```
Print #1, "FORCE    1       5804    0       " & CStr(load_sht.Cells(6 + loop_incr, 5)) & "       0.      0
-1."

Print #1, "FORCE    1       5806    0       " & CStr(load_sht.Cells(6 + loop_incr, 5)) & "       0.      0
-1."

Print #1, "$ Nodal Forces of Load Set : right_force"

Print #1, "FORCE    4       6280    0       " & CStr(load_sht.Cells(6 + loop_incr, 5)) & "       0.      0
-1."

Print #1, "FORCE    4       6282    0       " & CStr(load_sht.Cells(6 + loop_incr, 5)) & "       0.      0
-1."

Print #1, "FORCE    4       6321    0       " & CStr(load_sht.Cells(6 + loop_incr, 5)) & "       0.      0
-1."

Print #1, "FORCE    4       6323    0       " & CStr(load_sht.Cells(6 + loop_incr, 5)) & "       0.      0
-1."

Print #1, "FORCE    4       6362    0       " & CStr(load_sht.Cells(6 + loop_incr, 5)) & "       0.      0
-1."

Print #1, "FORCE    4       6364    0       " & CStr(load_sht.Cells(6 + loop_incr, 5)) & "       0.      0
-1."

Print #1, "FORCE    4       6403    0       " & CStr(load_sht.Cells(6 + loop_incr, 5)) & "       0.      0
-1."

Print #1, "FORCE    4       6405    0       " & CStr(load_sht.Cells(6 + loop_incr, 5)) & "       0.      0
-1."

Print #1, "FORCE    4       6444    0       " & CStr(load_sht.Cells(6 + loop_incr, 5)) & "       0.      0
-1."

Print #1, "FORCE    4       6446    0       " & CStr(load_sht.Cells(6 + loop_incr, 5)) & "       0.      0
-1."

Print #1, "$ Nodal Forces of Load Set : centre_right_force"

Print #1, "FORCE    5       6281    0       " & CStr(load_sht.Cells(6 + loop_incr, 4)) & "       0.      0
-1."

Print #1, "FORCE    5       6322    0       " & CStr(load_sht.Cells(6 + loop_incr, 4)) & "       0.      0
-1."

Print #1, "FORCE    5       6363    0       " & CStr(load_sht.Cells(6 + loop_incr, 4)) & "       0.      0
-1."

Print #1, "FORCE    5       6404    0       " & CStr(load_sht.Cells(6 + loop_incr, 4)) & "       0.      0
-1."
```

```
Print #1, "FORCE    5       6445    0       " & CStr(load_sht.Cells(6 + loop_incr, 4)) & "      0.      0
-1."

Print #1, "ENDDATA 07b00524"

Close #1

'PASTE WARPTOW ELASTIC CONSTANT VALUES

Dim E_sht, repair_crt_sht As Worksheet

Set E_sht = Worksheets("MESO_CONST_TANG")

Set repair_crt_sht = Worksheets("REPAIR_CRIT_STR")

repair_crt_sht.Cells(8 + loop_incr, 28) = E_sht.Cells(14, 22) 'E1

repair_crt_sht.Cells(8 + loop_incr, 29) = E_sht.Cells(15, 22) 'E2

repair_crt_sht.Cells(8 + loop_incr, 30) = E_sht.Cells(16, 22) 'E3


Next loop_incr


End Sub
```

10. VBA Script repair_critical_strs ()

```
Option Explicit

Option Base 1

Dim loop_incr, meso_lim_loop, micro_lim_loop As Integer

Sub repair_critical_strs(loop_incr)

Dim elem_num As Integer

Dim disp_max_y, i As Double

Dim file1 As String

Dim stressbook As Workbooks

Dim stresssht, resultsht As Worksheet

Dim str_tensor(3, 3) As Variant

Dim crit_str_range As Range

Set resultsht = Worksheets("REPAIR_CRIT_STR")

'loop_incr = 1

file1 = "C:\MsFEM_Project\GRAND_MASTER\REPAIR_INCR_" & CStr(loop_incr) & "\repair_" & CStr(loop_incr) & ".f06"

Workbooks.OpenText FileName:= _

    file1 _

    , Origin:=437, StartRow:=1, DataType:=xlDelimited, TextQualifier:= _

    xlDoubleQuote, ConsecutiveDelimiter:=True, Tab:=True, Semicolon:=False, _

    Comma:=False, Space:=True, Other:=False, FieldInfo:=Array(Array(1, 1), _

    Array(2, 1), Array(3, 1), Array(4, 1), Array(5, 1), Array(6, 1), Array(7, 1), Array(8, 1), _

    Array(9, 1), Array(10, 1), Array(11, 1), Array(12, 1), Array(13, 1)), _

    TrailingMinusNumbers:=True

Set stresssht = Workbooks("repair_" & CStr(loop_incr)).Worksheets("repair_" & CStr(loop_incr))

For i = 1 To 50000

'get critical element strains

  If stresssht.Cells(i, 2) = 11101 Then
```

```
    If stresssht.Cells(1 + i, 2) = "MSC.NASTRAN" Then


    str_tensor(1, 1) = stresssht.Cells(i + 9, 4) 'e11
    str_tensor(1, 2) = stresssht.Cells(i + 9, 6) 'e12
    str_tensor(2, 1) = stresssht.Cells(i + 9, 6) 'e12
    str_tensor(2, 2) = stresssht.Cells(i + 10, 3) 'e22
    str_tensor(2, 3) = stresssht.Cells(i + 10, 5) 'e23
    str_tensor(3, 2) = stresssht.Cells(i + 10, 5) 'e32
    str_tensor(3, 3) = stresssht.Cells(i + 11, 3) 'e33
    str_tensor(3, 1) = stresssht.Cells(i + 11, 5) 'e31
    str_tensor(1, 3) = stresssht.Cells(i + 11, 5) 'e13
    Else
    'MsgBox ("Maximum stress !!!")
    str_tensor(1, 1) = stresssht.Cells(i + 1, 4) 'e11
    str_tensor(1, 2) = stresssht.Cells(i + 1, 6) 'e12
    str_tensor(2, 1) = stresssht.Cells(i + 1, 6) 'e12
    str_tensor(2, 2) = stresssht.Cells(i + 2, 3) 'e22
    str_tensor(2, 3) = stresssht.Cells(i + 2, 5) 'e23
    str_tensor(3, 2) = stresssht.Cells(i + 2, 5) 'e32
    str_tensor(3, 3) = stresssht.Cells(i + 3, 3) 'e33
    str_tensor(3, 1) = stresssht.Cells(i + 3, 5) 'e31
    str_tensor(1, 3) = stresssht.Cells(i + 3, 5) 'e13
    End If
    End If
'get maximum displacements
    If Not IsError(stresssht.Cells(i, 3)) Then
    If stresssht.Cells(i, 3) = "DISPLACEMENTS" Then
    'MsgBox ("Maximum displacement !!!")
```

```
    resultsht.Cells(6 + loop_incr, 7) = stresssht.Cells(3 + i, 4)

    resultsht.Cells(6 + loop_incr, 6) = loop_incr

    End If

    End If

Next i

'MATRIX_2_ELM.Resize(UBound(MATRIX_2, 1), 1).Value = MATRIX_2

'Set MATRIX_2_ELM = Range(target_sht.Cells(60000, 109 + j_L * 18), target_sht.Cells(60000, 109 + j_L * 18))

Set crit_str_range = Range(resultsht.Cells(loop_incr * 4 + 3, 11), resultsht.Cells(loop_incr * 4 + 5, 13))

crit_str_range.Resize(3, 3).Value = str_tensor

Workbooks("repair_" & CStr(loop_incr) & ".f06").Close SaveChanges:=False

End Sub
```

11. VBA Script meso_crit_strain ()

```
Option Explicit
Option Base 1
Dim loop_incr, meso_lim_loop, micro_lim_loop As Integer
Sub meso_crit_strain(loop_incr)
Dim elem_num As Integer
Dim disp_max_y, i As Double
Dim file1 As String
Dim stressbook As Workbooks
Dim stresssht, resultsht As Worksheet
Dim str_tensor(3, 3) As Variant
Dim crit_str_range As Range
Set resultsht = Worksheets("MESO_CRIT_STR")
'loop_incr = 1
file1 = "C:\MsFEM_Project\GRAND_MASTER\REPAIR_INCR_" & CStr(loop_incr) & "\MESO_CRIT_" & CStr(loop_incr) & "\meso_rve_55.f06"
Workbooks.OpenText FileName:= _
    file1 _
    , Origin:=437, StartRow:=1, DataType:=xlDelimited, TextQualifier:= _
    xlDoubleQuote, ConsecutiveDelimiter:=True, Tab:=True, Semicolon:=False, _
    Comma:=False, Space:=True, Other:=False, FieldInfo:=Array(Array(1, 1), _
    Array(2, 1), Array(3, 1), Array(4, 1), Array(5, 1), Array(6, 1), Array(7, 1), Array(8, 1), _
    Array(9, 1), Array(10, 1), Array(11, 1), Array(12, 1), Array(13, 1)), _
    TrailingMinusNumbers:=True
Set stresssht = Workbooks("meso_rve_55").Worksheets("meso_rve_55")
For i = 1 To 4550
'get critical element strains
```

```
    If stresssht.Cells(i, 2) = 25650 Then
    If stresssht.Cells(1 + i, 2) = "MSC.NASTRAN" Then
    str_tensor(1, 1) = stresssht.Cells(i + 9, 4) 'e11
    str_tensor(1, 2) = stresssht.Cells(i + 9, 6) 'e12
    str_tensor(2, 1) = stresssht.Cells(i + 9, 6) 'e12
    str_tensor(2, 2) = stresssht.Cells(i + 10, 3) 'e22
    str_tensor(2, 3) = stresssht.Cells(i + 10, 5) 'e23
    str_tensor(3, 2) = stresssht.Cells(i + 10, 5) 'e32
    str_tensor(3, 3) = stresssht.Cells(i + 11, 3) 'e33
    str_tensor(3, 1) = stresssht.Cells(i + 11, 5) 'e31
    str_tensor(1, 3) = stresssht.Cells(i + 11, 5) 'e13
    Else
    'MsgBox ("Maximum stress !!!")
    str_tensor(1, 1) = stresssht.Cells(i + 1, 4) 'e11
    str_tensor(1, 2) = stresssht.Cells(i + 1, 6) 'e12
    str_tensor(2, 1) = stresssht.Cells(i + 1, 6) 'e12
    str_tensor(2, 2) = stresssht.Cells(i + 2, 3) 'e22
    str_tensor(2, 3) = stresssht.Cells(i + 2, 5) 'e23
    str_tensor(3, 2) = stresssht.Cells(i + 2, 5) 'e32
    str_tensor(3, 3) = stresssht.Cells(i + 3, 3) 'e33
    str_tensor(3, 1) = stresssht.Cells(i + 3, 5) 'e31
    str_tensor(1, 3) = stresssht.Cells(i + 3, 5) 'e13
    End If
    End If
Next i
'MATRIX_2_ELM.Resize(UBound(MATRIX_2, 1), 1).Value = MATRIX_2
'Set MATRIX_2_ELM = Range(target_sht.Cells(60000, 109 + j_L * 18), target_sht.Cells(60000,
109 + j_L * 18))
```

```
Set crit_str_range = Range(resultsht.Cells(5 + 4 * loop_incr, 4), resultsht.Cells(7 + 4 * loop_incr, 7))

crit_str_range.Resize(3, 3).Value = str_tensor

Workbooks("meso_rve_55.f06").Close SaveChanges:=False

End Sub
```

