

**Investigating a Finite Element Model Updating
Methodology for Characterizing Mechanical Properties of
NCF Composites**

by

Pasworn Chanchai

Submitted in partial fulfillment of the requirements
for the degree of

MASTER OF SCIENCE IN AEROSPACE ENGINEERING

THE UNIVERSITY OF TEXAS AT ARLINGTON

August 2021

Copyright © by Pasworn Chanchai 2021

All Rights Reserved

Acknowledgements

I would like to express my sincere gratitude to Dr. Andrew Makeev for giving me an opportunity to work with one of the most talented and amazing people at the UTA's Advanced Materials and Structures Laboratory.

I would also like to express my thanks and appreciation from the bottom of my heart to Dr. Guillaume Seon who has always been patient in every step of my learning during this Covid pandemic. Without his guidance and his unwavering willingness to dedicate the time mentoring me, this thesis would not have been possible. The knowledge and experience I have acquired while working with him will allow me to become a better engineer.

I wish to express my appreciation to Dr. Kent Lawrence, and Dr. Paul Davidson for serving on my thesis committee and giving me their precious time. My acknowledgement also goes to Mr. Hubert Schreier and his team for a technical support with VIC3D software.

Additionally, I am grateful for having such wonderful parents who give me a beautiful life and encourage me to pursue my passions and fulfill my dreams. I wish to express my love and gratitude to Kring who always stands by me in all the highs and lows. Lastly, special thanks go to Mr. Ron Reber, my professional and life mentor, who always be there when help is needed.

June 22, 2021

Table of Content

Acknowledgements.....	iii
List of Figures	vi
List of Tables.....	viii
Abstract.....	ix
Chapter 1 - Introduction.....	1
1.1 Finite Element Model Updating.....	2
1.2 Short-Beam Shear Experiment	2
1.3 Non-Crimp Fabric Composites	6
Chapter 2 - Finite Element Model Updating Algorithm.....	8
2.1 The Inverse Problem	8
2.2 The Optimization Algorithm	10
2.3 The Jacobian Matrix.....	12
2.3.1 The Analytical Jacobian	12
2.3.2 The Numerical Jacobian	14
2.4 The Levenberg-Marquardt Damping Parameter.....	14
Chapter 3 - The Finite Element Model	16
3.1 Short Beam Shear Finite Element Model.....	16
3.2 Transferring DIC data onto FE models.	18
Chapter 4 - Result and Verification.....	20
4.1 The Virtual Experiment.....	20

4.1.1 Mesh Density	22
4.1.2 Jacobian Matrix	24
4.1.3 Robustness of the Solution	27
4.1.4 The Damping Parameter and Tau	29
4.1.5 Conclusion of FEMU Algorithm from the Virtual Experiment.....	30
4.2 Working with DIC data	31
4.2.1 The Weight Matrix	33
4.2.2 Convergence of the Solution.....	34
4.2.3 Result from Five Specimens.....	38
4.2.4 Conclusion on FEMU with DIC Data	41
Chapter 5 - Streamline the Process.....	44
1. Reading the DIC .out File.....	47
2. Aligning Coordinate System.....	48
Chapter 6 - Conclusion	50
References	53

List of Figures

Figure 1. Short-beam shear experimental set up. Left: DIC system. Right: A specimen undergoes three-point bending test.....	3
Figure 2. SBS experimental schematic.	4
Figure 3. DIC full-field strain measurement.	5
Figure 4. SBS specimen fail in tension.	5
Figure 5. Stitching structure of an NCF composite from cross-sectional X-ray CT image.....	7
Figure 6. SBS specimens of unstitched NCF composite.	7
Figure 7. Finite Element Model Updating Schematic.	10
Figure 8. Convergence of the solution.	11
Figure 9. SBS finite element model in Abaqus.	16
Figure 10. Boundary condition at supports.	17
Figure 11. Boundary conditions of the beam.	17
Figure 12. Top: DIC data. Bottom: DIC interpolated data on FEM grids.	19
Figure 13. An approximated nonlinear stress-strain curves of a DIC raw data and a DIC interpolated data.	19
Figure 14. Convergence of the solution at each iteration.....	20
Figure 15. Area where strain data are used for interpolation.	21
Figure 16. Four SBS FE models. Top left: coarse mesh, Top right: medium mesh, Bottom left: fine mesh, Bottom right: extra-fine mesh.	22
Figure 17. Convergence of the solution from analytical Jacobian.	23
Figure 18. Convergence of the solution from both Jacobians. Top: Numerical Jacobian. Bottom: Analytical Jacobian.....	24
Figure 19. RMS strain error of analytical and numerical Jacobian.	26

Figure 20. Convergence of the solution from four different initial approximations.	27
Figure 21. Short-beam shear finite element model with 15,840 elements.....	31
Figure 22. Approximated nonlinear stress-strain curves from two data interpolations.....	32
Figure 23. Top: DIC interpolated strain data on FEM grids.	
Bottom: DIC strain measurement on a specimen.....	32
Figure 24. Left: DIC data from specimen 5 contains less noise.	
Right: DIC data from specimen 3 contains lots of noise.....	33
Figure 25. Nonlinear stress-strain curves of the solution from analytical and numerical Jacobian.....	36
Figure 26. Convergence of the solution from three different models.....	36
Figure 27. Convergence of the solution from two different initial approximations.....	37
Figure 28. Nonlinear stress-strain curves from all five specimens.....	38
Figure 29. Magnitude of maximum positive and negative strain vs step.	40
Figure 30. Top: Uneven strain distribution on specimen 3.	
Bottom: Even strain distribution on specimen 5.....	40
Figure 31. Top: RMS strain error of five specimens.	
Bottom: RMS strain error from virtual testing.....	42
Figure 32. VIC3D 9 Interface.....	44
Figure 33. Old Process: Exporting CSV file, reading CSV file and aligning coordinate system	45
Figure 34. New process: Reading output file and aligning coordinate using Python.....	46
Figure 35. Top: DIC x-coordinate before the adjustment.	
Bottom: DIC x-coordinate after the adjustment.....	47

List of Tables

Table 1. Converge of the material shear parameters.....	21
Table 2. Convergence of shear parameters from the analytical Jacobian.....	25
Table 3. Convergence of shear parameters from the numerical Jacobian.....	25
Table 4. Convergence of shear parameters from 80% higher initial approximation.....	28
Table 5. Convergence of G13 from different tau value.....	29
Table 6. A converged solution from different weight matrix.....	34
Table 7. Convergence of shear parameters using DIC data and the analytical Jacobian.....	35
Table 8. Convergence of shear parameters using DIC data and the numerical Jacobian.....	35
Table 9. Convergence of shear parameters from two different initial approximations.....	37
Table 10. Convergence of shear parameters from all five specimens with the average value and coefficient of variation of shear parameters.....	39

Abstract

Composite materials have been a key breakthrough in developing aircraft structures. Their superior strength and stiffness-to-density ratio, which are unmatched by no other materials, allow engineers to create high strength and ultra-lightweight aircraft structures. Recently, non-Crimp Fabric (NCF) composites have emerged as attractive alternatives to traditional autoclave pre-impregnated composites allowing for lower production costs, better handling and improved shelf life while maintaining excellent in-plane mechanical properties compared to other types of textile composites. NCF out-of-plane mechanical properties can also be enhanced by stitching a high tensile strength yarn throughout their entire laminate thickness, improving the resistance to delamination. However, stitch-reinforced NCF materials remain complex multiscale materials and the development of constitutive models able to accurately capture deformation and failure mechanisms in a virtual design platform is required before they can be utilized effectively in more aerospace applications. Accordingly, this thesis work is part of an on-going research effort at UTA's Advanced Materials and Structures Laboratory (AMSL) focusing on the development of a methodology for characterization of stitched-reinforced NCFs. In particular, characterization of the constitutive shear properties of unstitched NCF using a Short-Beam Shear (SBS) data driven method developed at AMSL is considered. The approach implements a Finite Element Model Updating (FEMU) method that uses full-field strain data obtained from Digital Image Correlation (DIC) measurements to capture nonlinear shear properties. A commercial FEA software, Abaqus, is utilized to perform the finite element analysis and a Python code is implemented for data processing as well as for conducting optimization. Several factors that may affect the value of material inputs generated are also studied, including mesh density, accuracy of the Jacobian matrix and initial approximation of material properties. Finally, the practicality of a new feature available in the latest version of the DIC software used in this work that may streamline the FEMU process is investigated.

Chapter 1

Introduction

This thesis work is part of an on-going research effort at UTA's Advanced Materials and Structures Laboratory (AMSL) in characterizing nonlinear material properties of stitched-reinforced non-crimp fabric (NCF) composites using a methodology that combine a Short-Beam Shear (SBS) test and a Finite element model updating (FEMU) method. The FEMU process will first be tested with the strain data generated from the virtual experiment. Later, the algorithm will be used to characterize nonlinear shear properties in 1-3 direction of five unstitched NCF composite specimens using full-field strain data from the Digital Image Correlation (DIC) system. The finite element analysis (FEA) will be conducted in a commercial software, Abaqus, and Python will be implemented for FEMU optimization process.

One of the objectives in this work is to study factors that may affect the solution of FEMU which included but are not limited to mesh density, the accuracy of Jacobian matrix and the initial approximation of material parameters. The study will be done on both data from a virtual experiment and from the DIC system. Lastly, the VicPy Python module from VIC3D 9 software that could potentially speed up the entire FEMU process is investigated.

In the following sections, finite element model updating, a short-beam shear experiment and non-crimp fabric composites will be reviewed in order to give readers a thorough background of this work. In addition, The FEMU algorithm will be discussed in detail later in chapter two.

1.1 Finite Element Model Updating

Finite element model updating is a method implemented to calibrate the finite element (FE) model in order to match the experimental results through an iterative parameter optimization. Since 1980s, FEMU has been widely used in the fields of aerospace, mechanical and civil engineering to precisely reflect physical behavior of structures and materials as a numerical model [1]. Some of the FEMU applications are predicting the dynamics of structures and characterizing material properties. The technique can be executed by conducting an experiment on a specimen, creating an FE model with predictive material parameters, and correlating the experimental result with the FE simulation to find the difference in parameters. Later, FE models are regenerated repeatedly with updating parameters until the difference between the simulation and the experimental data is less than a user-defined threshold. This results in an accurate FE model representing the behavior of material. Although, FEMU is a convenient method, it only provides an approximation of the solution and requires computational power to obtain an accurate result. In this work, FEMU will be applied to characterize the constitutive nonlinear shear properties of an unstitched non-crimp fabric composite. The FEMU process will be utilized to iteratively minimize the strain error between the FEA and the DIC data using least square optimization. The material shear properties will be updated in each iteration until the change in parameters is less than 0.1%.

1.2 Short-Beam Shear Experiment

A short-beam shear is an experiment aimed to determine interlaminar shear strength of high-modulus fiber-reinforced composites. The American Society for Testing and Materials (ASTM) describes a short-beam shear testing in ASTM D2344 as a three-point bending test on a specimen with uniform rectangular cross section. The recommended width and length of a specimen are 2 x thickness and 6 x thickness respectively [2]. However, this conventional standard fails to produce

a pure shear condition on a specimen which could possibly fail in other modes, such as bending, leading to an inaccurate measurement of shear properties.

To eliminate this shortcoming, a new SBS test method was developed at the Advanced Materials and Structures Laboratory at the University of Texas at Arlington. By combining a modified version of ASTM D2344 SBS test and a full-field strain measurement from Digital Image Correlation (DIC) system, all three dimensions of material constitutive shear properties can be measured simultaneously with high precision [3]. Figure 1 shows the SBS experimental set up.

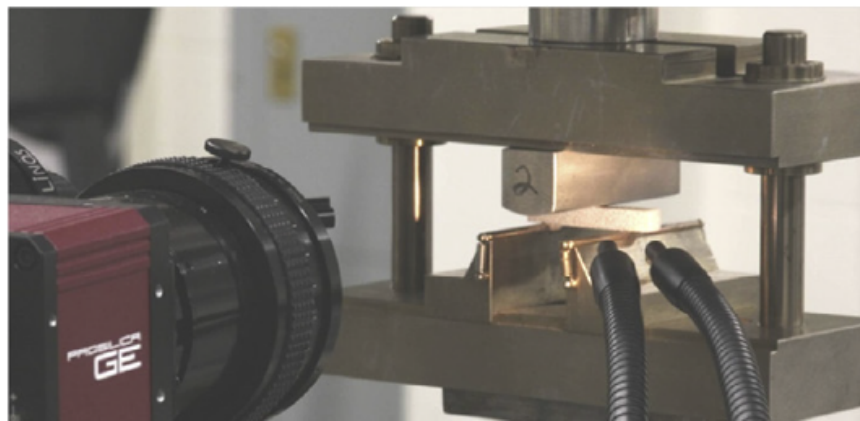


Figure 1. Short-beam shear experimental set up. Left: DIC system. Right: A specimen undergoes three-point bending test [4].