12. VBA Script meso_check_mat_update ()

```
Option Explicit
Dim loop_incr, meso_lim_loop, micro_lim_loop As Integer
Option Base 1
Sub meso_check_mat_update(loop_incr)
'READ F06 FILE
Dim norm_x, norm_y, norm_z, shr_xy, shr_yz, shr_zx, von_mises As Double
Dim fNameAndPath As Variant
Dim TextLine, fName
Dim myCount, counter, a_elm, a_11, a_22, a_33, a_44, A_ii As Long
Dim counterp, a_elmp, a_11p, a_22p, a_33p, a_44p As Long
Dim range_row(1, 45000) As Variant
Dim stress_matrix(45000, 8) As Variant
Dim r As Integer
'*********************************************************************
Dim volumebook, stressbook, resultbook As Workbook
Dim volumesht, E_sht, stresssht, resultsht, meso_crt_sht, meso_elem_prop_sht, micro_crt_sht As Worksheet
Dim norm_x_min, norm_y_min, norm_z_min, shr_xy_min, shr_yz_min, shr_zx_min As Double
Dim norm_x_max, norm_y_max, norm_z_max, shr_xy_max, shr_yz_max, shr_zx_max, elem_vm_row As Double
Dim numelem, elem, i, j, k, a, b, c, ss, ff As Integer
Dim elem_x_tens, elem_y_comp, elem_xy, elem_vm As Integer
Dim xx_str, yy_str, zz_str, xy_str, yz_str, zx_str As Double
Dim Fn As Object
Set Fn = Application.WorksheetFunction
Application.ScreenUpdating = False
```

```
Workbooks("GRAND_MASTER_1.xlsm").Activate
Set resultsht = Worksheets("REPAIR_CRIT_STR")
Set meso_crt_sht = Worksheets("MESO_CRIT_STR")
Set meso_elem_prop_sht = Worksheets("MESO_RVE_ELEM_PROPS")
Set E_sht = Worksheets("MESO_CONST_TANG")
Set micro_crt_sht = Worksheets("MICRO_CRIT_STR")
E_sht.Cells(14, 24) = E_sht.Cells(14, 22)
j = 55
'loop_incr = 1
meso_lim_loop = 1
'*****************************************************************
numelem = 45000
'loop_incr = 1
    myCount = 1
    counter = 1
    counterp = 1
    a_elm = 8
    a_11 = 9
    a_22 = 11
    a_33 = 12
    a_44 = 14
    a_elmp = 2240015
    a_11p = 2240016
    a_22p = 2240018
    a_33p = 2240019
    a_44p = 2240021
    numelem = 45000
    'working input folder location
```

```
fNameAndPath = "C:\MsFEM_Project\GRAND_MASTER\REPAIR_INCR_" & CStr(loop_incr) & "\MESO_CRIT_" & CStr(loop_incr) & _

"\MESO_RVE_" & CStr(j) & ".pch"

Open fNameAndPath For Input As #1    ' Open file.

Do While Not EOF(1)    ' Loop until end of file.

    Line Input #1, TextLine    ' Read line into variable.

    'HEXA ELEMENT RESULTS

    A_ii = a_elm + (counter - 1) * 64

    If A_ii = myCount And Left(TextLine, 3) = "   " Then

        'Print #2, TextLine & " ELEMENT NO " & CStr(myCount) & "   " & CStr(counter) & "   " & Mid(TextLine, 6, 5) ' write variable to output file.

        'counter = counter + 1

        'elem

            stress_matrix(counter, 1) = CLng((Mid(TextLine, 6, 5)))

    End If

    A_ii = a_11 + (counter - 1) * 64

    If A_ii = myCount And Left(TextLine, 6) = "-CONT-" Then

        'Print #2, TextLine & " X and XY  " & CStr(myCount) & "   " & CStr(counter) & " X " & Mid(TextLine, 42, 13) & " XY " & Mid(TextLine, 60, 13) ' write variable to output file.

        'counter = counter + 1

        'norm_x

            stress_matrix(counter, 2) = CSng(Mid(TextLine, 42, 13))

        'shr_xy

            stress_matrix(counter, 5) = CSng(Mid(TextLine, 60, 13))

    End If

    A_ii = a_22 + (counter - 1) * 64

    If A_ii = myCount And Left(TextLine, 6) = "-CONT-" Then

        'Print #2, TextLine & " VM  " & CStr(myCount) & " VM " & Mid(TextLine, 60, 13) ' write variable to output file.

        'counter = counter + 1
```

```
        'von_mises

            stress_matrix(counter, 8) = CSng(Mid(TextLine, 60, 13))

        End If

        A_ii = a_33 + (counter - 1) * 64

        If A_ii = myCount And Left(TextLine, 6) = "-CONT-" Then

            'Print #2, TextLine & " Y and YZ  " & CStr(myCount) & "    " & CStr(counter) & " Y " & Mid(TextLine, 24, 13) & " YZ " & Mid(TextLine, 42, 13) ' write variable to output file.

            'counter = counter + 1

            'norm_y

                stress_matrix(counter, 3) = CSng(Mid(TextLine, 24, 13))

            'shr_yz

                stress_matrix(counter, 6) = CSng(Mid(TextLine, 42, 13))

        End If

        A_ii = a_44 + (counter - 1) * 64

        If A_ii = myCount And Left(TextLine, 6) = "-CONT-" Then

            'Print #2, TextLine & " Z and ZX  " & CStr(myCount) & "    " & CStr(counter) & " Z " & Mid(TextLine, 24, 13) & " ZX " & Mid(TextLine, 42, 13) ' write variable to output file.

            'norm_z

                stress_matrix(counter, 4) = CSng(Mid(TextLine, 24, 13))

            'shr_zx

                stress_matrix(counter, 7) = CSng(Mid(TextLine, 42, 13))

            counter = counter + 1

        End If

        'PENTA ELEMENT RESULTS

        A_ii = a_elmp + (counterp - 1) * 50

        If A_ii = myCount And Left(TextLine, 3) = "   " Then

            'Print #2, TextLine & " ELEMENT NO " & CStr(myCount) & "    " & CStr(counterp + counter - 1) & "   " & Mid(TextLine, 6, 5) ' write variable to output file.

            'counter = counter + 1
```

```
        'elem
        stress_matrix(counterp + counter - 1, 1) = CLng((Mid(TextLine, 6, 5)))
    End If
    A_ii = a_11p + (counterp - 1) * 50
    If A_ii = myCount And Left(TextLine, 6) = "-CONT-" Then
        'Print #2, TextLine & " X and XY  " & CStr(myCount) & "   " & CStr(counterp + counter - 1) & " X " & Mid(TextLine, 42, 13) & " XY " & Mid(TextLine, 60, 13) ' write variable to output file.
        'counter = counter + 1
        'norm_x
            stress_matrix(counterp + counter - 1, 2) = CSng(Mid(TextLine, 42, 13))
        'shr_xy
            stress_matrix(counterp + counter - 1, 5) = CSng(Mid(TextLine, 60, 13))
    End If
    A_ii = a_22p + (counterp - 1) * 50
    If A_ii = myCount And Left(TextLine, 6) = "-CONT-" Then
        'Print #2, TextLine & " VM  " & CStr(myCount) & " VM " & Mid(TextLine, 60, 13)  ' write variable to output file.
        'counter = counter + 1
        'von_mises
            stress_matrix(counter + counterp - 1, 8) = CSng(Mid(TextLine, 60, 13))
    End If
    A_ii = a_33p + (counterp - 1) * 50
    If A_ii = myCount And Left(TextLine, 6) = "-CONT-" Then
        'Print #2, TextLine & " Y and YZ  " & CStr(myCount) & "   " & CStr(counterp + counter - 1) & " Y " & Mid(TextLine, 24, 13) & " YZ " & Mid(TextLine, 42, 13) ' write variable to output file.
        'counter = counter + 1
        'norm_y
            stress_matrix(counter + counterp - 1, 3) = CSng(Mid(TextLine, 24, 13))
```

```
            'shr_yz
                stress_matrix(counter + counterp - 1, 6) = CSng(Mid(TextLine, 42, 13))
        End If
        A_ii = a_44p + (counterp - 1) * 50
        If A_ii = myCount And Left(TextLine, 6) = "-CONT-" Then
            'Print #2, TextLine & " Z and ZX  " & CStr(myCount) & "    " & CStr(counterp + counter - 1) & " Z " & Mid(TextLine, 24, 13) & " ZX " & Mid(TextLine, 42, 13) ' write variable to output file.
            'norm_z
                stress_matrix(counter + counterp - 1, 4) = CSng(Mid(TextLine, 24, 13))
            'shr_zx
                stress_matrix(counter + counterp - 1, 7) = CSng(Mid(TextLine, 42, 13))
            counterp = counterp + 1
        End If
        myCount = myCount + 1
    Loop
    'Close #2
    Close #1    ' Close file.
'EXTRACT CRITICAL ELEMENTS AND THEIR STRESSES
    'COLLECT MAXIMUM VON MISES STRESS
Dim vm_MATRIX_1(), vm_MATRIX_2(), vm_WEFTTOW() As Variant
Dim vm_WEFTTOW_RAMP_1(), vm_WEFTTOW_RAMP_2(), vm_WARPTOW(), vm_WARPTOW_RAMP_1(), vm_WARPTOW_RAMP_2() As Variant
Dim dest_MATRIX_1, dest_MATRIX_2, dest_WEFTTOW, dest_WEFTTOW_RAMP_1, dest_WEFTTOW_RAMP_2, dest_ As Range
Dim dest_WARPTOW, dest_WARPTOW_RAMP_1, dest_WARPTOW_RAMP_2 As Range
Dim MATRIX_1_elem(), MATRIX_2_elem(), WEFTTOW_elem(), WEFTTOW_RAMP_1_elem(), WEFTTOW_RAMP_2_elem() As Variant
Dim WARPTOW_elem(), WARPTOW_RAMP_1_elem(), WARPTOW_RAMP_2_elem() As Variant
```

```
Dim Arr_MATRIX_1(), Arr_MATRIX_2(), Arr_WEFTTOW(), Arr_WEFTTOW_RAMP_1(), Arr_WEFTTOW_RAMP_2() As Variant

Dim Arr_WARPTOW(), Arr_WARPTOW_RAMP_1(), Arr_WARPTOW_RAMP_2() As Variant

Set dest_MATRIX_1 = Range(meso_elem_prop_sht.Cells(3, 111 + loop_incr * 12), meso_elem_prop_sht.Cells(3, 111 + loop_incr * 12))

Set dest_MATRIX_2 = Range(meso_elem_prop_sht.Cells(3, 112 + loop_incr * 12), meso_elem_prop_sht.Cells(3, 112 + loop_incr * 12))

Set dest_WEFTTOW = Range(meso_elem_prop_sht.Cells(3, 113 + loop_incr * 12), meso_elem_prop_sht.Cells(3, 113 + loop_incr * 12))

Set dest_WEFTTOW_RAMP_1 = Range(meso_elem_prop_sht.Cells(3, 114 + loop_incr * 12), meso_elem_prop_sht.Cells(3, 114 + loop_incr * 12))

Set dest_WEFTTOW_RAMP_2 = Range(meso_elem_prop_sht.Cells(3, 115 + loop_incr * 12), meso_elem_prop_sht.Cells(3, 115 + loop_incr * 12))

Set dest_WARPTOW = Range(meso_elem_prop_sht.Cells(3, 116 + loop_incr * 12), meso_elem_prop_sht.Cells(3, 116 + loop_incr * 12))

Set dest_WARPTOW_RAMP_1 = Range(meso_elem_prop_sht.Cells(3, 117 + loop_incr * 12), meso_elem_prop_sht.Cells(3, 117 + loop_incr * 12))

Set dest_WARPTOW_RAMP_2 = Range(meso_elem_prop_sht.Cells(3, 118 + loop_incr * 12), meso_elem_prop_sht.Cells(3, 118 + loop_incr * 12))

'IDENTIFY FAILED MATRIX_1,MATRIX_2 ,WEFTTOW,WEFTTOW_RAMP_1 ,WEFTTOW_RAMP_2 ,WARPTOW,WARPTOW_RAMP_1 ,WARPTOW_RAMP_2 ELEMENTS

   'CHECK MATRIX_1 ELEMENT FAILURE