Few adjustments have been made on the SBS test of ASTM D2344. The width of the specimen is reduced to 100% - 200% of its thickness for a better strain distribution. This square cross section is also simple to be machined in 1 (0°) and 2 (90°) direction from a single unidirectional panel. The geometries of a specimen used in this work are 0.30 in. in both width and thickness, and 1.5 in. in length. Note that the specimen size is varied by using different materials. The loading nose diameter is increased to 2 – 4 in., as well as the support diameter, which is increased to 0.25 in. This reduces contact stress which could affect a pure shear condition on a specimen. The specimen is then loaded with a monotonic load of 0.005 in/min until failure occur. Figure 2 illustrates SBS test schematic.

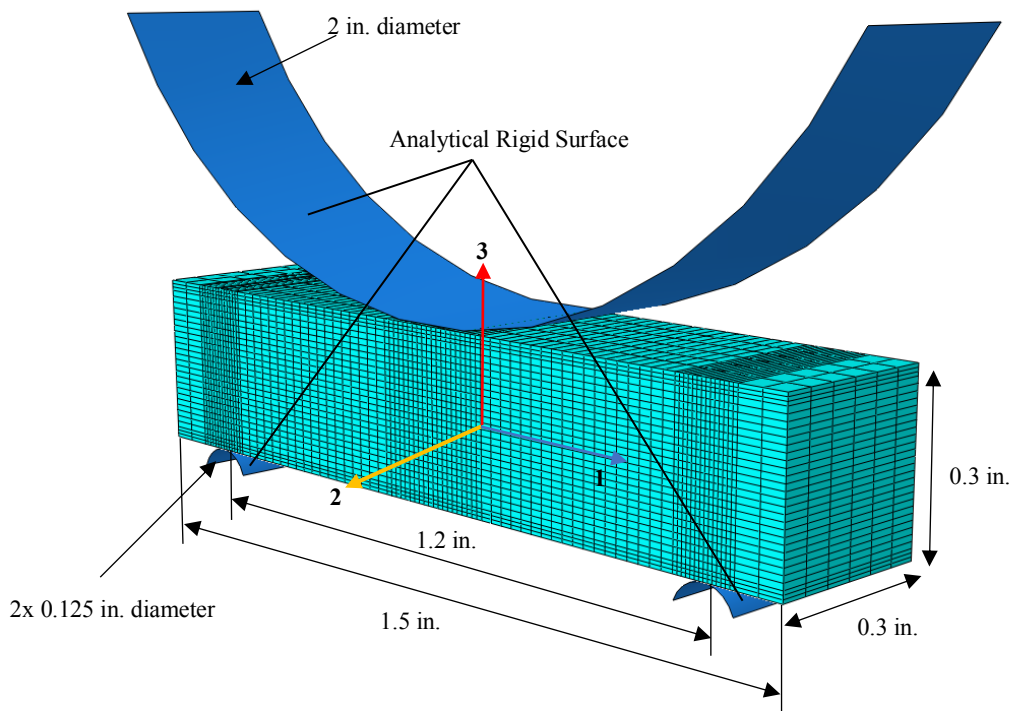


Figure 2. SBS experimental schematic.

A DIC system is utilized to measure a full-field surface strain on a specimen instead of implementing a strain gauge, because DIC provides more data point than a strain gauge. This results in more accurate strain measurement. A DIC system captures strain on all data points which created by spraying black speckles on the surface of a specimen. For each load case, the DIC system captures about 10,000 data points. Besides, DIC can simultaneously measure multiple strain components. This way, several properties can be extracted from a single experiment. Also, a full-field strain measurement is preferable for FEMU because the solution might not be unique if only one strain data point is used for the optimization.

While the SBS specimen is loaded, a sequence image is captured through a camera system at each load step. Figure 3 illustrates an example of a strain distribution from DIC system.

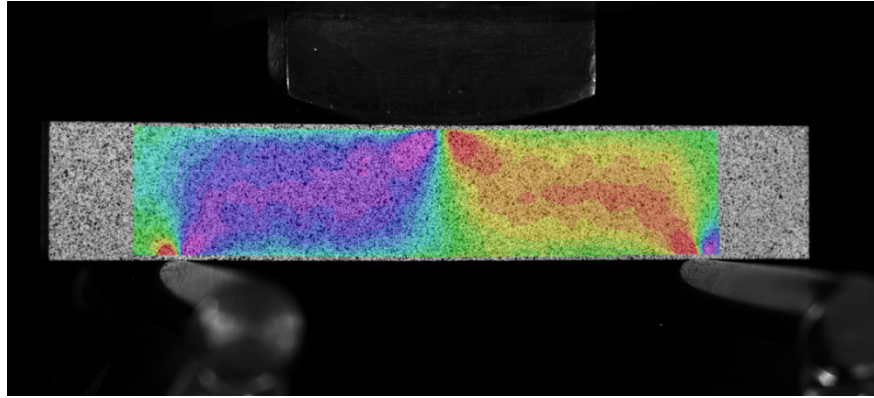


Figure 3. DIC full-field strain measurement.

Although SBS test provides ability to simultaneously capture several shear properties in one experiment, it cannot be used to evaluate shear properties in a 2-3 plane. SBS specimens made from polymer-matrix composites often fail before stress in a 2-3 plane reaches nonlinearity because fiber oriented in direction 1 has a low tensile strength in 2-3 direction. Figure 4 shows a SBS specimen machined in 90 direction breaks in a 2-3 plane. However, other test methods developed at the AMSL such as, Small Plate Twist can be implemented to measure material nonlinear shear properties in a 2-3 plane [4].

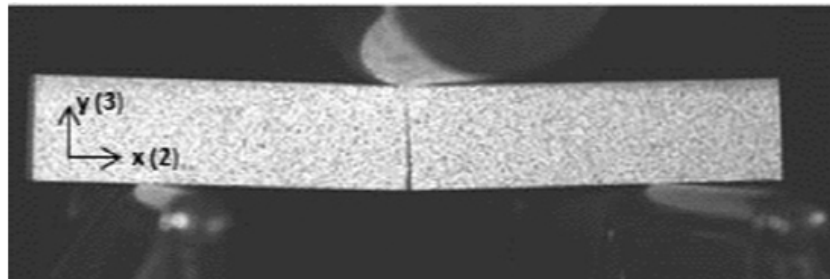


Figure 4. SBS specimen fail in tension.

1.3 Non-Crimp Fabric Composites

A composite material is a combination of two or more materials on a macroscopic scale to form a useful third material [5]. A well-engineered composite material can exhibit the best properties of several combined materials which neither of their individual components possesses. These properties include high strength, high stiffness, ultra-light weight, wear resistance, etc. Four commonly accepted classifications of composite materials are Fibrous composites, Laminated composites, Particulate composites and a Combination of some or all types.

Non-Crimp Fabric (NCF) Composites are part of fibrous composites which consist of long fibers embedded in a matrix. The word crimp refers to the difference between the actual length of fiber bundle and the length of fiber when it is inside the matrix [6]. The fibers in NCF composites are stacked in layers and stitched in position together by a non-structural thin yarn, resulting in minimal crimp and a high in-plane strength [7]. NCF composites emerged as an alternative to traditional autoclave pre-impregnated composites allowing for lower production costs, better handling and improved shelf life while maintaining excellent in-plane mechanical properties compared to other types of textile composites. Their out-of-plane properties can be enhanced further by stitching a high strength yarn through the thickness of the laminate. With a low production cost, NCF composites have a potential to become a valuable material in aerospace industry. Figure 5 shows the cross sectional cut of a stitched structure of an NCF composite from the work of Guillaume Seon on An Integrated Approach for Characterizing Non-Crimp Fabric Composites with Interlaminar Stitching Reinforcement and figure 6 shows unstitched NCF test specimens.

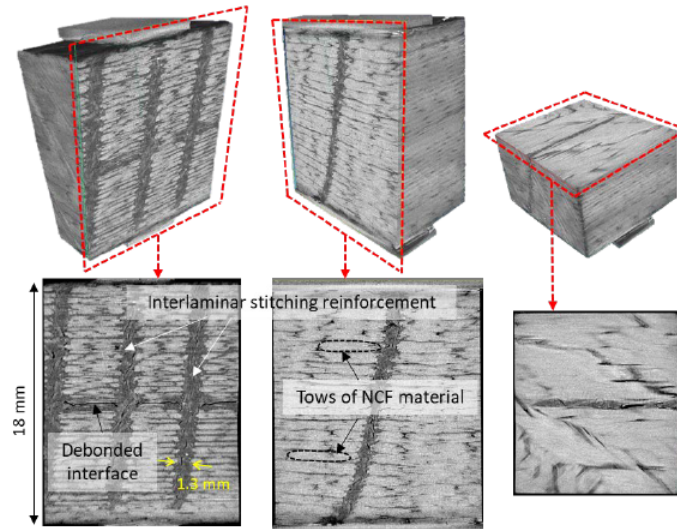


Figure 5. Stitching structure of an NCF composite from cross-sectional X-ray CT image [7, p. 3].



Figure 6. SBS specimens of an unstitched NCF composite.

Although, NCF composites offer great performance, the complexity in their macro structure makes it difficult to precisely access their constitutive parameters for a virtual design platform. Since several parameters such as, stitching pattern, yarn size and fiber orientation can affect their properties, deformation and failure analysis are a challenging task to perform on NCF composites. These impede the implementation of such material in aerospace industry and in many other applications. Therefore, more study regarding the properties of NCF composites is needed before they can be utilized effectively.

Chapter 2

Finite Element Model Updating Algorithm

2.1 The Inverse Problem

The inverse problem is described as a method used to identify material constitutive parameters. There are several techniques used to solve the inverse problem, for example, Finite Element Model Updating (FEMU), Constitutive Equation Gap Method (CEGM) and Virtual Fields Method (VFM) [3, p. 51]. In this chapter, we will focus simply on the principal of the inverse problem and the FEMU method.

Let us consider a standard finite element equation,

$$[K(\theta)]\{u\} = \{R\} \quad (1)$$

where stiffness matrix K is a function of constitutive parameters θ , u is a nodal vector and R is a force vector. Since u and R can be measured as strains and applied forces, K can be solved by inverting u , which gives,

$$[K(\theta)] = \{u\}^{-1}\{R\} \quad (2)$$

Thus, the material parameters θ , can be determined from the relationship in equation (2). This method of solving an inverse refers to as a closed-form solution. However, this approach is applicable only when there is a clear established relation between measured quantities and the constitutive parameters, such as an explicit formular. For example, when strain distribution is constant, the elastic equilibrium equation can be used to link the constitutive parameters to the measured strain and force. Often time, such test condition is not feasible. Therefore, a closed-form solution is not available, and other solving methods must be implemented.

Finite element model updating method, on the other hand, utilizes a nonlinear least square optimization procedure which minimize the objective function of the form,

$$c_u(\theta) = (\tilde{u} - u(\theta))^T \cdot W_u \cdot (\tilde{u} - u(\theta)) \quad (3)$$

Where $u(\theta)$ is a computed strain solution of u , \tilde{u} is a measured strain, and W_u is a weight matrix. The material parameters θ are then solved by an iterative numerical procedure. For this work, u represents a strain measured from a full-field DIC system and $u(\theta)$ is a computed strain obtained from an FE analysis. Equation (3) is then rewritten as,

$$c(p) = (\varepsilon^{DIC} - \varepsilon^{FEM})^T \cdot W \cdot (\varepsilon^{DIC} - \varepsilon^{FEM}) \quad (4)$$

Since only material nonlinear shear properties will be considered, p used in equation 4 is referred to as a set of unknown material constitutive shear parameters. W is a positive-definite diagonal weight matrix of the form,

$$W = [W_{ij}]_{MN} \quad \text{where } W_{ij} = \begin{cases} W_{ii} & \text{if } i=j \\ 0 & \text{if } i \neq j \end{cases} \quad (5)$$

Note that M is the number of grid points projected on the SBS specimen and N is the number of steps in which the DIC image is captured. For this work, the value of w_{ii} is equal to

$$w_{ii} = \frac{\varepsilon_{13,i}}{\varepsilon_{13,max}} \quad (6)$$

Where $\varepsilon_{13,max}$ is the maximum 1-3 strain in each load step.

The advantage of FEMU is that such method relies only on the test data and does not require any prior knowledge of material parameters. Besides, the optimization procedure is usually executed using Gauss-Newton or Levenberg-Marquardt method which is a very time-efficient method. However, the solution from the FEMU is in the form of approximation. Therefore, to obtain accurate material parameters, care has to be taken during the experimental testing and FE modeling procedure.

2.2 The Optimization Algorithm

The material constitutive shear parameters are optimized iteratively through equation (4) where its value at iteration k is expressed as,

$$p^k = p^{k-1} + \Delta p^k \quad (7)$$

and,

$$\Delta p^k = [J^T W J + \lambda^k \text{diag}(J^T W J)]^{-1} \cdot J^T W (\varepsilon^{DIC} - \varepsilon^{FEM}(p^k)) \quad (8)$$

J is the Jacobian matrix, often called as a sensitivity matrix. λ is a Levenberg-Marquardt damping parameter with more detail provided in the next topic. The whole process of the calculation can be summarized in one diagram as illustrated in figure 7.

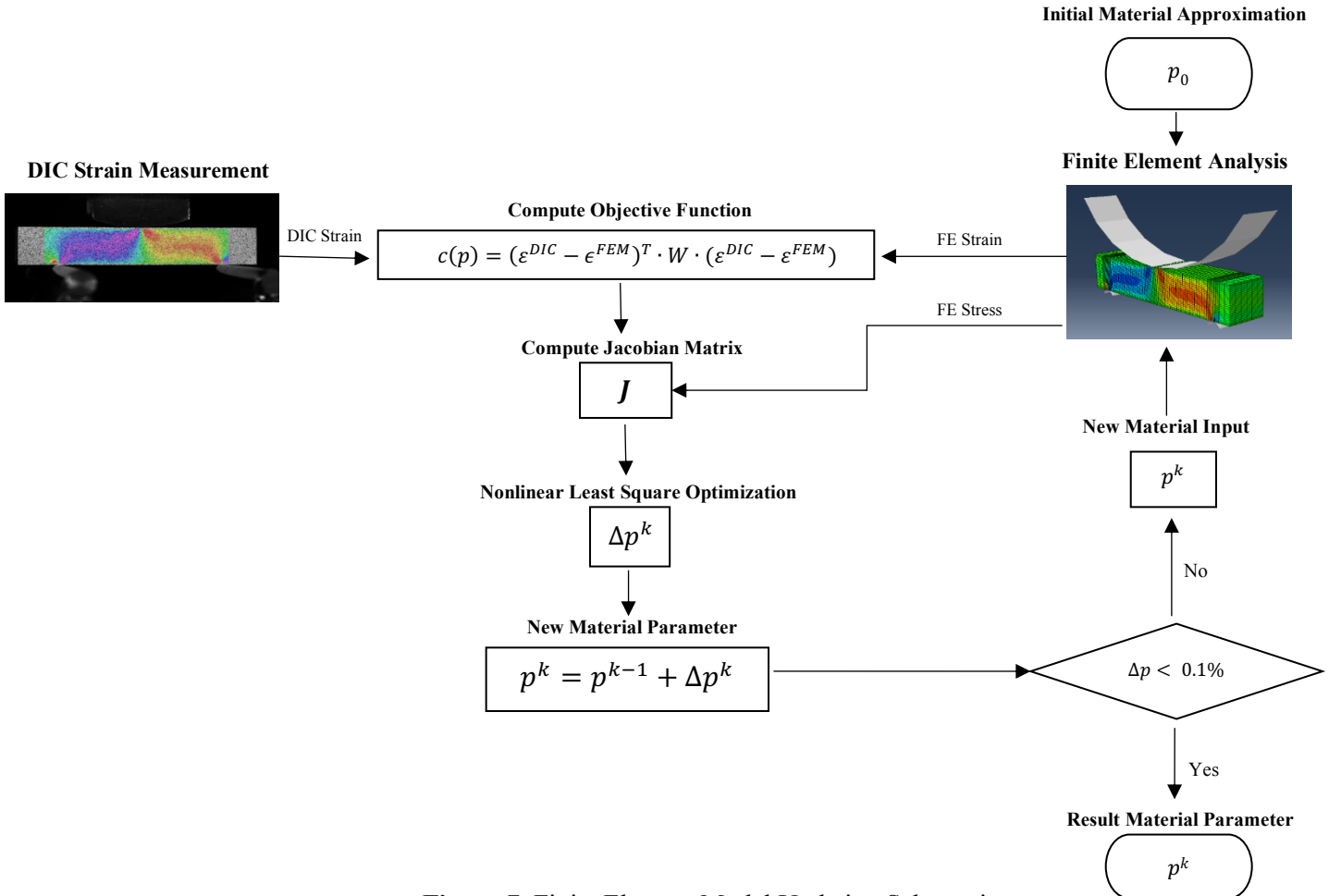


Figure 7. Finite Element Model Updating Schematic.

The initial material parameters or “approximated parameters”, p^0 , are used as input for compiling the first nonlinear FEM analysis. The FEM strain is then compared with the strain data measured from a DIC system to calculate the strain error. In order to calculate the Jacobian matrix, the FEM stress is implemented. Lastly, the change in material parameter, Δp , is calculated by using equation (8). With that, the new material parameters are updated through equation (7) and the whole process continues until the change in the input parameters is less than 0.1%. The Root Mean Square (RMS) error between DIC and FEM strain is then calculated in each iteration to determine how close the strain is to the actual strain. Figure 8 shows an example of the material parameters as the number of iterations increases. The solution converges in only two iterations. The FEMU optimization is done with Python due to its simplicity in data analysis. The fact that Abaqus supports Python scripting also makes it a very convenient choice for this work.

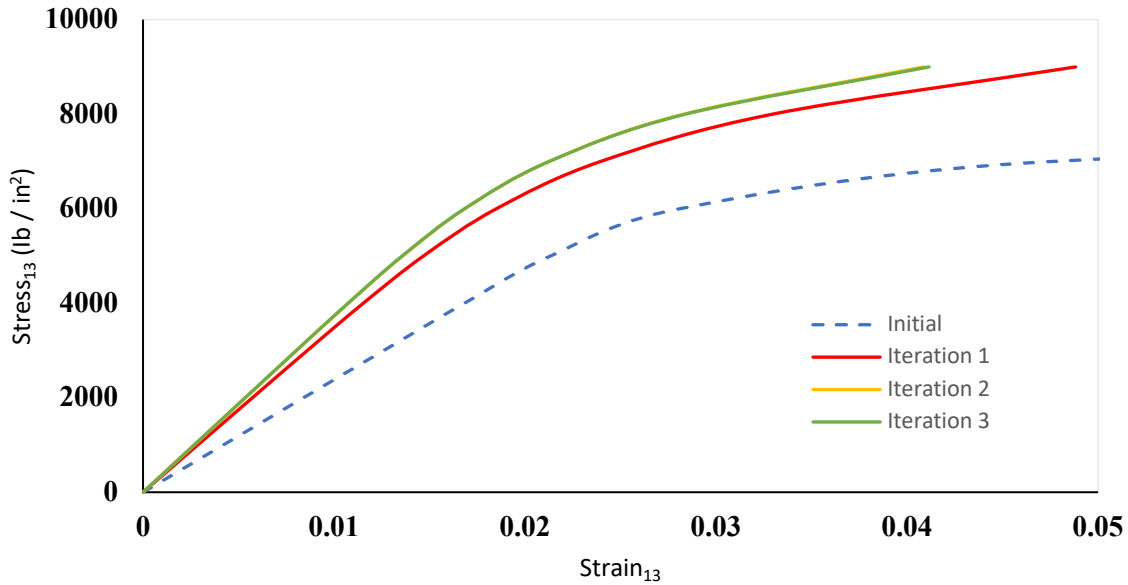


Figure 8. Convergence of the solution.

2.3 The Jacobian Matrix

In general, the Jacobian matrix is used to transform a vector-value function from one coordinate system to another coordinate system, for example, from a cartesian coordinate to a polar coordinate. At a point where a function is differentiable, the Jacobian matrix describes how the function is transforming locally near that point. In other word, the Jacobian matrix is the best linear approximation for transformation of the function near that point [8]. In this work, the Jacobian matrix is implemented to describe how the change in FEM strain compare to the change in material parameters, and is expressed as,

$$J = \left[\frac{\partial \varepsilon^{FEM}}{\partial p} \right]_{M \cdot N \times P} \quad (9)$$

The size of the Jacobian matrix is defined by the number of grid points M, the number of load steps N and the number of material parameters P. For example, for 10 grid points, 10 load steps and 3 material parameters, the Jacobian matrix will have its size of 100x3. As illustrated in figure 7, the Jacobian matrix is updated in every iteration. For this thesis, two types of the Jacobian matrix are considered, the analytical Jacobian and the numerical Jacobian.

2.3.1 The Analytical Jacobian

The material constitutive model implemented in this work is a combination of an elastic stress-strain orthotropic material model and nonlinear stress-strain Ramberg-Osgood equations to represent the shear response in the three principal material planes. The following set of equations represent the stress-strain relation of the material model.

$$\begin{bmatrix} \varepsilon_{11} \\ \varepsilon_{22} \\ \varepsilon_{33} \end{bmatrix} = \begin{bmatrix} \frac{1}{E_{11}} & \frac{-\nu_{12}}{E_{11}} & \frac{-\nu_{13}}{E_{11}} \\ \frac{-\nu_{12}}{E_{11}} & \frac{1}{E_{22}} & \frac{-\nu_{23}}{E_{22}} \\ \frac{-\nu_{13}}{E_{11}} & \frac{-\nu_{23}}{E_{22}} & \frac{1}{E_{33}} \end{bmatrix} \begin{bmatrix} \sigma_{11} \\ \sigma_{22} \\ \sigma_{33} \end{bmatrix} \quad (10)$$

$$\gamma_{12} = \frac{\tau_{12}}{G_{12}} + \tau_{12} \left(\frac{\tau_{12}}{K_{12}} \right)^{\frac{1}{n_{12}}}$$

$$\gamma_{13} = \frac{\tau_{13}}{G_{13}} + \tau_{13} \left(\frac{\tau_{13}}{K_{13}} \right)^{\frac{1}{n_{13}}}$$

$$\gamma_{23} = \frac{\tau_{23}}{G_{23}} + \tau_{23} \left(\frac{\tau_{23}}{K_{23}} \right)^{\frac{1}{n_{23}}}$$

Where ν is passion ratio, E is elastic modulus, G is linear shear modulus and K, n are the secant-intercept modulus.

The analytical Jacobian matrix is based on the assumption that the stress variation is small in the evaluating area compared to the change in material parameter. Therefore, it can take advantage of a closed-form stress derivative from equation (10). For instance, for three variable material parameters in a 1-3 plane, the analytical Jacobian can be written as,

$$J = \begin{bmatrix} \frac{\partial \varepsilon_{13}}{\partial G_{13}} & \frac{\partial \varepsilon_{13}}{\partial K_{13}} & \frac{\partial \varepsilon_{13}}{\partial n_{13}} \end{bmatrix}_{M \cdot N \times 3}$$

$$J = \begin{bmatrix} \frac{-\tau_{13}}{G_{13}^2}, & \frac{-\tau_{13}^{\frac{1}{n_{13}}}}{n_{13} \cdot K_{13}^{\frac{1}{n_{13}} + 1}}, & \frac{-1}{n_{13}^2} \cdot \left(\frac{\tau_{13}}{K_{13}} \right)^{\frac{1}{n_{13}}} \cdot \ln \left(\frac{\tau_{13}}{K_{13}} \right) \end{bmatrix}_{M \cdot N \times 3} \quad (11)$$

The analytical Jacobian provides convenience in calculation since there is no additional FE analysis required. However, the rate of convergence will depend on how stress is varied, and which material constitutive model is chosen.