Range(meso_elem_prop_sht.Cells(3, 123 + loop_incr * 12), meso_elem_prop_sht.Cells(50000, 123 + loop_incr * 12)).ClearContents

MATRIX_1_elem = Range(meso_elem_prop_sht.Cells(3, 93), meso_elem_prop_sht.Cells(10002, 93))

   'REDIMENSION ARRAYS TO UBOUND

ReDim vm_MATRIX_1(UBound(MATRIX_1_elem), 1)

vm_MATRIX_1 = Application.Index(stress_matrix, MATRIX_1_elem, 8)

dest_MATRIX_1.Resize(UBound(vm_MATRIX_1, 1), 1).Value = vm_MATRIX_1 'paste values

   'CHECK MATRIX_2 ELEMENT FAILURE

Range(meso_elem_prop_sht.Cells(3, 124 + loop_incr * 12), meso_elem_prop_sht.Cells(50000, 124 + loop_incr * 12)).ClearContents
```

```
MATRIX_2_elem = Range(meso_elem_prop_sht.Cells(3, 97), meso_elem_prop_sht.Cells(25002, 97))

   'REDIMENSION ARRAYS TO UBOUND

ReDim vm_MATRIX_2(UBound(MATRIX_2_elem), 1)

vm_MATRIX_2 = Application.Index(stress_matrix, MATRIX_2_elem, 8)

dest_MATRIX_2.Resize(UBound(vm_MATRIX_2, 1), 1).Value = vm_MATRIX_2 'paste values

   'CHECK WEFTTOW ELEMENT FAILURE

Range(meso_elem_prop_sht.Cells(3, 125 + loop_incr * 12), meso_elem_prop_sht.Cells(50000, 125 + loop_incr * 12)).ClearContents

WEFTTOW_elem = Range(meso_elem_prop_sht.Cells(3, 101), meso_elem_prop_sht.Cells(2002, 101))

   'REDIMENSION ARRAYS TO UBOUND

ReDim vm_WEFTTOW(UBound(WEFTTOW_elem), 1)

vm_WEFTTOW = Application.Index(stress_matrix, WEFTTOW_elem, 8)

dest_WEFTTOW.Resize(UBound(vm_WEFTTOW, 1), 1).Value = vm_WEFTTOW 'paste values

   'CHECK WEFTTOW_RAMP_1 ELEMENT FAILURE

Range(meso_elem_prop_sht.Cells(3, 126 + loop_incr * 12), meso_elem_prop_sht.Cells(50000, 126 + loop_incr * 12)).ClearContents

WEFTTOW_RAMP_1_elem = Range(meso_elem_prop_sht.Cells(3, 105), meso_elem_prop_sht.Cells(1502, 105))

   'REDIMENSION ARRAYS TO UBOUND

ReDim vm_WEFTTOW_RAMP_1(UBound(WEFTTOW_RAMP_1_elem), 1)

vm_WEFTTOW_RAMP_1 = Application.Index(stress_matrix, WEFTTOW_RAMP_1_elem, 8)

dest_WEFTTOW_RAMP_1.Resize(UBound(vm_WEFTTOW_RAMP_1, 1), 1).Value = vm_WEFTTOW_RAMP_1 'paste values

   'CHECK WEFTTOW_RAMP_2 ELEMENT FAILURE


Range(meso_elem_prop_sht.Cells(3, 127 + loop_incr * 12), meso_elem_prop_sht.Cells(50000, 127 + loop_incr * 12)).ClearContents

WEFTTOW_RAMP_2_elem = Range(meso_elem_prop_sht.Cells(3, 109), meso_elem_prop_sht.Cells(1502, 109))
```

```
    'REDIMENSION ARRAYS TO UBOUND

ReDim vm_WEFTTOW_RAMP_2(UBound(WEFTTOW_RAMP_2_elem), 1)

vm_WEFTTOW_RAMP_2 = Application.Index(stress_matrix, WEFTTOW_RAMP_2_elem, 8)

dest_WEFTTOW_RAMP_2.Resize(UBound(vm_WEFTTOW_RAMP_2, 1), 1).Value = vm_WEFTTOW_RAMP_2 'paste values

    'CHECK WARPTOW ELEMENT FAILURE

Range(meso_elem_prop_sht.Cells(3, 128 + loop_incr * 12), meso_elem_prop_sht.Cells(50000, 128 + loop_incr * 12)).ClearContents

WARPTOW_elem = Range(meso_elem_prop_sht.Cells(3, 113), meso_elem_prop_sht.Cells(2002, 113))

    'REDIMENSION ARRAYS TO UBOUND

ReDim vm_WARPTOW(UBound(WARPTOW_elem), 1)

vm_WARPTOW = Application.Index(stress_matrix, WARPTOW_elem, 8)

dest_WARPTOW.Resize(UBound(vm_WARPTOW, 1), 1).Value = vm_WARPTOW 'paste values

    'CHECK WARPTOW_RAMP_1 ELEMENT FAILURE

Range(meso_elem_prop_sht.Cells(3, 129 + loop_incr * 12), meso_elem_prop_sht.Cells(50000, 129 + loop_incr * 12)).ClearContents

WARPTOW_RAMP_1_elem = Range(meso_elem_prop_sht.Cells(3, 117), meso_elem_prop_sht.Cells(1502, 117))

    'REDIMENSION ARRAYS TO UBOUND

ReDim vm_WARPTOW_RAMP_1(UBound(WARPTOW_RAMP_1_elem), 1)

vm_WARPTOW_RAMP_1 = Application.Index(stress_matrix, WARPTOW_RAMP_1_elem, 8)

dest_WARPTOW_RAMP_1.Resize(UBound(vm_WARPTOW_RAMP_1, 1), 1).Value = vm_WARPTOW_RAMP_1 'paste values

    'CHECK WARPTOW_RAMP_2 ELEMENT FAILURE

Range(meso_elem_prop_sht.Cells(3, 130 + loop_incr * 12), meso_elem_prop_sht.Cells(50000, 130 + loop_incr * 12)).ClearContents

WARPTOW_RAMP_2_elem = Range(meso_elem_prop_sht.Cells(3, 121), meso_elem_prop_sht.Cells(1502, 121))

    'REDIMENSION ARRAYS TO UBOUND

ReDim vm_WARPTOW_RAMP_2(UBound(WARPTOW_RAMP_2_elem), 1)
```

```
vm_WARPTOW_RAMP_2 = Application.Index(stress_matrix, WARPTOW_RAMP_2_elem, 8)

dest_WARPTOW_RAMP_2.Resize(UBound(vm_WARPTOW_RAMP_2, 1), 1).Value = vm_WARPTOW_RAMP_2 'paste values

  'IDENTIFY FAILED MATRIX_1,MATRIX_2 ,WEFTTOW,WEFTTOW_RAMP_1 ,WEFTTOW_RAMP_2 ,WARPTOW,WARPTOW_RAMP_1 ,WARPTOW_RAMP_2 ELEMENTS

If meso_lim_loop = 1 Then

   Range(meso_elem_prop_sht.Cells(59999 + ff, 108 + loop_incr * 18), _

   meso_elem_prop_sht.Cells(100000, 108 + (loop_incr) * 18)).ClearContents

   Range(meso_elem_prop_sht.Cells(59999 + ff, 110 + loop_incr * 18), _

   meso_elem_prop_sht.Cells(100000, 110 + loop_incr * 18)).ClearContents

   Range(meso_elem_prop_sht.Cells(59999 + ff, 112 + loop_incr * 18), _

   meso_elem_prop_sht.Cells(100000, 112 + (loop_incr) * 18)).ClearContents

   Range(meso_elem_prop_sht.Cells(59999 + ff, 114 + loop_incr * 18), _

   meso_elem_prop_sht.Cells(100000, 114 + loop_incr * 18)).ClearContents

   Range(meso_elem_prop_sht.Cells(59999 + ff, 116 + loop_incr * 18), _

   meso_elem_prop_sht.Cells(100000, 116 + (loop_incr) * 18)).ClearContents

   Range(meso_elem_prop_sht.Cells(59999 + ff, 118 + loop_incr * 18), _

   meso_elem_prop_sht.Cells(100000, 118 + loop_incr * 18)).ClearContents

   Range(meso_elem_prop_sht.Cells(59999 + ff, 120 + loop_incr * 18), _

   meso_elem_prop_sht.Cells(100000, 120 + (loop_incr) * 18)).ClearContents

   Range(meso_elem_prop_sht.Cells(59999 + ff, 122 + loop_incr * 18), _

   meso_elem_prop_sht.Cells(100000, 122 + loop_incr * 18)).ClearContents

End If

For ff = 1 To 5400

   'MATRIX_1 group

   If meso_elem_prop_sht.Cells(2 + ff, 111 + loop_incr * 12) > 0.063 Or _

   meso_elem_prop_sht.Cells(59999 + ff, 108 + (loop_incr - 1) * 18) = 111 Then

      meso_elem_prop_sht.Cells(59999 + ff, 108 + loop_incr * 18) = 111
```

```
End If
'MATRIX_2 group
If meso_elem_prop_sht.Cells(2 + ff, 112 + loop_incr * 12) > 0.063 Or _
meso_elem_prop_sht.Cells(59999 + ff, 110 + (loop_incr - 1) * 18) = 222 Then
   meso_elem_prop_sht.Cells(59999 + ff, 110 + loop_incr * 18) = 222
End If
'WEFTTOW group
If meso_elem_prop_sht.Cells(2 + ff, 113 + loop_incr * 12) > 0.8 Or _
meso_elem_prop_sht.Cells(59999 + ff, 112 + (loop_incr - 1) * 18) = 333 Then
   meso_elem_prop_sht.Cells(59999 + ff, 112 + loop_incr * 18) = 333
End If
'WEFTTOW_RAMP_1 group
If meso_elem_prop_sht.Cells(2 + ff, 114 + loop_incr * 12) > 0.8 Or _
meso_elem_prop_sht.Cells(59999 + ff, 114 + (loop_incr - 1) * 18) = 444 Then
   meso_elem_prop_sht.Cells(59999 + ff, 114 + loop_incr * 18) = 444
End If
'WEFTTOW_RAMP_2 group
If meso_elem_prop_sht.Cells(2 + ff, 115 + loop_incr * 12) > 0.8 Or _
meso_elem_prop_sht.Cells(59999 + ff, 116 + (loop_incr - 1) * 18) = 555 Then
   meso_elem_prop_sht.Cells(59999 + ff, 116 + loop_incr * 18) = 555
End If
'WARPTOW group
If meso_elem_prop_sht.Cells(2 + ff, 116 + loop_incr * 12) > 0.8 Or _
meso_elem_prop_sht.Cells(59999 + ff, 118 + (loop_incr - 1) * 18) = 666 Then
   meso_elem_prop_sht.Cells(59999 + ff, 118 + loop_incr * 18) = 666
End If
'WARPTOW_RAMP_1 group
   If meso_elem_prop_sht.Cells(2 + ff, 117 + loop_incr * 12) > 0.8 Or _
```

```
    meso_elem_prop_sht.Cells(59999 + ff, 120 + (loop_incr - 1) * 18) = 777 Then
        meso_elem_prop_sht.Cells(59999 + ff, 120 + loop_incr * 18) = 777
    End If
    'WARPTOW_RAMP_2 group
    If meso_elem_prop_sht.Cells(2 + ff, 118 + loop_incr * 12) > 0.8 Or _
    meso_elem_prop_sht.Cells(59999 + ff, 122 + (loop_incr - 1) * 18) = 888 Then
        meso_elem_prop_sht.Cells(59999 + ff, 122 + loop_incr * 18) = 888
    End If
Next ff
'EXTRACT CRITICAL ELEMENTS AND THEIR STRESSES
'write results to spreadsheet
'xx_str
micro_crt_sht.Cells(25, 11) = Fn.Max(Fn.Index(stress_matrix, WARPTOW_elem, 2))
'yy_str
micro_crt_sht.Cells(26, 12) = Fn.Max(Fn.Index(stress_matrix, WARPTOW_elem, 3))
'zz_str
micro_crt_sht.Cells(27, 13) = Fn.Max(Fn.Index(stress_matrix, WARPTOW_elem, 4))
'xy_str
micro_crt_sht.Cells(25, 12) = Fn.Max(Fn.Index(stress_matrix, WARPTOW_elem, 5))
'yz_str
micro_crt_sht.Cells(26, 13) = Fn.Max(Fn.Index(stress_matrix, WARPTOW_elem, 6))
'zx_str
micro_crt_sht.Cells(25, 13) = Fn.Max(Fn.Index(stress_matrix, WARPTOW_elem, 7))
'TABULATE MATRIX_1, MATRIX_2,WEFTTOW,WEFTTOW_RAMP_1,WEFTTOW_RAMP_2,WARPTOW,WARPTOW_RAMP_1 AND WARPTOW_RAMP_2
    'MATRIX_1 vm str
    meso_crt_sht.Cells(15 + loop_incr, 16) = meso_elem_prop_sht.Cells(3, 122 + 12 * loop_incr)
    'MATRIX_2 vm str
```

```
    meso_crt_sht.Cells(15 + loop_incr, 17) = meso_elem_prop_sht.Cells(5, 122 + 12 * loop_incr)
    'WEFTTOW vm str
    meso_crt_sht.Cells(15 + loop_incr, 18) = meso_elem_prop_sht.Cells(7, 122 + 12 * loop_incr)
    'WEFTTOW_RAMP_1 vm str
    meso_crt_sht.Cells(15 + loop_incr, 19) = meso_elem_prop_sht.Cells(11, 122 + 12 * loop_incr)
    'WEFTTOW_RAMP_2 vm str
    meso_crt_sht.Cells(15 + loop_incr, 20) = meso_elem_prop_sht.Cells(13, 122 + 12 * loop_incr)
    'WARPTOW vm str
    meso_crt_sht.Cells(15 + loop_incr, 21) = meso_elem_prop_sht.Cells(14, 122 + 12 * loop_incr)
    'WARPTOW_RAMP_1 vm str
    meso_crt_sht.Cells(15 + loop_incr, 22) = meso_elem_prop_sht.Cells(15, 122 + 12 * loop_incr)
    'WARPTOW_RAMP_2 vm str
    meso_crt_sht.Cells(15 + loop_incr, 23) = meso_elem_prop_sht.Cells(17, 122 + 12 * loop_incr)
'UPDATE FAILED ELEMENTS COLUMNS
Application.ScreenUpdating = True
End Sub
```