2.3.2 The Numerical Jacobian

In case there is no explicit relationship between stress and strain as in equation (10), Finite Difference Method (FDM) will be implemented to calculate the Jacobian matrix. A forward difference is used in this work to approximate the derivative of strains with respect to the material parameters as shown in equation (11).

$$J = \begin{bmatrix} \frac{\partial \varepsilon_{13}}{\partial G_{13}} & \frac{\partial \varepsilon_{13}}{\partial K_{13}} & \frac{\partial \varepsilon_{13}}{\partial n_{13}} \end{bmatrix}_{M \cdot N \times 3}$$

$$J = \begin{bmatrix} \frac{\varepsilon_{13,FEM2} - \varepsilon_{13,FEM1}}{\Delta G_{13}}, & \frac{\varepsilon_{13,FEM2} - \varepsilon_{13,FEM1}}{\Delta K_{13}}, & \frac{\varepsilon_{13,FEM2} - \varepsilon_{13,FEM1}}{\Delta n_{13}} \end{bmatrix}_{M \cdot N \times 3} \quad (12)$$

FEM₁ and FEM₂ refer to the number of simulation (job) run in Abaqus software. FEM₁ uses the material parameter at iteration k to run the simulation while FEM₂ uses material at the same iteration but with the change in its value by 1%. For example, if $G_{13} = 737,000$ psi is used as a material parameter to run FEM₁, $G_{13} = 744,370$ will be used to run FEM₂. The disadvantage of using numerical Jacobian is that several jobs must be compiled in each iteration which is computationally expensive. The rate of the solution convergence from using the analytical and numerical Jacobian will be thoroughly examined later in chapter 4.

2.4 The Levenberg-Marquardt Damping Parameter

In this work, a least square optimization problem is solved iteratively with Levenberg-Marquardt method. The method combines Gauss-Newton method and the gradient descent method together. In Gauss-Newton method, the algorithm minimizes the sum of the squared errors through an assumption that the least square function is locally quadratic in the parameters [9]. This method requires no calculation of a second derivatives, unlike regular Newton method, and converges faster than a gradient descent method in a moderate size problem. On the other hand, the gradient

descent method converges much faster in a small size problem since it minimizes the objective function in the steepest-descent direction.

Levenberg-Marquardt method takes advantage of both Gauss-Newton and gradient descent method by introducing a damping parameter λ , which will vary the parameter update between Gauss-Newton and gradient descent method. The method acts like a Gauss-Newton when the parameters are close to the solution and acts like a gradient descent method when the parameters are far away from the solution. With small damping value results in Gauss-Newton update and large damping value result in gradient descent update, the Levenberg-Marquardt proves to be a computational effective method.

The damping parameter will get updated in each iteration to maximize the convergence rate by comparing the objective function of the current iteration to the previous one. The gain factor, Q , is used as a mean to compare the objective function. Where Q at each iteration is expressed as,

$$Q^k = \frac{c^{k-1} - c^k}{2\Delta p^{kT} (\lambda_k \Delta p^k + J^T W (\varepsilon^{DIC} - \varepsilon^{FEM}(p^k)))} \quad (13)$$

The initial value of Levenberg-Marquardt damping parameter, λ , is approximated by,

$$\lambda = \tau \times \max (\text{diag}(J^T W J)) \quad (14)$$

Where τ is equal to 0.001 [10]. The damping parameter is updated in the following conditions.

1. $\lambda = \lambda/3.0$ if Q in that iteration is greater than 0.75.
2. $\lambda = \lambda \times 2.0$ if Q in that iteration is less than 0.25.
3. $\lambda = 10.0$ if λ is greater than 10.0.

Another advantage of implementing the damping parameter in the least square optimization algorithm is that it reduces the effect of noise from the experimental data, which improves the accuracy of the resulting parameters.

Chapter 3

The Finite Element Model

3.1 Short Beam Shear Finite Element Model

In order to obtain computational stress and strain values, finite element simulation is conducted in the commercial software Abaqus 2020. Surface stress and strain captured on the model will be utilized to calculate a Jacobian matrix and optimize the constitutive shear 1-3 properties. Figure 9 illustrates a SBS finite element model. The model consisted of 15,480 elements with more element in contact areas for more accurate solutions.

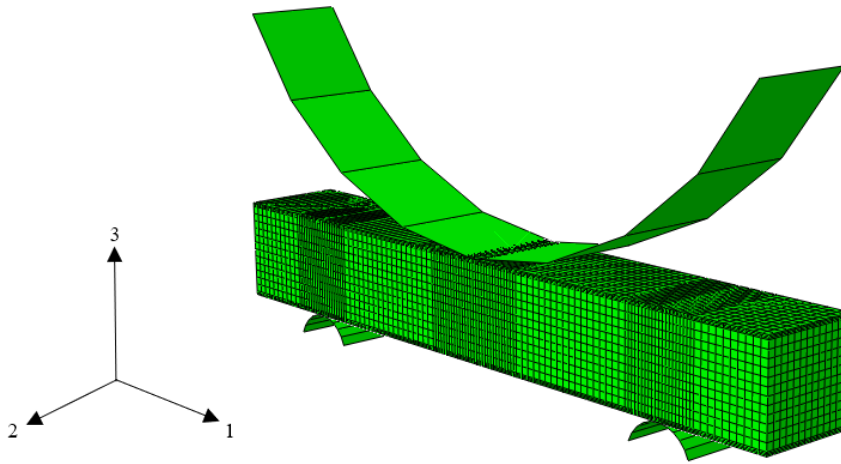


Figure 9. SBS finite element model in Abaqus.

C3D8I, 3D incompatible brick element, is used in the SBS model since it gives best result in rectangular shape and removes shear locking in bending applications. Although C3D8R, 3D reduce integration brick element, can be used interchangeably, it provides less accurate results due to its limited integration point. Analytical rigid surfaces are used to create loading nose and two

supports with frictionless contact at contact points. The FEM analysis considers both the effect of material and geometrical nonlinearities using the option `nlsgeom=yes` in the definition of the Abaqus Implicit Static Analysis step.

For boundary conditions, reference points of both supports are encastre ($u_1 = u_2 = u_3 = u_{r1} = u_{r2} = u_{r3} = 0$) as shown in figure 10. The reason that the whole contact area between the beam and supports is not fixed is that the beam is allowed to slide in both 1 and 2 directions.

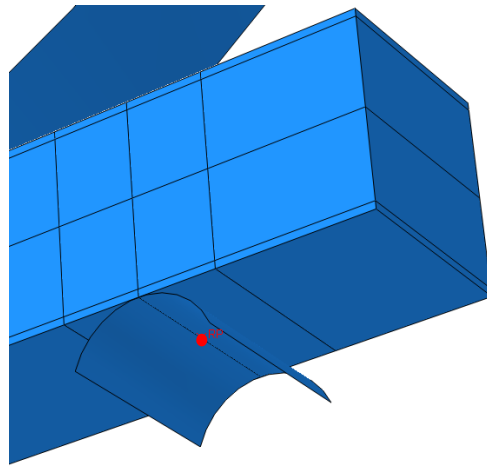


Figure 10. Boundary condition at supports.

The mid plane of the beam is fixed with $u_1 = 0$ and $u_2 = 0$ at the square and rectangular cross section respectively to retain rigid body motion as shown in figure 11.

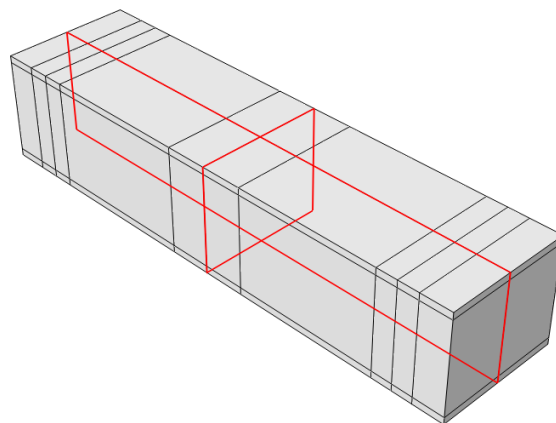


Figure 11. Boundary conditions of the beam.

A concentrate load is assigned at the loading nose and the magnitude of the load comes from the DIC test data.

A user subroutine, UMAT, for shear nonlinear material developed by the ASML is implemented for describing material behavior of the beam in which the initial material parameters are

$$E_{11} = 7.56 \times 10^6, E_{22} = 7.06 \times 10^6, E_{33} = 1.26 \times 10^6 \text{ psi}$$

$$\gamma_{12} = \gamma_{13} = \gamma_{23} = 0.40 \text{ psi}$$

$$G_{12} = 2.77 \times 10^6, G_{13} = 3.7 \times 10^3, G_{23} = 4.4 \times 10^5 \text{ psi}$$

$$K_{12} = 4.17 \times 10^5, K_{13} = 1.8 \times 10^4, K_{23} = 36,100 \text{ psi}$$

$$n_{12} = 0.273, n_{13} = 0.15, n_{23} = 0.201 \text{ psi}$$

The fiber direction is 1, the in-ply traverse direction is 2 and laminate thickness direction is 3.

3.2 Transferring DIC data onto FE models.

The VIC3D 7 software is utilized to calculate a strain distribution from a DIC system. In order to correlate the test data with FEA result, the origin of the DIC and FEM coordinate has to be at the same location. Therefore, all DIC test data must be shifted in both X and Y direction until it coincides with the FEM grid. In this work, rotation of the DIC data to align with the FEM reference coordinate system was not needed. The left support and the mid-point on the beam are used as a reference for shifting coordinate in x and y direction, respectively. For example, if the left support has the FEM coordinate of $x = -0.6$, but on DIC coordinate $x = 0.5$, then all the data points in DIC have to be subtracted by 1.1. The same goes with the y coordinate.

The function `griddata` from Python's SciPy library is used to interpolate the DIC data onto each node of the FEM mesh. Figure 12 compares the DIC raw data with the DIC interpolated data on FEM grids.

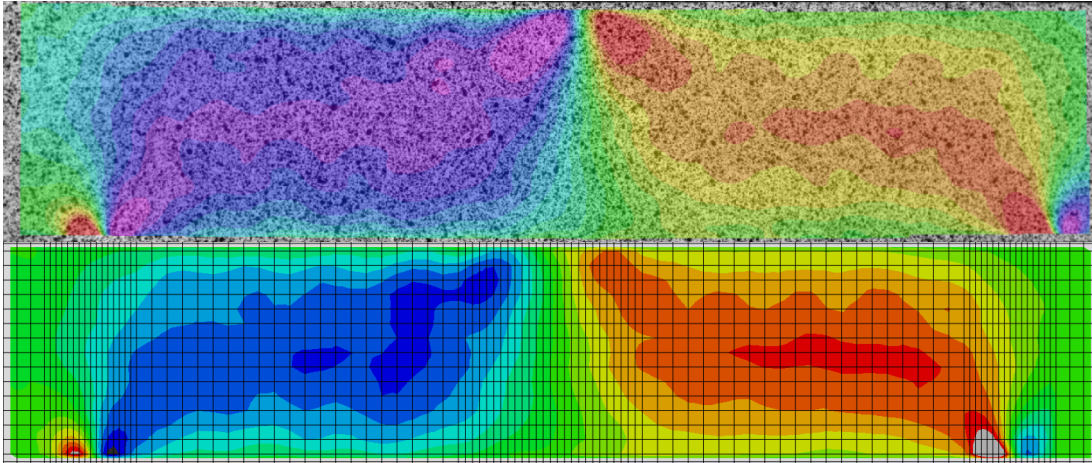


Figure 12. Top: DIC data. Bottom: DIC interpolated data on FEM grids.

To verify the accuracy of the interpolated data, nonlinear stress-strain curves of both data are plotted together as illustrated in figure 13. A good interpolation will result in an identical stress-strain curve of both graphs. The mesh density as well as the shift in data coordinate will affect the interpolated result. Therefore, care has to be taken during the coordinate adjustment.