13. VBA Script crit_micro_RunNastran ()

```
Sub crit_micro_RunNastran(loop_incr)
' Send a file with full path to this subroutine
' The command will be run in a shell window
Dim WshShell As Object
Dim ShellCmd As String
Dim FilePath As String
Dim FilenameAndPath As String
Dim FileName, f06_file, f04_file, log_file As String   'executable file name
Dim const_law_sht As Worksheet
Dim j As Integer
Application.ScreenUpdating = False
'loop_incr = 1
Set const_law_sht = Worksheets("MICRO_CONST_TANG")
For j = 55 To 55
FilenameAndPath = "C:\MsFEM_Project\GRAND_MASTER\REPAIR_INCR_" & CStr(loop_incr) & "\MESO_CRIT_" & CStr(loop_incr) & _
   "\MICRO_CRIT_" & CStr(loop_incr) & "\MICRO_RVE_" & CStr(55) & ".bdf"
FileName = Dir(FilenameAndPath) 'Dir returns the name of the file
'Workbooks("read-data-from-text-file").Worksheets("run_nastran").Cells(2, 2) = FileName
FilePath = Left(FilenameAndPath, Len(FilenameAndPath) - Len(FileName) - 1)
'Delete previous run files


f06_file = "C:\MsFEM_Project\GRAND_MASTER\REPAIR_INCR_" & CStr(loop_incr) & "\MESO_CRIT_" & CStr(loop_incr) & _
   "\MICRO_CRIT_" & CStr(loop_incr) & "\MICRO_RVE_" & CStr(55) & ".f06"
f04_file = "C:\MsFEM_Project\GRAND_MASTER\REPAIR_INCR_" & CStr(loop_incr) & "\MESO_CRIT_" & CStr(loop_incr) & _
```

```
    "\MICRO_CRIT_" & CStr(loop_incr) & "\MICRO_RVE_" & CStr(55) & ".f04"
log_file = "C:\MsFEM_Project\GRAND_MASTER\REPAIR_INCR_" & CStr(loop_incr) & "\MESO_CRIT_" & CStr(loop_incr) & _
    "\MICRO_CRIT_" & CStr(loop_incr) & "\MICRO_RVE_" & CStr(55) & ".log"
If Len(Dir(f06_file)) > 0 Then
     Kill f06_file
     Kill f04_file
     Kill log_file
End If
Set WshShell = CreateObject("WScript.Shell")
' Setting the window directory to the incoming file location
WshShell.CurrentDirectory = FilePath
' chr(34) is a double quote (used if spaces in the path)
' add additional arguments to the end of the ShellCmd variable as needed.
ShellCmd = Chr(34) & "C:\MSC.Software\MSC_Nastran\20140\bin\nastran.exe" & Chr(34) _
    & " " _
    & Chr(34) & FileName & Chr(34)
' Run options: See http://ss64.com/vb/run.html for more options.
' 1 displays the shell window
' True waits for the command to complete
WshShell.Run ShellCmd, 1, True
Set WshShell = Nothing
Next j
'MsgBox ("Nastran runs complete !!!")
Application.ScreenUpdating = True
End Sub
```