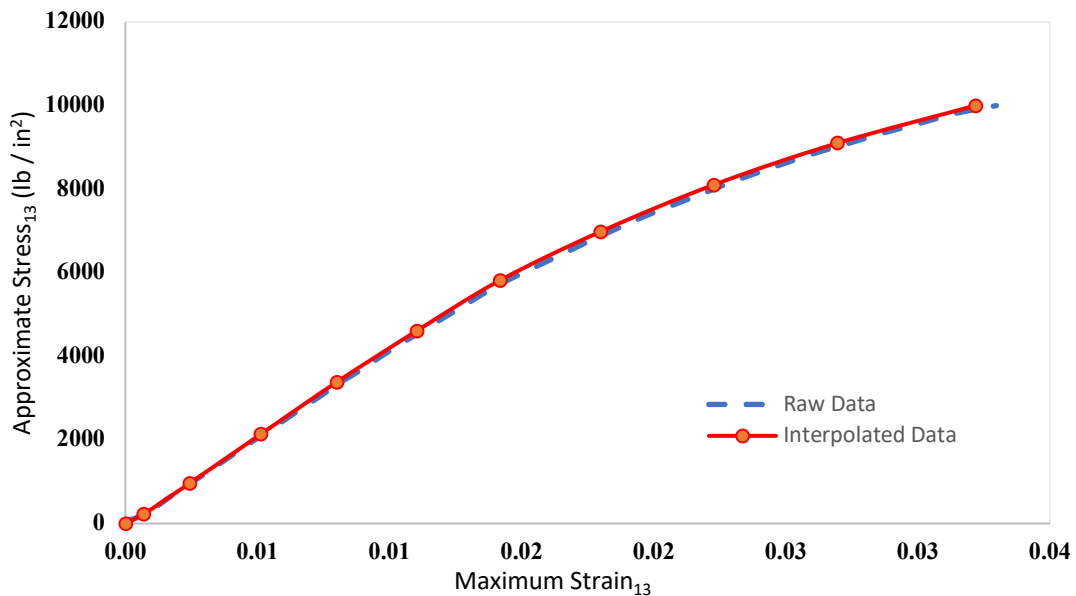


Figure 13. An approximated nonlinear stress-strain curves of a DIC raw data and a DIC interpolated data.

Chapter 4

Results and Verification

4.1 The Virtual Experiment

To verify the FEMU least square optimization algorithm, two FE SBS models are created. One model was created to replicate the physical experiment using material properties in 1-3 direction that were 40% higher than the other one. The second model was used to conduct FEMU and would be calibrated so that it matched with the material properties of the first model which were, $G_{13} = 374,000$, $K_{13} = 15,400$ and $n_{13} = 0.132$. Figure 14 and table 1 illustrate the convergence of the material shear parameter as the number of iterations increases.

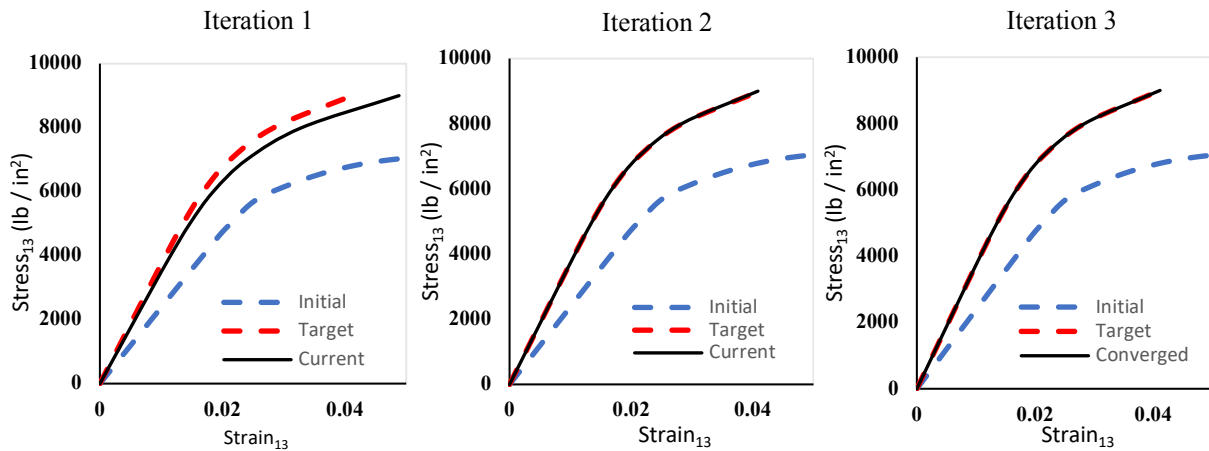


Figure 14. Convergence of the solution at each iteration.

Iteration / Parameters	G_{13} (psi)	K_{13} (psi)	n_{13} (psi)	RMS strain error (%)
Initial	238,000	9,800	0.084	17.66%
1	349,146	15,355	0.141	3.13%
2	373,216	15,457	0.132	0.07%
3	374,003	15,402	0.132	0.001%
Real parameter	374,000	15,400	0.132	

Table 1. Converge of the material shear parameters.

The process took about 10 minutes using 8 CPUs for 3 iterations. The root mean square (RMS) strain error is also calculated in each iteration to measure how close the strain is to the actual strain value.

It is important that the generated DIC data and the FEA result are evaluated at the same node points. As well as the order of the node in the weight and Jacobian matrices must be in the same order. Otherwise, the optimization will be invalid. Also, only the positive shear strain is used to calculate the weight matrix and Jacobian matrix. The strain data of each node that lie within the black window as shown in figure 15 are used for the FEMU process.

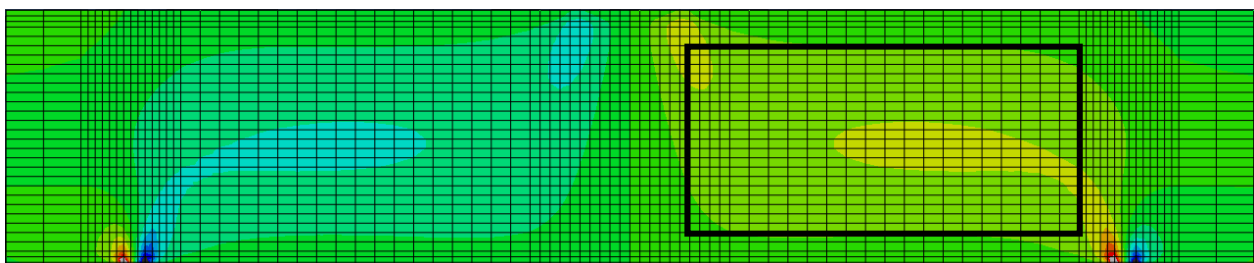


Figure 15. Area where strain data are used for interpolation.

The sensitivity of the model including mesh density, Jacobian matrix of transformation and the initial approximation of material properties are examined in the following section.

4.1.1 Mesh Density

Four SBS models meshed with the same element, C3D8I, but different mesh density were used to run the FEMU procedure. The coarse mesh model has 624 elements, the medium mesh model has 1,520 elements, the fine mesh model has 2,016 elements, and extra-fine mesh model has 4,704 elements. All four models used both analytical Jacobian with the initial approximation of $G_{13} = 340,000$, $K_{13} = 14,000$ and $n_{13} = 0.12$. Figure 16 shows all four FE models.

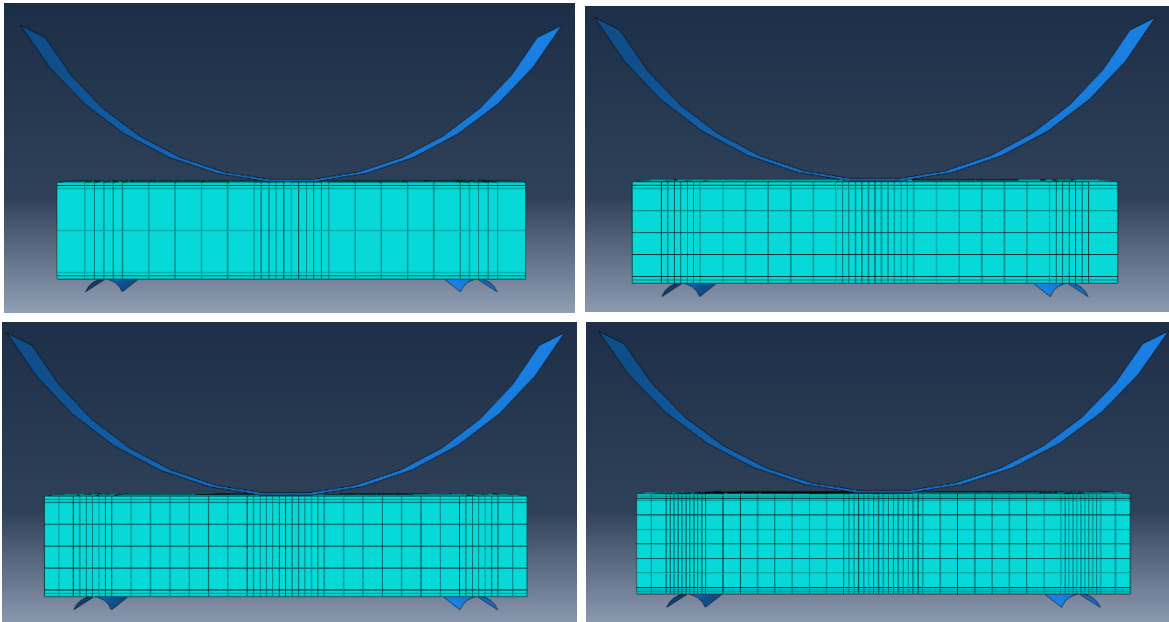


Figure 16. Four SBS FE models. Top left: coarse mesh, Top right: medium mesh, Bottom left: fine mesh, Bottom right: extra-fine mesh.

After getting the material parameters from the optimization procedure, the nonlinear stress strain curves are plotted using Ramberg-Osgood formula to compare the solution convergence from different mesh. Figure 17 shows that the solution from all mesh density converge to the real value.

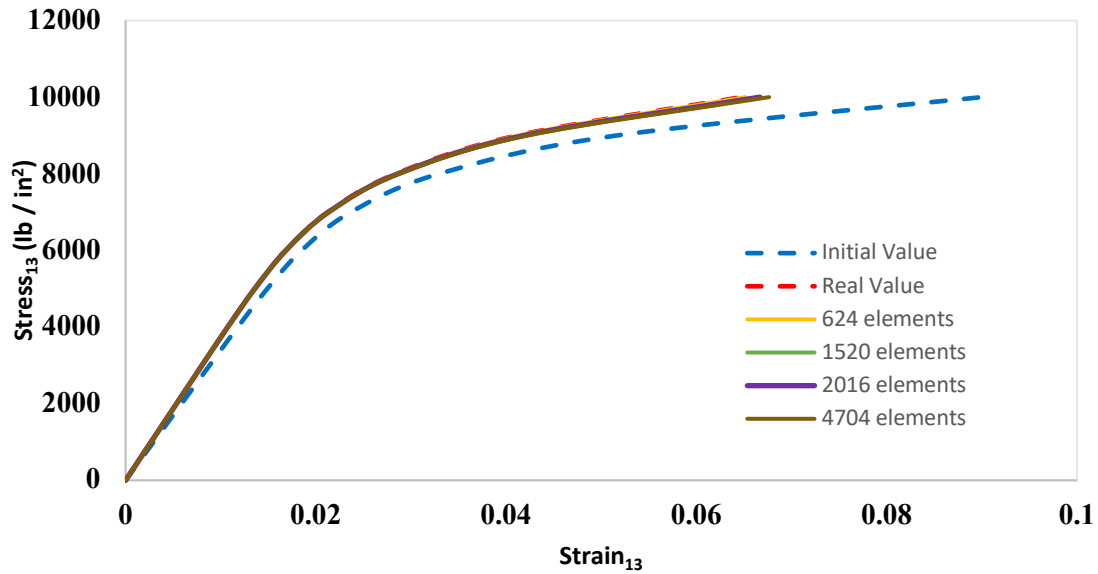


Figure 17. Convergence of the solution from analytical Jacobian.

Notice that even a 624-element-model converged to the real solution. Besides, the mesh density seems to have very little effect on the convergence of the FEMU solution. This means that the FEMU algorithm is very robust and using few elements will reduce the FEMU calculation time. The 4704-element-model took 10 minutes to complete the optimization while the 624-element-model only required 3 minutes.