14. VBA Script micro_mat_update ()

```
Option Explicit

Option Base 1

Sub micro_check_mat_update(loop_incr)

'Sub micro_check_mat_update(j_L, lim_loop)

'READ F06 FILE

Dim file1 As String

Dim volumebook, stressbook, resultbook As Workbook

Dim volumesht, E_sht, stresssht, resultsht, micro_crt_sht, micro_elem_prop_sht As Worksheet

Dim Arr_fibre(), Arr_matrix(), Arr_interface(), volume_mult, r As Variant

Dim norm_x_min, norm_y_min, norm_z_min, shr_xy_min, shr_yz_min, shr_zx_min, von_mises As Double

Dim norm_x_max, norm_y_max, norm_z_max, shr_xy_max, shr_yz_max, shr_zx_max, elem_vm_row As Double

Dim numelem, elem, i, j, k, a, b, c, ss, ff As Integer

'Dim loop_incr as integer

Dim lim_loop As Integer

'loop_incr = 1

'lim_loop = 1

numelem = 9216


Dim stress_matrix(9216, 8) As Variant

Dim range_row As Variant

Dim Fn As Object

Set Fn = Application.WorksheetFunction

Dim elem_x_tens, elem_y_comp, elem_xy, elem_vm As Integer

Dim xx_str, yy_str, zz_str, xy_str, yz_str, zx_str As Double
```

```
Application.ScreenUpdating = False

Workbooks("GRAND_MASTER_1.xlsm").Activate

Set resultsht = Worksheets("MESO_CRIT_STR")

Set micro_crt_sht = Worksheets("MICRO_CRIT_STR")

Set micro_elem_prop_sht = Worksheets("MICRO_RVE_ELEM_PROPS")

Set E_sht = Worksheets("MICRO_CONST_TANG")

E_sht.Cells(14, 24) = E_sht.Cells(14, 22)

j = 55

'j_L = 1

'For j = 1 To 3 ' loop through results files

'"C:\MsFEM_Project\TEST_BEAM\incrmnt_" & CStr(j_L) & "\micro_models\micro_" & CStr(j) & ".f06"

file1 = "C:\MsFEM_Project\GRAND_MASTER\REPAIR_INCR_" & CStr(loop_incr) & "\MESO_CRIT_" & CStr(loop_incr) & _

   "\MICRO_CRIT_" & CStr(loop_incr) & "\MICRO_RVE_" & CStr(55) & ".f06"

Workbooks.OpenText FileName:= _

      file1 _

      , Origin:=437, StartRow:=1, DataType:=xlDelimited, TextQualifier:= _

      xlDoubleQuote, ConsecutiveDelimiter:=True, Tab:=True, Semicolon:=False, _

      Comma:=False, Space:=True, Other:=False, FieldInfo:=Array(Array(1, 1), _

      Array(2, 1), Array(3, 1), Array(4, 1), Array(5, 1), Array(6, 1), Array(7, 1), Array(8, 1), _

      Array(9, 1), Array(10, 1), Array(11, 1), Array(12, 1), Array(13, 1)), _

      TrailingMinusNumbers:=True

Set stresssht = Workbooks("MICRO_RVE_" & CStr(j) & ".f06").Worksheets("MICRO_RVE_" & CStr(j))

k = 1

For i = 1 To 340000

   If stresssht.Cells(i, 2).Value = "CENTER" Then

      'elem
```

```
    stress_matrix(k, 1) = stresssht.Cells((i - 1), 2)
        If stresssht.Cells(i - 1, 2).Value = "ELEMENT-ID" Then
    'elem
            stress_matrix(k, 1) = stresssht.Cells(i - 8, 2)
        Else
        End If
'norm_x
    stress_matrix(k, 2) = stresssht.Cells(i, 4)
'norm_y
    stress_matrix(k, 3) = stresssht.Cells(i + 1, 3)
'norm_z
    stress_matrix(k, 4) = stresssht.Cells(i + 2, 3)
'shr_xy
    stress_matrix(k, 5) = stresssht.Cells(i, 6)
'shr_yz
    stress_matrix(k, 6) = stresssht.Cells(i + 1, 5)
'shr_zx
    stress_matrix(k, 7) = stresssht.Cells(i + 2, 5)
'von_mises
    stress_matrix(k, 8) = stresssht.Cells(i, 13)
        If stresssht.Cells(i, 13).Value = "" And stresssht.Cells(i, 12).Value <> "" Then
    'von_mises
            stress_matrix(k, 8) = stresssht.Cells(i, 12)
        ElseIf stresssht.Cells(i, 12).Value = "" Then
    'von_mises
            stress_matrix(k, 8) = stresssht.Cells(i, 11)
        Else
        End If
```

```
            k = k + 1
        Else
        End If
Next i
Workbooks("MICRO_RVE_" & CStr(j) & ".f06").Close SaveChanges:=False
'EXTRACT CRITICAL ELEMENTS AND THEIR STRESSES
'COLLECT MAXIMUM VON MISES STRESS
Dim vm_fibre(), vm_matrix(), vm_interface() As Variant
Dim dest_fibre, dest_matrix, dest_interface As Range
Dim fibre_elem(), matrix_elem(), interface_elem() As Variant
Set dest_fibre = Range(micro_elem_prop_sht.Cells(3, 76 + loop_incr * 6), micro_elem_prop_sht.Cells(3, 76 + loop_incr * 6))
Set dest_matrix = Range(micro_elem_prop_sht.Cells(3, 75 + loop_incr * 6), micro_elem_prop_sht.Cells(3, 75 + loop_incr * 6))
Set dest_interface = Range(micro_elem_prop_sht.Cells(3, 74 + loop_incr * 6), micro_elem_prop_sht.Cells(3, 74 + loop_incr * 6))
'CHECK MATRIX ELEMENT FAILURE
Range(micro_elem_prop_sht.Cells(3, 75 + loop_incr * 6), micro_elem_prop_sht.Cells(5380, 75 + loop_incr * 6)).ClearContents
Arr_matrix = Range(micro_elem_prop_sht.Cells(3, 69), micro_elem_prop_sht.Cells(6913, 69))
matrix_elem = Range(micro_elem_prop_sht.Cells(3, 70), micro_elem_prop_sht.Cells(3458, 70))
'*************************************************************
ReDim vm_matrix(UBound(matrix_elem), 1)
vm_matrix = Application.Index(stress_matrix, matrix_elem, 8)
dest_matrix.Resize(UBound(vm_matrix, 1), 1).Value = vm_matrix
'CHECK FIBRE ELEMENT FAILURE
Range(micro_elem_prop_sht.Cells(3, 76 + loop_incr * 6), micro_elem_prop_sht.Cells(5380, 76 + loop_incr * 6)).ClearContents
Arr_fibre = Range(micro_elem_prop_sht.Cells(3, 77), micro_elem_prop_sht.Cells(10753, 77))
fibre_elem = Range(micro_elem_prop_sht.Cells(3, 78), micro_elem_prop_sht.Cells(5378, 78))
```

```
'*****************************************************************

ReDim vm_fibre(UBound(fibre_elem), 1)

vm_fibre = Application.Index(stress_matrix, fibre_elem, 8)

dest_fibre.Resize(UBound(vm_fibre, 1), 1).Value = vm_fibre

'CHECK INTERFACE ELEMENT FAILURE

Range(micro_elem_prop_sht.Cells(3, 74 + loop_incr * 6), micro_elem_prop_sht.Cells(769, 74 + loop_incr * 6)).ClearContents

Arr_interface = Range(micro_elem_prop_sht.Cells(3, 73), micro_elem_prop_sht.Cells(769, 73))

interface_elem = Range(micro_elem_prop_sht.Cells(3, 74), micro_elem_prop_sht.Cells(386, 74))

'*****************************************************************

ReDim vm_interface(UBound(interface_elem), 1)

vm_interface = Application.Index(stress_matrix, interface_elem, 8)

dest_interface.Resize(UBound(vm_interface, 1), 1).Value = vm_interface

'IDENTIFY FAILED FIBRE*** , MATRIX, ELEMENTS ELEMENTS

If lim_loop = 1 Then

   Range(micro_elem_prop_sht.Cells(6000, 73 + loop_incr * 8), micro_elem_prop_sht.Cells(20000, 73 + loop_incr * 8)).ClearContents

   Range(micro_elem_prop_sht.Cells(6000, 75 + loop_incr * 8), micro_elem_prop_sht.Cells(20000, 75 + loop_incr * 8)).ClearContents

   Range(micro_elem_prop_sht.Cells(6000, 77 + loop_incr * 8), micro_elem_prop_sht.Cells(20000, 77 + loop_incr * 8)).ClearContents

Else

   'interface group

   Range(micro_elem_prop_sht.Cells(6000, 73 + (loop_incr) * 8), micro_elem_prop_sht.Cells(20000, 73 + (loop_incr) * 8)) = _

   Range(micro_elem_prop_sht.Cells(6000, 73 + (loop_incr - 1) * 8), micro_elem_prop_sht.Cells(20000, 73 + (loop_incr - 1) * 8))

   "matrix group

   Range(micro_elem_prop_sht.Cells(6000, 75 + loop_incr * 8), micro_elem_prop_sht.Cells(20000, 75 + loop_incr * 8)) = _
```

```
    Range(micro_elem_prop_sht.Cells(6000, 75 + (loop_incr - 1) * 8), micro_elem_prop_sht.Cells(20000, 75 + (loop_incr - 1) * 8))

    "fibre group

    Range(micro_elem_prop_sht.Cells(6000, 77 + loop_incr * 8), micro_elem_prop_sht.Cells(20000, 77 + loop_incr * 8)) = _

    Range(micro_elem_prop_sht.Cells(6000, 77 + (loop_incr - 1) * 8), micro_elem_prop_sht.Cells(20000, 77 + (loop_incr - 1) * 8))

End If

For ff = 1 To 5400

'interface group

    If micro_elem_prop_sht.Cells(2 + ff, 74 + loop_incr * 6) > 0.063 Or micro_elem_prop_sht.Cells(5999 + ff, 73 + (loop_incr - 1) * 8) = 555 Then

        micro_elem_prop_sht.Cells(5999 + ff, 73 + loop_incr * 8) = 555

    End If

'matrix group

    If micro_elem_prop_sht.Cells(2 + ff, 75 + loop_incr * 6) > 0.063 Or micro_elem_prop_sht.Cells(5999 + ff, 75 + (loop_incr - 1) * 8) = 777 Then

        micro_elem_prop_sht.Cells(5999 + ff, 75 + loop_incr * 8) = 777

    End If

'fibre group

    If micro_elem_prop_sht.Cells(2 + ff, 76 + loop_incr * 6) > 2.56 Or micro_elem_prop_sht.Cells(5999 + ff, 77 + (loop_incr - 1) * 8) = 888 Then

        micro_elem_prop_sht.Cells(5999 + ff, 77 + loop_incr * 8) = 888

    End If

Next ff

'TABULATE FIBRE, MATRIX AND INTERFACE STRESS RESULTS

'For ss = 1 To 10

    'fibre vm str

    micro_crt_sht.Cells(15 + loop_incr, 18) = micro_elem_prop_sht.Cells(7, 73 + 6 * loop_incr)

    'matrix vm str
```

```
    micro_crt_sht.Cells(15 + loop_incr, 17) = micro_elem_prop_sht.Cells(3, 73 + 6 * loop_incr)
    'interface vm str
    micro_crt_sht.Cells(15 + loop_incr, 16) = micro_elem_prop_sht.Cells(5, 73 + 6 * loop_incr)
'Next ss
'UPDATE FAILED ELEMENTS COLUMNS
Application.ScreenUpdating = True
End Sub
```