4.1.2 Jacobian Matrix

All four SBS FE models were also tested with the numerical Jacobian for the optimization. The analytical Jacobian is calculated using the stress approximation while the numerical Jacobian uses the FDM with forward operator for the calculation. The solution converged using numerical Jacobian as shown in figure 18, top. However, the solution from numerical Jacobian converged faster than the analytical Jacobian and more accurate solution can be obtained from using a numerical Jacobian. Table 2 and 3 compare the solution from using both analytical and numerical Jacobian.

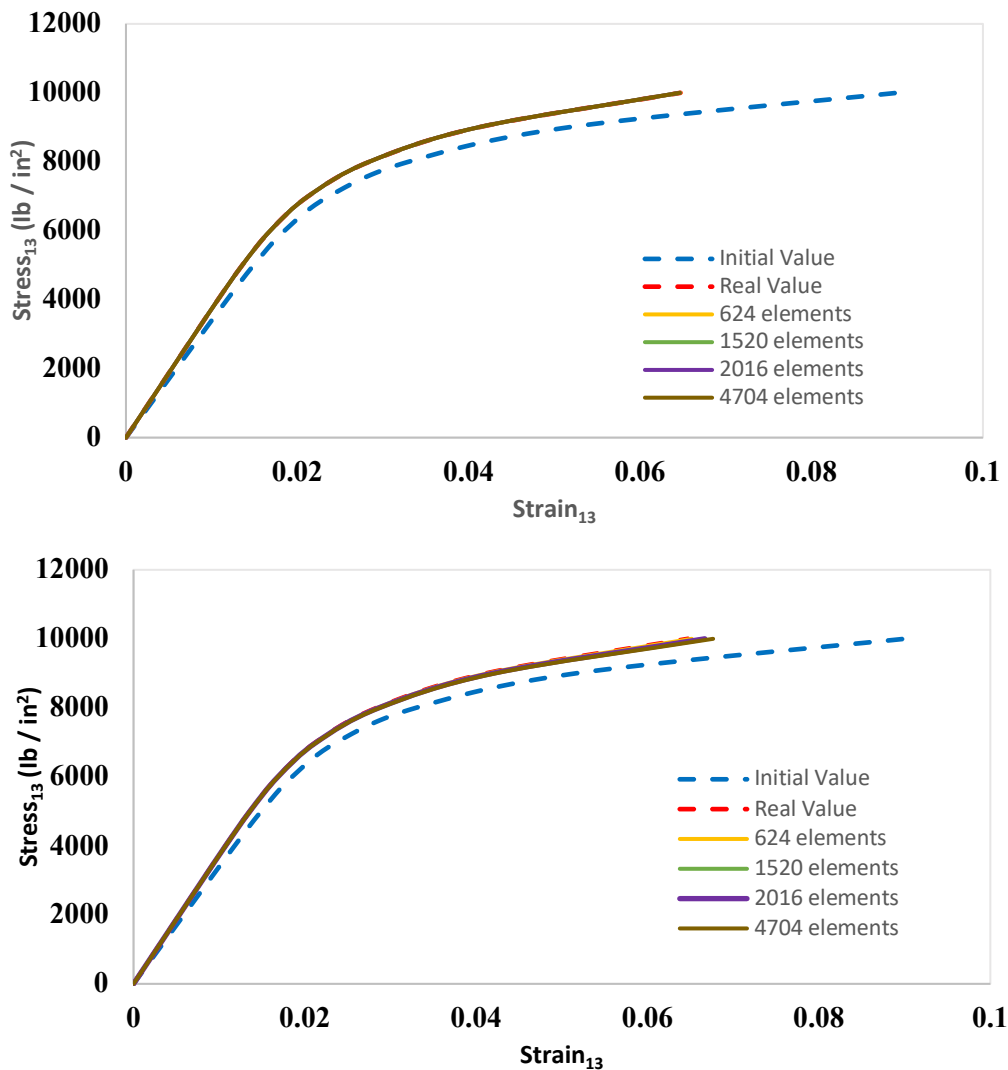


Figure 18. Convergence of the solution from both Jacobians. Top: Numerical Jacobian. Bottom: Analytical Jacobian

Iteration / Parameters	G_{13} (psi)	K_{13} (psi)	n_{13} (psi)	RMS strain error (%)
Initial	238,000	9,800	0.084	17.66%
5	355,558	12,007	0.112	3.72%
10	371,387	13,219	0.114	1.12%
15	373,309	14,447	0.124	0.34%
Real value	374,000	15,400	0.132	

Table 2. Convergence of shear parameters from the analytical Jacobian.

Iteration / Parameters	G_{13} (psi)	K_{13} (psi)	n_{13} (psi)	RMS strain error (%)
Initial	238,000	9,800	0.084	17.66%
1	349,146	15,355	0.141	3.13%
2	373,216	15,457	0.132	0.07%
3	374,003	15,402	0.132	0.001%
Real value	374,000	15,400	0.132	

Table 3. Convergence of shear parameters from the numerical Jacobian.

According to tables 2 and 3, numerical Jacobian only required 3 iterations for the solution to converge. In fact, the material property obtained is almost identical to the real material value. The analytical Jacobian gave a good result as well with 0.34% RMS strain error. However, it took 15 iterations to get to the solution and will need more to obtain the exact value. Figure 19 shows a root-mean-square (RMS) error as the number of iteration increase.

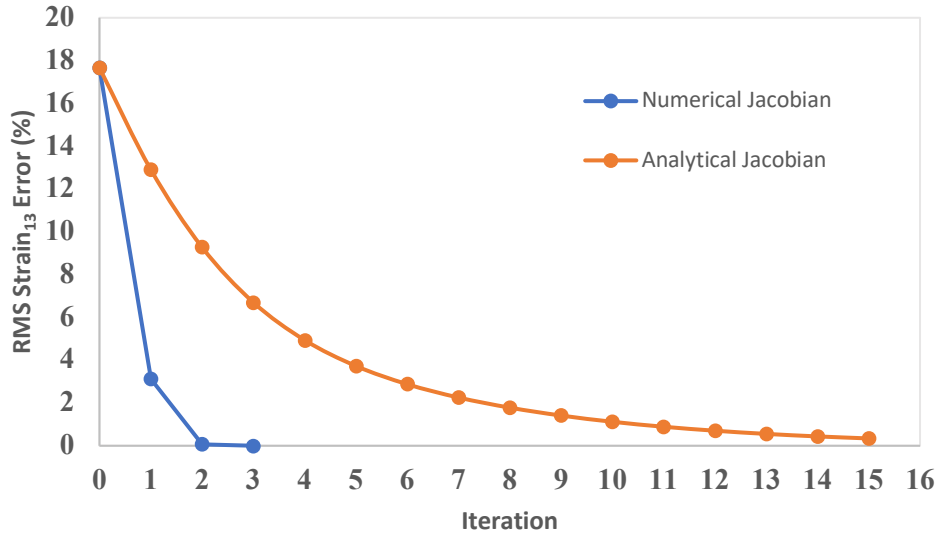


Figure 19. RMS strain error of analytical and numerical Jacobian.

A related point to consider is that the number of Abaqus jobs that need to be compiled using numerical Jacobian were 12 jobs and not 3. Since it took 3 extra jobs to calculate the numerical Jacobian in each iteration, the additional 9 jobs were added. Because the analytical Jacobian required no extra jobs to run, the number of jobs are equal to the number of iterations which is 15. In conclusion, in this case, the numerical Jacobian used less job while giving a better result. This is because the analytical Jacobian relies on a closed-form stress approximation. If the stress is not constant, more iterations are needed for the optimization. On the other hand, numerical Jacobian is calculated directly based on the difference between the finite strain of the two model which resulted in a better accuracy.

4.1.3 Robustness of the Solution

Since the solution is more accurate and converges faster using a numerical Jacobian, four different sets of initial material parameters were tested with the numerical Jacobian and the convergence of the solution was examined. The four sets of initial material parameters were,

1. $G_{13} = 238,000, K_{13} = 9,800, n_{13} = 0.08$.
2. $G_{13} = 340,000, K_{13} = 14,000, n_{13} = 0.12$
3. $G_{13} = 408,000, K_{13} = 16,800, n_{13} = 0.14$.
4. $G_{13} = 510,000, K_{13} = 21,000, n_{13} = 0.18$.

The real value of the material parameter was $G_{13} = 370,000, K_{13} = 15,400$ and $n_{13} = 0.132$. All four models took 4 iterations for the solution to converge, and results are shown in figure 20 below.

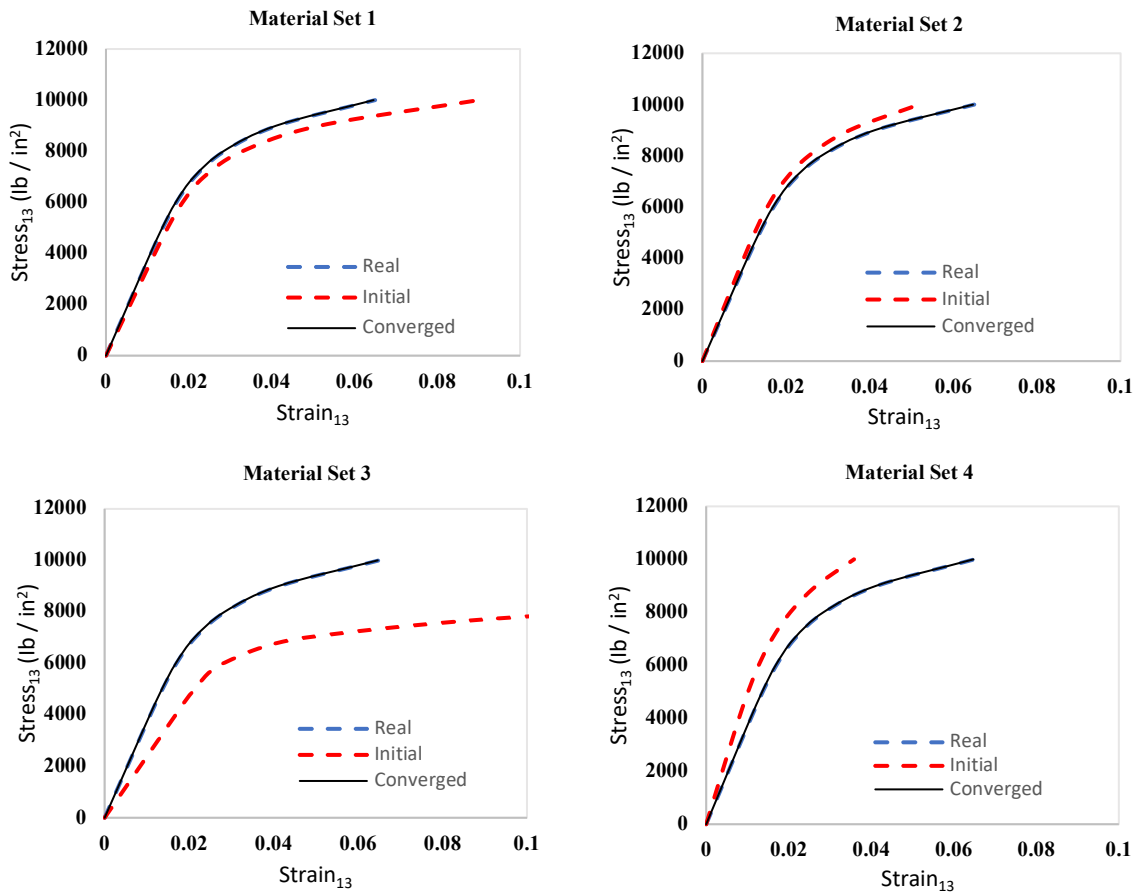


Figure 20. Convergence of the solution from four different initial approximations.

The initial approximations of material parameters that range from 40% lower to 40% higher than the actual parameter converged to the real solution with less than 0.1% RMS error using numerical Jacobian and a 2016-element-model. The solution still converged even when the initial set of material parameter was 80% higher than the actual parameters, but it would take more calculation time. Table 4 shows the convergence of the solution of the initial set of parameters that are 80% higher than the actual parameter. The process took 6 iterations. However, if the initial set of input parameters are too extreme, Abaqus will not be able to compile the job and the FEMU is not achievable. This means that, with the FE generated data, the solution will always converge using numerical Jacobian as long as the job is executable from the initial set of parameters. Also, the optimum value of initial parameters should be in the range of 40% lower to 40% higher than the actual parameter.