15. Fatigue VBA Script micro_mat_update ()

```
Option Explicit

Option Base 1

'Dim loop_incr, meso_lim_loop, micro_lim_loop As Integer

Sub micro_mat_update(loop_incr)

'Sub micro_check_mat_update(j_L, lim_loop)

Dim fNameAndPath As Variant

Dim TextLine, fName

Dim myCount, counter, a_elm, a_11, a_22, a_33, a_44, A_ii As Long

Dim counterp, a_elmp, a_11p, a_22p, a_33p, a_44p As Long

Dim range_row(1, 10320) As Variant

Dim stress_matrix(10320, 8) As Variant

'READ F06 FILE

Dim file1 As String

Dim volumebook, stressbook, resultbook As Workbook

Dim volumesht, E_sht, stresssht, resultsht, micro_crt_sht, micro_elem_prop_sht, load_sht As Worksheet

Dim Arr_fibre(), Arr_matrix(), Arr_interface(), volume_mult, r As Variant

Dim norm_x_min, norm_y_min, norm_z_min, shr_xy_min, shr_yz_min, shr_zx_min, von_mises As Double

Dim norm_x_max, norm_y_max, norm_z_max, shr_xy_max, shr_yz_max, shr_zx_max, elem_vm_row As Double

Dim numelem, elem, i, j, k, a, b, c, ss, ff As Integer

Dim lim_loop As Integer

lim_loop = 1

numelem = 10320

Dim Fn As Object

Set Fn = Application.WorksheetFunction
```

```
Dim elem_x_tens, elem_y_comp, elem_xy, elem_vm As Integer
Dim xx_str, yy_str, zz_str, xy_str, yz_str, zx_str As Double
Application.ScreenUpdating = False
Workbooks("GRAND_MASTER_2_F.xlsm").Activate
Set resultsht = Worksheets("MESO_CRIT_STR")
Set micro_crt_sht = Worksheets("MICRO_CRIT_STR")
Set micro_elem_prop_sht = Worksheets("MICRO_RVE_ELEM_PROPS")
Set E_sht = Worksheets("MICRO_CONST_TANG")
E_sht.Cells(14, 24) = E_sht.Cells(14, 22)
j = 55
   myCount = 1
   counter = 1
   counterp = 1
   a_elm = 8
   a_11 = 9
   a_22 = 11
   a_33 = 12
   a_44 = 14
   a_elmp = 2240015
   a_11p = 2240016
   a_22p = 2240018
   a_33p = 2240019
   a_44p = 2240021
   numelem = 10320
fNameAndPath = "C:\MsFEM_Project\GRAND_MASTER_F\REPAIR_INCR_" & CStr(loop_incr) & "\MESO_CRIT_" & CStr(loop_incr) & _
   "\MICRO_CRIT_" & CStr(loop_incr) & "\MICRO_RVE_" & CStr(55) & ".pch"
'Open outputFile For Output Shared As #2 ' Open file.
   Open fNameAndPath For Input As #1    ' Open file.
```

```
Do While Not EOF(1)    ' Loop until end of file.
   Line Input #1, TextLine    ' Read line into variable.
   'HEXA ELEMENT RESULTS
   A_ii = a_elm + (counter - 1) * 64
   If A_ii = myCount And Left(TextLine, 3) = "   " Then
         stress_matrix(counter, 1) = CLng((Mid(TextLine, 6, 5)))
   End If
   A_ii = a_11 + (counter - 1) * 64
   If A_ii = myCount And Left(TextLine, 6) = "-CONT-" Then
         stress_matrix(counter, 2) = CSng(Mid(TextLine, 42, 13))
      'shr_xy
         stress_matrix(counter, 5) = CSng(Mid(TextLine, 60, 13))
   End If
   A_ii = a_22 + (counter - 1) * 64
   If A_ii = myCount And Left(TextLine, 6) = "-CONT-" Then
      'von_mises
         stress_matrix(counter, 8) = CSng(Mid(TextLine, 60, 13))
   End If
   A_ii = a_33 + (counter - 1) * 64
   If A_ii = myCount And Left(TextLine, 6) = "-CONT-" Then
      'norm_y
         stress_matrix(counter, 3) = CSng(Mid(TextLine, 24, 13))
      'shr_yz
         stress_matrix(counter, 6) = CSng(Mid(TextLine, 42, 13))
   End If
   A_ii = a_44 + (counter - 1) * 64
   If A_ii = myCount And Left(TextLine, 6) = "-CONT-" Then
      'norm_z
```

```
            stress_matrix(counter, 4) = CSng(Mid(TextLine, 24, 13))
          'shr_zx
            stress_matrix(counter, 7) = CSng(Mid(TextLine, 42, 13))
          counter = counter + 1
        End If
        myCount = myCount + 1
    Loop
    Close #1    ' Close file.
'EXTRACT CRITICAL ELEMENTS AND THEIR STRESSES
'COLLECT MAXIMUM VON MISES STRESS
Dim vm_fibre(), vm_matrix(), vm_interface() As Variant
Dim dest_fibre, dest_matrix, dest_interface As Range
Dim fibre_elem(), matrix_elem(), interface_elem() As Variant
Set dest_fibre = Range(micro_elem_prop_sht.Cells(3, 76 + loop_incr * 6), micro_elem_prop_sht.Cells(3, 76 + loop_incr * 6))
Set dest_matrix = Range(micro_elem_prop_sht.Cells(3, 75 + loop_incr * 6), micro_elem_prop_sht.Cells(3, 75 + loop_incr * 6))
Set dest_interface = Range(micro_elem_prop_sht.Cells(3, 74 + loop_incr * 6), micro_elem_prop_sht.Cells(3, 74 + loop_incr * 6))
'CHECK MATRIX ELEMENT FAILURE
Range(micro_elem_prop_sht.Cells(3, 75 + loop_incr * 6), micro_elem_prop_sht.Cells(6002, 75 + loop_incr * 6)).ClearContents
Arr_matrix = Range(micro_elem_prop_sht.Cells(3, 69), micro_elem_prop_sht.Cells(6913, 69))
matrix_elem = Range(micro_elem_prop_sht.Cells(3, 69), micro_elem_prop_sht.Cells(2882, 69))
'REDIMENSION ARRAYS TO UBOUND
'*************************************************************
ReDim vm_matrix(UBound(matrix_elem), 1)
vm_matrix = Application.Index(stress_matrix, matrix_elem, 8)
dest_matrix.Resize(UBound(vm_matrix, 1), 1).Value = vm_matrix
'CHECK FIBRE ELEMENT FAILURE
```

```
Range(micro_elem_prop_sht.Cells(3, 76 + loop_incr * 6), micro_elem_prop_sht.Cells(6002, 76 + loop_incr * 6)).ClearContents

Arr_fibre = Range(micro_elem_prop_sht.Cells(3, 77), micro_elem_prop_sht.Cells(1075, 77))

fibre_elem = Range(micro_elem_prop_sht.Cells(3, 77), micro_elem_prop_sht.Cells(6002, 77))

'REDIMENSION ARRAYS TO UBOUND

'**********************************************************

ReDim vm_fibre(UBound(fibre_elem), 1)

vm_fibre = Application.Index(stress_matrix, fibre_elem, 8)

dest_fibre.Resize(UBound(vm_fibre, 1), 1).Value = vm_fibre

'CHECK INTERFACE ELEMENT FAILURE


Range(micro_elem_prop_sht.Cells(3, 74 + loop_incr * 6), micro_elem_prop_sht.Cells(6002, 74 + loop_incr * 6)).ClearContents

Arr_interface = Range(micro_elem_prop_sht.Cells(3, 73), micro_elem_prop_sht.Cells(769, 73))

interface_elem = Range(micro_elem_prop_sht.Cells(3, 73), micro_elem_prop_sht.Cells(1442, 73))

'**********************************************************

ReDim vm_interface(UBound(interface_elem), 1)

vm_interface = Application.Index(stress_matrix, interface_elem, 8)

dest_interface.Resize(UBound(vm_interface, 1), 1).Value = vm_interface

'IDENTIFY FAILED FIBRE*** , MATRIX, ELEMENTS ELEMENTS


If loop_incr = 1 Then

   'Clear failure makrs

   Range(micro_elem_prop_sht.Cells(6008, 73 + loop_incr * 8), micro_elem_prop_sht.Cells(12020, 73 + loop_incr * 8)).ClearContents

   Range(micro_elem_prop_sht.Cells(6008, 75 + loop_incr * 8), micro_elem_prop_sht.Cells(12020, 75 + loop_incr * 8)).ClearContents

   Range(micro_elem_prop_sht.Cells(6008, 77 + loop_incr * 8), micro_elem_prop_sht.Cells(12020, 77 + loop_incr * 8)).ClearContents
```