Iteration / Parameters	G_{13} (psi)	K_{13} (psi)	n_{13} (psi)	RMS strain error (%)
Initial	646,000	26,600	0.228	18.78%
1	262,923	19,506	0.184	13.94%
2	359,610	10,055	0.106	7.33%
3	379,101	15,350	0.164	3.41%
4	372,673	17,597	0.161	0.74%
5	373,190	15,360	0.132	0.17%
6	373,999	15,404	0.132	0.006%
Real value	374,000	15,400	0.132	

Table 4. Convergence of shear parameters from 80% higher initial approximation.

4.1.4 The Damping Parameter and Tau

As discussed in chapter 2, the initial value of Levenberg-Marquardt damping parameter, λ , is approximated by,

$$\lambda = \tau \times \max(\text{diag}(J^T W J)) \quad (13)$$

Where τ is a small such as 0.001. In some cases, if the mesh density is not high enough the value of tau, τ , has to be even lower than 0.001 for the solution to converge to the correct value. Table 5 shows the convergence of the solution using different value of tau. The model used 1,520 elements and was tested with both numerical and analytical Jacobian with the initial value of G_{13} equal to 737,000.

τ / Parameter	G_{13} Analytical (psi)	G_{13} Numerical (psi)
0.1	773,493	810,700
1×10^{-3}	801,083	810,700
1×10^{-6}	807,719	810,700
1×10^{-10}	807,723	810,700
Real Value	810,700	810,700

Table 5. Convergence of G_{13} from different tau value.

For analytical Jacobian, the value of tau should be less than 1×10^{-6} for the most accurate result, if the lower density mesh will be used. However, tau cannot be equal to 0. Otherwise, μ , λ , will be 0 and will not get updated. On the other hand, the value of tau has no effect on the numerical Jacobian. The solution will always converge no matter how large the value is.

4.1.5 Conclusion of FEMU Algorithm from the Virtual Experiment

The FEMU is done by conducting a virtual experiment using two FE models. One represented a virtual SBS test, and the other was used for an optimization. The FEMU algorithm worked out well as expected and several factors that affect the optimization were examined. Several conclusions based on the study which are,

1. With the FE generated data, the mesh density has little effect on the convergence of the FEMU solution. The solution from the 624-element-model took about the same iteration as the 4,704-element-model to converge. The resulting material parameters were also identical from both models. However, the SBS model must contain enough element to complete the Abaqus job.
2. The numerical Jacobian is more robust than the analytical Jacobian. The numerical Jacobian provides more accurate and faster convergence of the solution with the resulting material parameter identical to the actual value. However, analytical Jacobian can be good at approximating the material parameter since it only takes few Abaqus jobs to get a good approximation of a solution. Therefore, in cases the actual material parameter is unknown, a good approach is to use the analytical Jacobian to run the FEMU for few iterations and get the approximate material parameter and use the numerical Jacobian later to get a more accurate solution.
3. A good approximation of initial parameters should lie between 40% lower to 40% higher than the actual material parameters. Although, the solution still converges at 80% higher than the actual parameter, it takes more calculation time.
4. The value of tau, τ , should be below 1×10^{-6} but not 0, when using analytical Jacobian. For numerical Jacobian, tau has little effect to the convergence of the solution.

4.2 Working with DIC data

To characterize nonlinear shear properties from NCF composite specimens, the FE generated strain data was substituted by a DIC full-field strain measurement data measured from the DIC system. The DIC data was interpolated on the nodes of FE grid as discussed in chapter 3. The initial approximation of material parameters were, $G_{13} = 370,000$, $K_{13} = 18,000$ and $n_{13} = 0.15$ and the FEMU was done on 5 different specimens. As mentioned in section 3.2, the SBS FE model must contain enough node points for a good DIC data interpolation. Therefore, the SBS FE model of 15,840 elements was used as shown in figure 21.

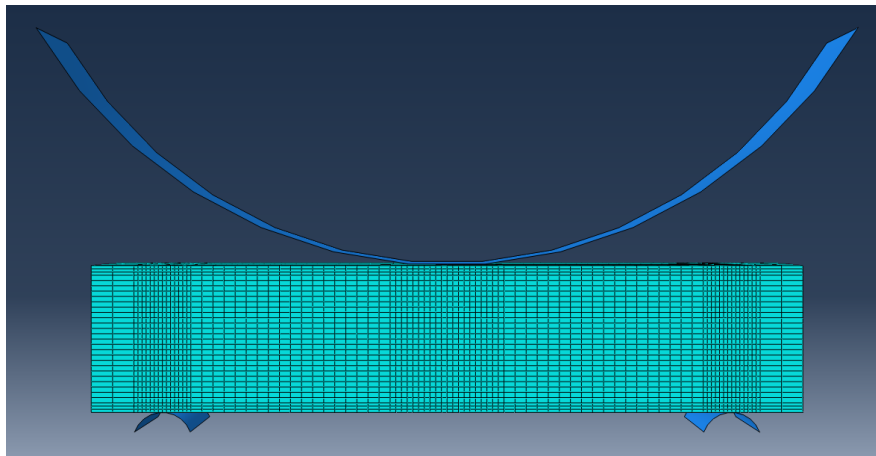


Figure 21. Short-beam shear finite element model with 15,840 elements

Figure 22 shows an approximated stress-strain curve from two different FE models based on the maximum interpolated shear strain and a closed-form solution ($0.75 * P/A$) for calculation of the maximum stress. The FE model with 15,840 elements contains more nodes which resulted in a more accurate data interpolation than the FE model with 4,704 elements. The approximated stress-strain curve of a 15,840-element model is identical to the approximated stress-strain curve from a raw DIC data.

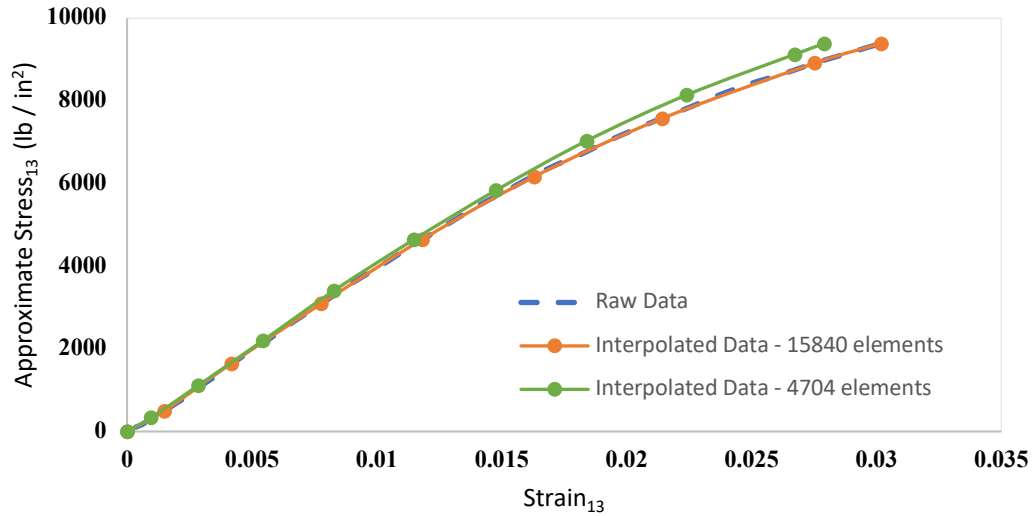


Figure 22. Approximated nonlinear stress-strain curves from two data interpolations.

Another point to consider is the area of interpolation. The DIC system cannot capture the strain on the entire face of the beam as illustrated in figure 3. There is still a strain field on the edge of the specimen that is not measured by the DIC system. Therefore, the missing data points must be taken into account when interpolating a DIC data on the FEM grid as shown in figure 23.

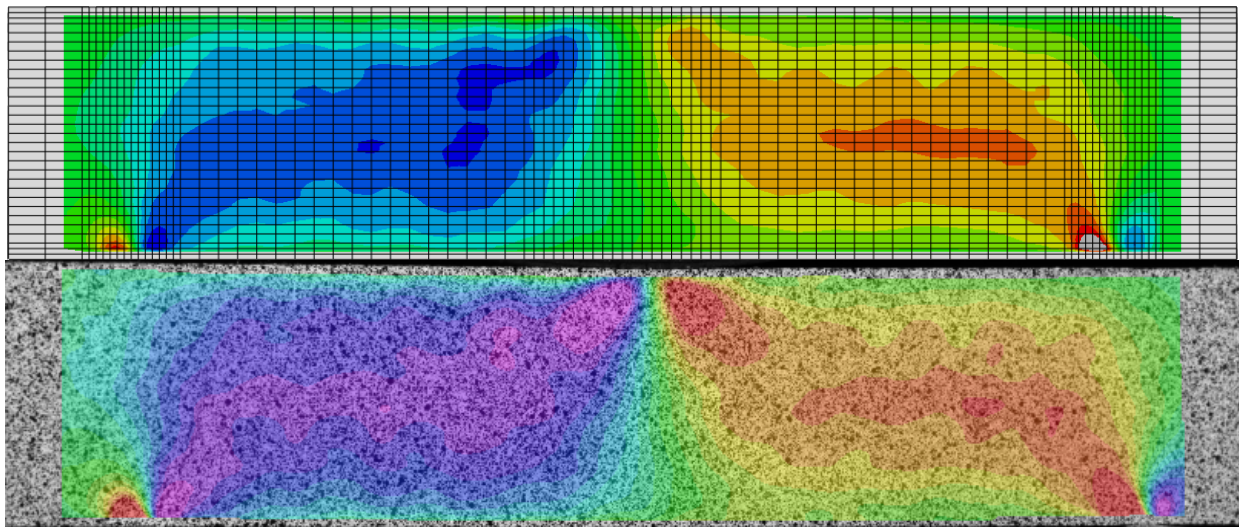


Figure 23. Top: DIC interpolated strain data on FEM grids. Bottom: DIC strain measurement on a specimen.

4.2.1 The Weight Matrix

According to equation (6) the value of each member in the weight matrix should be equal to $\frac{strain,i}{maximum\ strain}$. However, this expression provides an accurate result only when the data is smooth such as FE generated data. If the data contain lots of noise for low strain values, the weight matrix may be scaled up to compensate the noise as shown in equation (15).

$$w_{ii} = 100 \times \left(\frac{\varepsilon_{13,i}}{\varepsilon_{13,max}} \right)^2 \quad (15)$$

Figure 24 shows the difference between the data with less noise and data with lots of noise which occurred during the experimental testing.

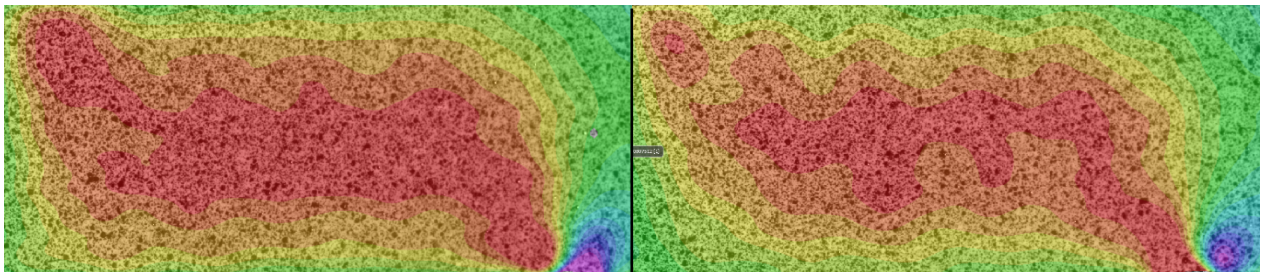


Figure 24. Left: DIC data from specimen 5 contains less noise. Right: DIC data from specimen 3 contains lots of noise.

Table 6 shows the result from one of the specimens with the optimization using the weight matrix from equation (6) and the weight matrix from equation (15). Note that using the scaled weight matrix resulted in less RMS strain error which means more accurate result. Therefore, the expression of the weight matrix in equation (15) was used for the optimization.

Iteration / Parameters	G_{13} (psi)	K_{13} (psi)	n_{13} (psi)	RMS strain error (%)
Initial Parameters	370,000	18,000	0.15	9.96%
Converged Eq. 6 W matrix	433,521	19,524	0.1570	5.75%
Converged Eq.15 W matrix	431,567	18,878	0.1506	5.50%

Table 6. A converged solution from different weight matrix.