'Clear failure indices

Range(micro_elem_prop_sht.Cells(12050, 73 + loop_incr * 8), micro_elem_prop_sht.Cells(18062, 73 + loop_incr * 8)).ClearContents

Range(micro_elem_prop_sht.Cells(12050, 75 + loop_incr * 8), micro_elem_prop_sht.Cells(18062, 75 + loop_incr * 8)).ClearContents

Range(micro_elem_prop_sht.Cells(12050, 77 + loop_incr * 8), micro_elem_prop_sht.Cells(18062, 77 + loop_incr * 8)).ClearContents

Else

If micro_lim_loop = 1 Then

'interface group

'Copy previous markers

Range(micro_elem_prop_sht.Cells(6008, 73 + (loop_incr) * 8), micro_elem_prop_sht.Cells(12020, 73 + (loop_incr) * 8)) = _

Range(micro_elem_prop_sht.Cells(6008, 73 + (loop_incr - 1) * 8), micro_elem_prop_sht.Cells(12020, 73 + (loop_incr - 1) * 8))

'Copy previous indices

Range(micro_elem_prop_sht.Cells(12050, 73 + loop_incr * 8), micro_elem_prop_sht.Cells(18062, 73 + loop_incr * 8)) = _

Range(micro_elem_prop_sht.Cells(12050, 73 + (loop_incr - 1) * 8), micro_elem_prop_sht.Cells(18062, 73 + (loop_incr - 1) * 8))

"matrix group

'Copy previous markers

Range(micro_elem_prop_sht.Cells(6008, 75 + loop_incr * 8), micro_elem_prop_sht.Cells(12020, 75 + loop_incr * 8)) = _

Range(micro_elem_prop_sht.Cells(6008, 75 + (loop_incr - 1) * 8), micro_elem_prop_sht.Cells(12020, 75 + (loop_incr - 1) * 8))

'Copy previous indices

Range(micro_elem_prop_sht.Cells(12050, 75 + loop_incr * 8), micro_elem_prop_sht.Cells(18062, 75 + loop_incr * 8)) = _

```
    Range(micro_elem_prop_sht.Cells(12050, 75 + (loop_incr - 1) * 8), micro_elem_prop_sht.Cells(18062, 75 + (loop_incr - 1) * 8))

    "fibre group

    'Copy previous markers

    Range(micro_elem_prop_sht.Cells(6008, 77 + loop_incr * 8), micro_elem_prop_sht.Cells(12020, 77 + loop_incr * 8)) = _

    Range(micro_elem_prop_sht.Cells(6008, 77 + (loop_incr - 1) * 8), micro_elem_prop_sht.Cells(12020, 77 + (loop_incr - 1) * 8))

    'Copy previous indices

    Range(micro_elem_prop_sht.Cells(12050, 77 + loop_incr * 8), micro_elem_prop_sht.Cells(18062, 77 + loop_incr * 8)) = _

    Range(micro_elem_prop_sht.Cells(12050, 77 + (loop_incr - 1) * 8), micro_elem_prop_sht.Cells(18062, 77 + (loop_incr - 1) * 8))

    End If

End If

'declare fatigue Nf_fib and Nf_mat

Dim Nf_fib, Nf_mat, deltaN As Double

Dim S_fib, S_mat As Double

Set load_sht = Worksheets("LOAD_CONTROL")

deltaN = load_sht.Cells(6 + loop_incr, 6)

micro_elem_prop_sht.Activate

For ff = 1 To 5400

'interface group

    If micro_elem_prop_sht.Cells(2 + ff, 74 + loop_incr * 6) > 3.85 Or _

       micro_elem_prop_sht.Cells(12050 + ff, 73 + loop_incr * 8) > 1 Or _

       micro_elem_prop_sht.Cells(6007 + ff, 73 + (loop_incr - 1) * 8) = 555 Or _

       micro_elem_prop_sht.Cells(12050 + ff, 73 + (loop_incr - 1) * 8) > 1 Then

        'Static update

        micro_elem_prop_sht.Cells(6007 + ff, 73 + loop_incr * 8) = 555

    Else
```

```
        'Fatigue update after static check
        If micro_lim_loop = 1 Then
        S_fib = micro_elem_prop_sht.Cells(2 + ff, 74 + loop_incr * 6)
        Nf_fib = 2.71828 ^ ((S_fib - 4.077) / -0.085)
        micro_elem_prop_sht.Cells(12050 + ff, 73 + loop_incr * 8) =
micro_elem_prop_sht.Cells(12050 + ff, 73 + loop_incr * 8) + deltaN / Nf_fib
        End If
    End If
'matrix group
    If micro_elem_prop_sht.Cells(2 + ff, 75 + loop_incr * 6) > 0.045 Or _
      micro_elem_prop_sht.Cells(12050 + ff, 75 + loop_incr * 8) > 1 Or _
      micro_elem_prop_sht.Cells(6007 + ff, 75 + (loop_incr - 1) * 8) = 777 Or _
      micro_elem_prop_sht.Cells(12050 + ff, 75 + (loop_incr - 1) * 8) > 1 Then
        ' Static update
        micro_elem_prop_sht.Cells(6007 + ff, 75 + loop_incr * 8) = 777
    Else
        'Fatigue update after static check
        If micro_lim_loop = 1 Then
        S_mat = micro_elem_prop_sht.Cells(2 + ff, 75 + loop_incr * 6)
        Nf_mat = 2.71828 ^ ((S_mat - 0.0453) / -0.002)
        micro_elem_prop_sht.Cells(12050 + ff, 75 + loop_incr * 8) =
micro_elem_prop_sht.Cells(12050 + ff, 75 + loop_incr * 8) + deltaN / Nf_mat
        End If
    End If
'fibre group
    If micro_elem_prop_sht.Cells(2 + ff, 76 + loop_incr * 6) > 3.85 Or _
      micro_elem_prop_sht.Cells(12050 + ff, 77 + loop_incr * 8) > 1 Or _
      micro_elem_prop_sht.Cells(6007 + ff, 77 + (loop_incr - 1) * 8) = 888 Or _
      micro_elem_prop_sht.Cells(12050 + ff, 77 + (loop_incr - 1) * 8) > 1 Then
```

```
        ' Static update
        micro_elem_prop_sht.Cells(6007 + ff, 77 + loop_incr * 8) = 888
    Else
        'Fatigue update after static check
        If micro_lim_loop = 1 Then
        S_fib = micro_elem_prop_sht.Cells(2 + ff, 76 + loop_incr * 6)
        Nf_fib = 2.71828 ^ ((S_fib - 4.077) / -0.085)
        micro_elem_prop_sht.Cells(12050 + ff, 77 + loop_incr * 8) = micro_elem_prop_sht.Cells(12050 + ff, 77 + loop_incr * 8) + deltaN / Nf_fib
        End If
    End If
Next ff
'TABULATE FIBRE, MATRIX AND INTERFACE STRESS RESULTS
'For ss = 1 To 10
    'fibre vm str
    micro_crt_sht.Cells(15 + loop_incr, 18) = micro_elem_prop_sht.Cells(7, 73 + 6 * loop_incr)
    'matrix vm str
    micro_crt_sht.Cells(15 + loop_incr, 17) = micro_elem_prop_sht.Cells(3, 73 + 6 * loop_incr)
    'interface vm str
    micro_crt_sht.Cells(15 + loop_incr, 16) = micro_elem_prop_sht.Cells(5, 73 + 6 * loop_incr)
'Next ss
'UPDATE FAILED ELEMENTS COLUMNS
micro_lim_loop = micro_lim_loop + 1
Application.ScreenUpdating = True
End Sub
```

16. Fatigue VBA Script meso_check_mat_update ()

```
Option Explicit
'Dim loop_incr, meso_lim_loop, micro_lim_loop As Integer
Option Base 1
Sub meso_check_mat_update(loop_incr)

'READ F06 FILE
Dim norm_x, norm_y, norm_z, shr_xy, shr_yz, shr_zx, von_mises As Double
Dim fNameAndPath As Variant
Dim TextLine, fName
Dim myCount, counter, a_elm, a_11, a_22, a_33, a_44, A_ii As Long
Dim counterp, a_elmp, a_11p, a_22p, a_33p, a_44p As Long
Dim range_row(1, 17920) As Variant
Dim stress_matrix(17920, 8) As Variant
Dim r As Integer
'*****************************************************************
Dim volumebook, stressbook, resultbook As Workbook
Dim volumesht, E_sht, stresssht, resultsht, meso_crt_sht, meso_elem_prop_sht, micro_crt_sht As Worksheet
Dim norm_x_min, norm_y_min, norm_z_min, shr_xy_min, shr_yz_min, shr_zx_min As Double
Dim norm_x_max, norm_y_max, norm_z_max, shr_xy_max, shr_yz_max, shr_zx_max, elem_vm_row As Double
Dim numelem, elem, i, j, k, a, b, c, ss, ff As Integer
Dim elem_x_tens, elem_y_comp, elem_xy, elem_vm As Integer
Dim xx_str, yy_str, zz_str, xy_str, yz_str, zx_str As Double
Dim Fn As Object
```

```
Set Fn = Application.WorksheetFunction
Application.ScreenUpdating = False
Workbooks("GRAND_MASTER_2_F.xlsm").Activate
Set resultsht = Worksheets("REPAIR_CRIT_STR")
Set meso_crt_sht = Worksheets("MESO_CRIT_STR")
Set meso_elem_prop_sht = Worksheets("MESO_RVE_ELEM_PROPS")
Set E_sht = Worksheets("MESO_CONST_TANG")
Set micro_crt_sht = Worksheets("MICRO_CRIT_STR")
E_sht.Cells(14, 24) = E_sht.Cells(14, 22)
j = 55
'trial inputs
'Dim loop_incr As Integer
'loop_incr = 1
'meso_lim_loop = 1
'*************************************************************
numelem = 17920
'loop_incr = 1
    myCount = 1
    counter = 1
    counterp = 1
    a_elm = 8
    a_11 = 9
    a_22 = 11
    a_33 = 12
    a_44 = 14
    a_elmp = 2240015
    a_11p = 2240016
    a_22p = 2240018
```

```
a_33p = 2240019

a_44p = 2240021

numelem = 17920

'working input folder location

fNameAndPath = "C:\MsFEM_Project\GRAND_MASTER_F\REPAIR_INCR_" & CStr(loop_incr) & "\MESO_CRIT_" & CStr(loop_incr) & _

"\MESO_RVE_" & CStr(j) & ".pch"

Open fNameAndPath For Input As #1    ' Open file.

Do While Not EOF(1)    ' Loop until end of file.

    Line Input #1, TextLine    ' Read line into variable.

    'HEXA ELEMENT RESULTS

    A_ii = a_elm + (counter - 1) * 64

    If A_ii = myCount And Left(TextLine, 3) = "   " Then

        'Print #2, TextLine & " ELEMENT NO " & CStr(myCount) & "   " & CStr(counter) & "   " & Mid(TextLine, 6, 5) ' write variable to output file.

        'counter = counter + 1

        'elem

            stress_matrix(counter, 1) = CLng((Mid(TextLine, 6, 5)))

    End If

    A_ii = a_11 + (counter - 1) * 64

    If A_ii = myCount And Left(TextLine, 6) = "-CONT-" Then

        'Print #2, TextLine & " X and XY  " & CStr(myCount) & "   " & CStr(counter) & " X " & Mid(TextLine, 42, 13) & " XY " & Mid(TextLine, 60, 13) ' write variable to output file.

        'counter = counter + 1

        'norm_x

            stress_matrix(counter, 2) = CSng(Mid(TextLine, 42, 13))

        'shr_xy

            stress_matrix(counter, 5) = CSng(Mid(TextLine, 60, 13))

    End If
```

```
A_ii = a_22 + (counter - 1) * 64

If A_ii = myCount And Left(TextLine, 6) = "-CONT-" Then

   'Print #2, TextLine & " VM  " & CStr(myCount) & " VM " & Mid(TextLine, 60, 13) ' write variable to output file.

   'counter = counter + 1

   'von_mises

      stress_matrix(counter, 8) = CSng(Mid(TextLine, 60, 13))

End If

A_ii = a_33 + (counter - 1) * 64

If A_ii = myCount And Left(TextLine, 6) = "-CONT-" Then

   'Print #2, TextLine & " Y and YZ  " & CStr(myCount) & "   " & CStr(counter) & " Y " & Mid(TextLine, 24, 13) & " YZ " & Mid(TextLine, 42, 13) ' write variable to output file.

   'counter = counter + 1

   'norm_y

      stress_matrix(counter, 3) = CSng(Mid(TextLine, 24, 13))

   'shr_yz

      stress_matrix(counter, 6) = CSng(Mid(TextLine, 42, 13))

End If

A_ii = a_44 + (counter - 1) * 64

If A_ii = myCount And Left(TextLine, 6) = "-CONT-" Then

   'Print #2, TextLine & " Z and ZX  " & CStr(myCount) & "   " & CStr(counter) & " Z " & Mid(TextLine, 24, 13) & " ZX " & Mid(TextLine, 42, 13) ' write variable to output file.

   'norm_z

      stress_matrix(counter, 4) = CSng(Mid(TextLine, 24, 13))

   'shr_zx

      stress_matrix(counter, 7) = CSng(Mid(TextLine, 42, 13))

   counter = counter + 1

End If

'PENTA ELEMENT RESULTS
```

```
A_ii = a_elmp + (counterp - 1) * 50

If A_ii = myCount And Left(TextLine, 3) = "   " Then

   'Print #2, TextLine & " ELEMENT NO " & CStr(myCount) & "    " & CStr(counterp + counter - 1) & "   " & Mid(TextLine, 6, 5) ' write variable to output file.

   'counter = counter + 1

   'elem

   stress_matrix(counterp + counter - 1, 1) = CLng((Mid(TextLine, 6, 5)))

End If

A_ii = a_11p + (counterp - 1) * 50

If A_ii = myCount And Left(TextLine, 6) = "-CONT-" Then

   'Print #2, TextLine & " X and XY  " & CStr(myCount) & "    " & CStr(counterp + counter - 1) & " X " & Mid(TextLine, 42, 13) & " XY " & Mid(TextLine, 60, 13) ' write variable to output file.

   'counter = counter + 1

   'norm_x

      stress_matrix(counterp + counter - 1, 2) = CSng(Mid(TextLine, 42, 13))

   'shr_xy

      stress_matrix(counterp + counter - 1, 5) = CSng(Mid(TextLine, 60, 13))

End If

A_ii = a_22p + (counterp - 1) * 50

If A_ii = myCount And Left(TextLine, 6) = "-CONT-" Then

   'Print #2, TextLine & " VM   " & CStr(myCount) & " VM " & Mid(TextLine, 60, 13)   ' write variable to output file.

   'counter = counter + 1

   'von_mises

      stress_matrix(counter + counterp - 1, 8) = CSng(Mid(TextLine, 60, 13))

End If

A_ii = a_33p + (counterp - 1) * 50

If A_ii = myCount And Left(TextLine, 6) = "-CONT-" Then
```

```
        'Print #2, TextLine & " Y and YZ  " & CStr(myCount) & "    " & CStr(counterp + counter - 1) & " Y " & Mid(TextLine, 24, 13) & " YZ " & Mid(TextLine, 42, 13) ' write variable to output file.

        'counter = counter + 1

        'norm_y

           stress_matrix(counter + counterp - 1, 3) = CSng(Mid(TextLine, 24, 13))

        'shr_yz

           stress_matrix(counter + counterp - 1, 6) = CSng(Mid(TextLine, 42, 13))

      End If

      A_ii = a_44p + (counterp - 1) * 50

      If A_ii = myCount And Left(TextLine, 6) = "-CONT-" Then

        'Print #2, TextLine & " Z and ZX  " & CStr(myCount) & "    " & CStr(counterp + counter - 1) & " Z " & Mid(TextLine, 24, 13) & " ZX " & Mid(TextLine, 42, 13) ' write variable to output file.

        'norm_z

           stress_matrix(counter + counterp - 1, 4) = CSng(Mid(TextLine, 24, 13))

        'shr_zx

           stress_matrix(counter + counterp - 1, 7) = CSng(Mid(TextLine, 42, 13))

        counterp = counterp + 1

      End If

      myCount = myCount + 1

   Loop

   'Close #2

   Close #1    ' Close file.

'EXTRACT CRITICAL ELEMENTS AND THEIR STRESSES

   'COLLECT MAXIMUM VON MISES STRESS

Dim vm_MATRIX_1(), vm_MATRIX_2(), vm_WEFTTOW() As Variant

Dim vm_WEFTTOW_RAMP_1(), vm_WEFTTOW_RAMP_2(), vm_WARPTOW(), vm_WARPTOW_RAMP_1(), vm_WARPTOW_RAMP_2() As Variant
```

```
Dim dest_MATRIX_1, dest_MATRIX_2, dest_WEFTTOW, dest_WEFTTOW_RAMP_1, dest_WEFTTOW_RAMP_2, dest_ As Range

Dim dest_WARPTOW, dest_WARPTOW_RAMP_1, dest_WARPTOW_RAMP_2 As Range

Dim MATRIX_1_elem(), MATRIX_2_elem(), WEFTTOW_elem(), WEFTTOW_RAMP_1_elem(), WEFTTOW_RAMP_2_elem() As Variant

Dim WARPTOW_elem(), WARPTOW_RAMP_1_elem(), WARPTOW_RAMP_2_elem() As Variant

Dim Arr_MATRIX_1(), Arr_MATRIX_2(), Arr_WEFTTOW(), Arr_WEFTTOW_RAMP_1(), Arr_WEFTTOW_RAMP_2() As Variant

Dim Arr_WARPTOW(), Arr_WARPTOW_RAMP_1(), Arr_WARPTOW_RAMP_2() As Variant

Set dest_MATRIX_1 = Range(meso_elem_prop_sht.Cells(3, 111 + loop_incr * 12), meso_elem_prop_sht.Cells(3, 111 + loop_incr * 12))

Set dest_WEFTTOW = Range(meso_elem_prop_sht.Cells(3, 112 + loop_incr * 12), meso_elem_prop_sht.Cells(3, 112 + loop_incr * 12))

Set dest_WARPTOW = Range(meso_elem_prop_sht.Cells(3, 113 + loop_incr * 12), meso_elem_prop_sht.Cells(3, 113 + loop_incr * 12))

'IDENTIFY FAILED MATRIX_1,MATRIX_2 ,WEFTTOW,WEFTTOW_RAMP_1 ,WEFTTOW_RAMP_2 ,WARPTOW,WARPTOW_RAMP_1 ,WARPTOW_RAMP_2 ELEMENTS

   'CHECK MATRIX_1 ELEMENT FAILURE

Range(meso_elem_prop_sht.Cells(3, 123 + loop_incr * 12), meso_elem_prop_sht.Cells(9995, 123 + loop_incr * 12)).ClearContents

MATRIX_1_elem = Range(meso_elem_prop_sht.Cells(3, 93), meso_elem_prop_sht.Cells(6402, 93))

   'REDIMENSION ARRAYS TO UBOUND

ReDim vm_MATRIX_1(UBound(MATRIX_1_elem), 1)

vm_MATRIX_1 = Application.Index(stress_matrix, MATRIX_1_elem, 8)

dest_MATRIX_1.Resize(UBound(vm_MATRIX_1, 1), 1).Value = vm_MATRIX_1 'paste values

   'CHECK WEFTTOW ELEMENT FAILURE

Range(meso_elem_prop_sht.Cells(3, 124 + loop_incr * 12), meso_elem_prop_sht.Cells(9995, 124 + loop_incr * 12)).ClearContents

WEFTTOW_elem = Range(meso_elem_prop_sht.Cells(3, 97), meso_elem_prop_sht.Cells(5762, 97))
```

```
    'REDIMENSION ARRAYS TO UBOUND
ReDim vm_MATRIX_2(UBound(WEFTTOW_elem), 1)
vm_WEFTTOW = Application.Index(stress_matrix, WEFTTOW_elem, 8)
dest_WEFTTOW.Resize(UBound(vm_WEFTTOW, 1), 1).Value = vm_WEFTTOW 'paste values
    'CHECK WARPTOW ELEMENT FAILURE
Range(meso_elem_prop_sht.Cells(3, 125 + loop_incr * 12), meso_elem_prop_sht.Cells(9995, 125 + loop_incr * 12)).ClearContents
WARPTOW_elem = Range(meso_elem_prop_sht.Cells(3, 101), meso_elem_prop_sht.Cells(5762, 101))
    'REDIMENSION ARRAYS TO UBOUND
ReDim vm_WEFTTOW(UBound(WEFTTOW_elem), 1)
vm_WARPTOW = Application.Index(stress_matrix, WARPTOW_elem, 8)
dest_WARPTOW.Resize(UBound(vm_WARPTOW, 1), 1).Value = vm_WARPTOW 'paste values
    'IDENTIFY FAILED MATRIX_1,MATRIX_2 ,WEFTTOW,WEFTTOW_RAMP_1 ,WEFTTOW_RAMP_2 ,WARPTOW,WARPTOW_RAMP_1 ,WARPTOW_RAMP_2 ELEMENTS
If loop_incr = 1 Then
    'Clear failure makrs
    Range(meso_elem_prop_sht.Cells(10004, 108 + loop_incr * 18), _
    meso_elem_prop_sht.Cells(20009, 108 + (loop_incr) * 18)).ClearContents
    Range(meso_elem_prop_sht.Cells(10004, 110 + loop_incr * 18), _
    meso_elem_prop_sht.Cells(20009, 110 + loop_incr * 18)).ClearContents
    Range(meso_elem_prop_sht.Cells(10004, 112 + loop_incr * 18), _
    meso_elem_prop_sht.Cells(20006, 112 + (loop_incr) * 18)).ClearContents
    'Clear failure indices
    Range(meso_elem_prop_sht.Cells(20050, 108 + loop_incr * 18), _
    meso_elem_prop_sht.Cells(29000, 108 + loop_incr * 18)).ClearContents
    Range(meso_elem_prop_sht.Cells(20050, 110 + loop_incr * 18), _
    meso_elem_prop_sht.Cells(29000, 110 + (loop_incr) * 18)).ClearContents
```

```
    Range(meso_elem_prop_sht.Cells(20050, 112 + loop_incr * 18), _
    meso_elem_prop_sht.Cells(29000, 112 + loop_incr * 18)).ClearContents
Else
'MATRIX_1 group
    If meso_lim_loop = 1 Then
    'Copy previous markers
    Range(meso_elem_prop_sht.Cells(10004, 108 + loop_incr * 18), meso_elem_prop_sht.Cells(20009, 108 + (loop_incr) * 18)) = _
    Range(meso_elem_prop_sht.Cells(10004, 108 + (loop_incr - 1) * 18), meso_elem_prop_sht.Cells(20009, 108 + (loop_incr - 1) * 18))
    'Copy previous indices
    Range(meso_elem_prop_sht.Cells(20050, 108 + loop_incr * 18), meso_elem_prop_sht.Cells(29000, 108 + loop_incr * 18)) = _
    Range(meso_elem_prop_sht.Cells(20050, 108 + (loop_incr - 1) * 18), meso_elem_prop_sht.Cells(29000, 108 + (loop_incr - 1) * 18))
'WEFTTOW group
    'Copy previous markers
    Range(meso_elem_prop_sht.Cells(10004, 110 + loop_incr * 18), meso_elem_prop_sht.Cells(20009, 110 + loop_incr * 18)) = _
    Range(meso_elem_prop_sht.Cells(10004, 110 + (loop_incr - 1) * 18), meso_elem_prop_sht.Cells(20009, 110 + (loop_incr - 1) * 18))
    'Copy previous indices
    Range(meso_elem_prop_sht.Cells(20050, 110 + loop_incr * 18), meso_elem_prop_sht.Cells(29000, 110 + loop_incr * 18)) = _
    Range(meso_elem_prop_sht.Cells(20050, 110 + (loop_incr - 1) * 18), meso_elem_prop_sht.Cells(29000, 110 + (loop_incr - 1) * 18))
'WARPTOW group
    'Copy previous markers
    Range(meso_elem_prop_sht.Cells(10004, 112 + loop_incr * 18), meso_elem_prop_sht.Cells(20009, 112 + (loop_incr) * 18)) = _
    Range(meso_elem_prop_sht.Cells(10004, 112 + (loop_incr - 1) * 18), meso_elem_prop_sht.Cells(20009, 112 + (loop_incr - 1) * 18))
```

```
    'Copy previous indices
    Range(meso_elem_prop_sht.Cells(20050, 112 + loop_incr * 18), meso_elem_prop_sht.Cells(29000, 112 + loop_incr * 18)) = _
    Range(meso_elem_prop_sht.Cells(20050, 112 + (loop_incr - 1) * 18), meso_elem_prop_sht.Cells(29000, 112 + (loop_incr - 1) * 18))
    End If
End If
'declare fatigue Nf_weft, Nf_warp and Nf_mat
Dim Nf_weft, Nf_warp, Nf_mat, deltaN As Double
Dim S_weft, S_warp, S_mat As Double
Dim load_sht As Worksheet
Set load_sht = Worksheets("LOAD_CONTROL")
deltaN = load_sht.Cells(6 + loop_incr, 6)
For ff = 1 To 6400
    'MATRIX_1 group
    If meso_elem_prop_sht.Cells(2 + ff, 111 + loop_incr * 12) > 45 Or _
    meso_elem_prop_sht.Cells(20050 + ff, 108 + (loop_incr) * 18) > 1 Or _
    meso_elem_prop_sht.Cells(10004 + ff, 108 + (loop_incr - 1) * 18) = 111 Or _
    meso_elem_prop_sht.Cells(20050 + ff, 108 + (loop_incr - 1) * 18) > 1 Then
        'Static update
        meso_elem_prop_sht.Cells(10004 + ff, 108 + loop_incr * 18) = 111
   Else
         'Fatigue update after static check
        If meso_lim_loop = 1 Then
        S_mat = meso_elem_prop_sht.Cells(2 + ff, 111 + loop_incr * 12)
        Nf_mat = 2.71828 ^ ((S_mat - 45.25) / -2.085)
        meso_elem_prop_sht.Cells(20050 + ff, 108 + loop_incr * 18) = meso_elem_prop_sht.Cells(20050 + ff, 108 + loop_incr * 18) + deltaN / Nf_mat
        End If
```

```
End If
'WEFTTOW group
If meso_elem_prop_sht.Cells(2 + ff, 112 + loop_incr * 12) > 570 Or _
meso_elem_prop_sht.Cells(20050 + ff, 110 + (loop_incr) * 18) > 1 Or _
meso_elem_prop_sht.Cells(10004 + ff, 110 + (loop_incr - 1) * 18) = 222 Or _
meso_elem_prop_sht.Cells(20050 + ff, 110 + (loop_incr - 1) * 18) > 1 Then

    'Static update
    meso_elem_prop_sht.Cells(10004 + ff, 110 + loop_incr * 18) = 222
   Else
        'Fatigue update after static check
     If meso_lim_loop = 1 Then
     S_weft = meso_elem_prop_sht.Cells(2 + ff, 112 + loop_incr * 12)
     Nf_weft = 2.71828 ^ ((S_weft - 570) / -20.85)
     meso_elem_prop_sht.Cells(20050 + ff, 110 + loop_incr * 18) =
meso_elem_prop_sht.Cells(20050 + ff, 110 + loop_incr * 18) + deltaN / Nf_weft
     End If
End If
'WARPTOW group
If meso_elem_prop_sht.Cells(2 + ff, 113 + loop_incr * 12) > 2400 Or _
meso_elem_prop_sht.Cells(20050 + ff, 112 + (loop_incr) * 18) > 1 Or _
meso_elem_prop_sht.Cells(10004 + ff, 112 + (loop_incr - 1) * 18) = 333 Or _
meso_elem_prop_sht.Cells(20050 + ff, 112 + (loop_incr - 1) * 18) > 1 Then
    'Static update
    meso_elem_prop_sht.Cells(10004 + ff, 112 + loop_incr * 18) = 333
Else
        'Fatigue update after static check
    If meso_lim_loop = 1 Then
    S_warp = meso_elem_prop_sht.Cells(2 + ff, 113 + loop_incr * 12)
```

```
        Nf_warp = 2.71828 ^ ((S_warp - 2400) / -84.6)
        meso_elem_prop_sht.Cells(20050 + ff, 112 + loop_incr * 18) = meso_elem_prop_sht.Cells(20050 + ff, 112 + loop_incr * 18) + deltaN / Nf_warp
        End If
    End If
Next ff
'EXTRACT CRITICAL ELEMENTS AND THEIR STRESSES
'xx_str
micro_crt_sht.Cells(25, 11) = Fn.Max(Fn.Index(stress_matrix, WARPTOW_elem, 2))
'yy_str
micro_crt_sht.Cells(26, 12) = Fn.Max(Fn.Index(stress_matrix, WARPTOW_elem, 3))
'zz_str
micro_crt_sht.Cells(27, 13) = Fn.Max(Fn.Index(stress_matrix, WARPTOW_elem, 4))
'xy_str
micro_crt_sht.Cells(25, 12) = Fn.Max(Fn.Index(stress_matrix, WARPTOW_elem, 5))
'yz_str
micro_crt_sht.Cells(26, 13) = Fn.Max(Fn.Index(stress_matrix, WARPTOW_elem, 6))
'zx_str
micro_crt_sht.Cells(25, 13) = Fn.Max(Fn.Index(stress_matrix, WARPTOW_elem, 7))
    'MATRIX_1 vm str
    meso_crt_sht.Cells(15 + loop_incr, 16) = meso_elem_prop_sht.Cells(3, 122 + 12 * loop_incr)
    'WEFTTOW vm str
    meso_crt_sht.Cells(15 + loop_incr, 17) = meso_elem_prop_sht.Cells(5, 122 + 12 * loop_incr)
    'WARPTOW vm str
    meso_crt_sht.Cells(15 + loop_incr, 18) = meso_elem_prop_sht.Cells(7, 122 + 12 * loop_incr)
meso_lim_loop = meso_lim_loop + 1
Application.ScreenUpdating = True
End Sub
```