4.2.2 Convergence of the Solution

As the number of iteration increase, the shear parameters will start to converge to the real value. The Levenberg-Marquardt optimization stops when the relative change in parameter is less 0.1%. Table 7 and 8 show the convergence of the solution from the specimen 5 using two different Jacobian, analytical and numerical.

Iteration / Parameters	G_{13} (psi)	K_{13} (psi)	n_{13} (psi)	RMS strain error (%)
0	370,000	18,000	0.15	10.52%
1	411,652	18,363	0.153	2.58%
2	421,418	18,484	0.154	2.49%
3	422,600	18,527	0.155	2.45%
4	422,723	18,545	0.155	2.39%

5	422,740	18,553	0.155	2.39%
---	---------	--------	-------	-------

Table 7. Convergence of shear parameters using DIC data and the analytical Jacobian.

Iteration / Parameters	G_{13} (psi)	K_{13} (psi)	n_{13} (psi)	RMS strain error (%)
0	370,000	18,000	0.15	10.52%
1	416,916	18,557	0.155	2.61%
2	423,040	18,617	0.155	2.25%
3	423,110	18,629	0.156	2.24%
4	423,106	18,628	0.156	2.23%

Table 8. Convergence of shear parameters using DIC data and the numerical Jacobian.

Using numerical Jacobian resulted in less RMS strain error as expected. However, the analytical Jacobian also provided a good result in 5 iterations. Note that, the numerical Jacobian took a total of 16 jobs for the solution to converge, while the analytical Jacobian only took 5 jobs. Therefore, with the real data, analytical Jacobian tend to be a better choice since it used less jobs for the solution to converge with acceptable amount of RMS strain error. If time is not the factor, numerical Jacobian will still provide a slightly more accurate solution. Figure 25 shows stress-strain curves of specimen 5 using different Jacobian for the FEMU. The result is almost identical.

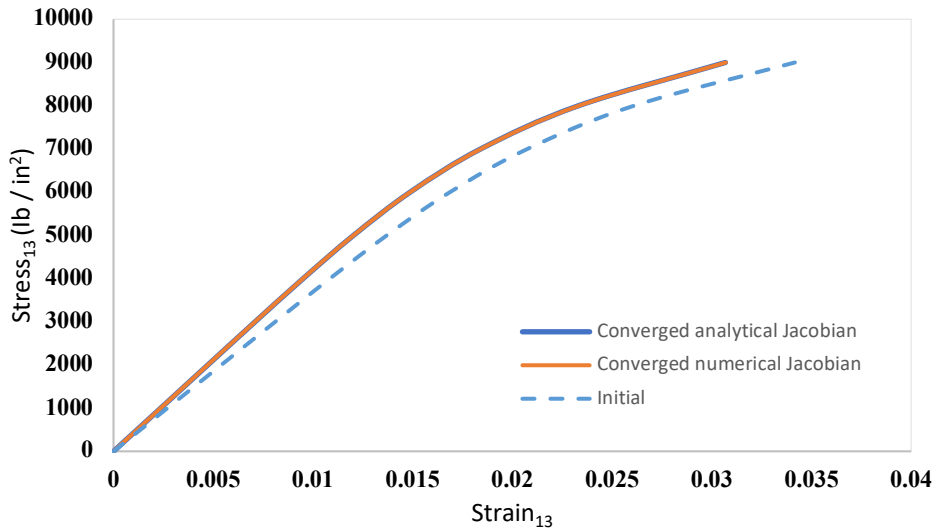


Figure 25. Nonlinear stress-strain curves of the solution from analytical and numerical Jacobian.

Three SBS FE models with three different mesh were tested with the DIC data to verify the convergence of the solution. The three model were 15,840-element-model, 22,800-element-model, and the 26,600-element-model. Figure 26 shows the convergence of the solution from three different model. As expected, the mesh density has little effect on the convergence of the solution. All three models provided identical stress-strain curve. However, it is worth noting that the mesh density might have some effect with the solution if less element were used, but that will result in an inaccurate DIC data interpolation.

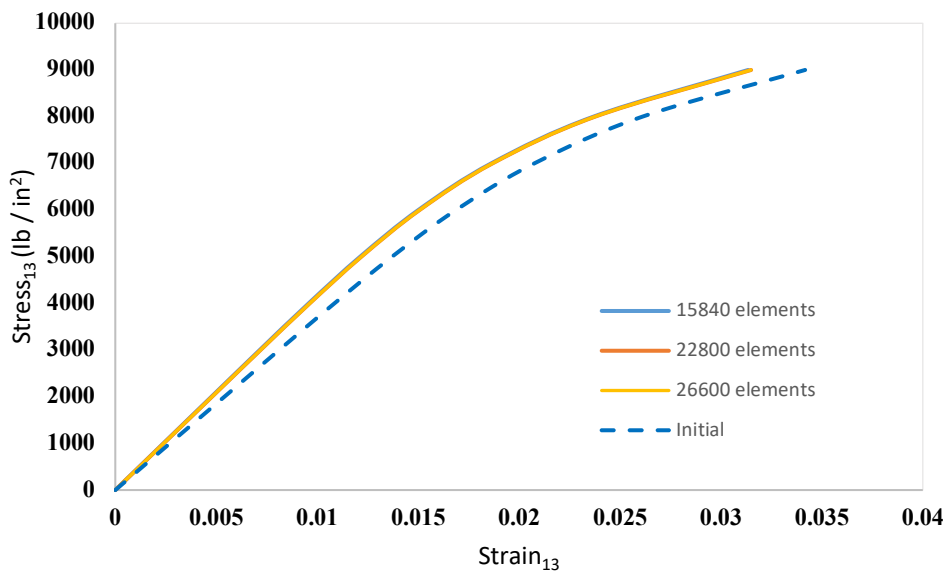


Figure 26. Convergence of the solution from three different models.

Several initial approximations of material parameter were used to verify the convergence of the solution as well. Figure 27 shows the convergence of the solution from two different initial approximations. The first approximation was $G_{13} = 370,000$, $K_{13} = 18,000$ and $n_{13} = 0.15$, and the second approximation was $G_{13} = 450,000$, $K_{13} = 25,000$ and $n_{13} = 0.20$. The solution from both approximations took 4 iterations to converge with the RMS strain error less than 5% from both approximations.

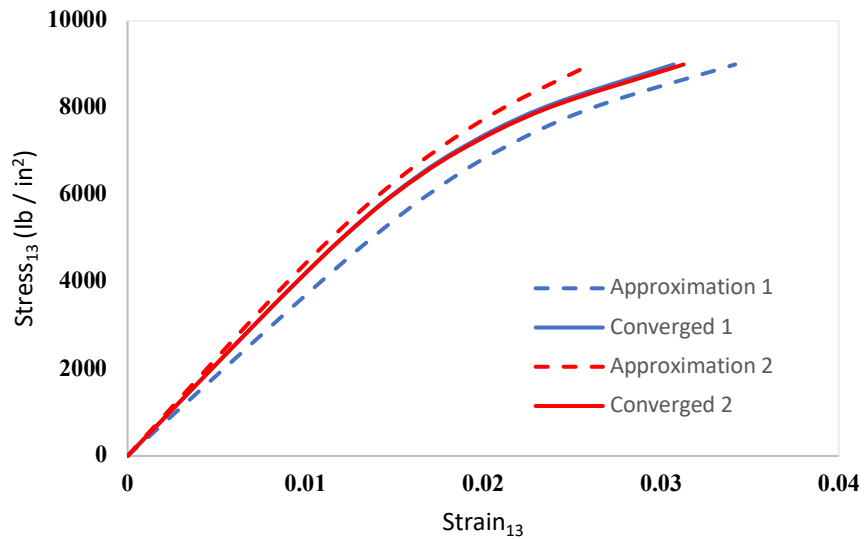


Figure 27. Convergence of the solution from two different initial approximations.

Approximation / Parameters	G_{13} (psi)	K_{13} (psi)	n_{13} (psi)	RMS strain error (%)
Approximation 1	370,000	18,000	0.15	10.52%
Converged 1	423,106	18,628	0.156	2.23%
Approximation 2	450,000	25,000	0.20	7.66%
Converged 2	423,117	18,631	0.158	2.6%

Table 9. Convergence of shear parameters from two different initial approximations.

4.2.3 Result from five Specimens

For the most accurate result, the numerical Jacobian, and the FE model with 15,840 elements were used to conduct FEMU and characterize material shear properties of five unstitched NCF composite specimens. The FEMU process took about 40 minutes for one specimen. After obtaining the material shear parameters from the optimization procedure, a Ramberg-Osgood stress-strain curves from all 5 specimens are plotted in figure 28 and the results are shown in table 10.

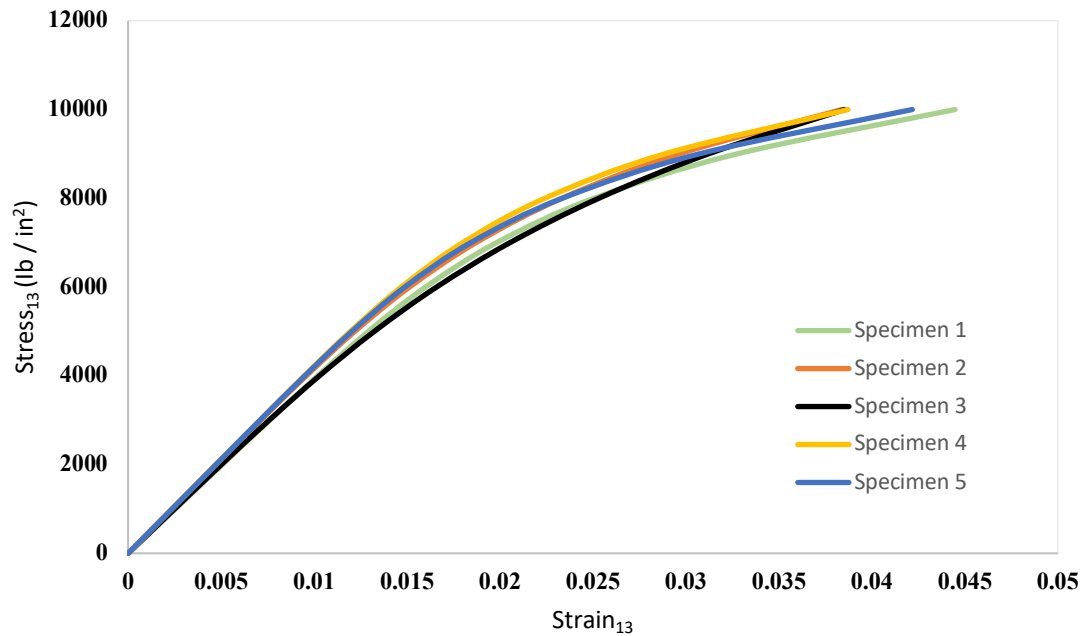


Figure 28. Nonlinear stress-strain curves from all five specimens.

Specimen/ Parameter	G_{13} (psi)	K_{13} (psi)	n_{13} (psi)	RMS strain error (%)
1	394,806	18,897	0.161	4.91%
2	422,014	21,734	0.184	4.48%
3	400,238	27,959	0.239	8.56%
4	424,172	19,065	0.154	5.58%
5	423,106	18,628	0.156	2.23%
Mean*	416,024	19,581	0.1609	
Standard Deviation*	14,172.88	1,446.37	0.0138	
Coefficient of Variation*	3.40%	7.38%	8.44%	

*The value does not include specimen 3 due to its high RMS error.

Table 10. Convergence of shear parameters from all five specimens with the average value and coefficient of variation of shear parameters.

Note that the RMS strain error does not go to 0 even if the solution is already converged, unlike FE generated data where RMS will always approach 0. There are several factors that prevent RMS strain error to become 0. First, the experimental testing was not done in the ideal condition where the SBS specimen slides on the frictionless contacts. There were some frictions between contacts and the specimens which could contributed to the noise in data. Second, the magnitude of the negative and positive shear strain on the SBS surface were asymmetric. Figure 29 shows the plot of a magnitude of maximum negative and positive strains. The maximum positive strain is a little higher than the maximum negative strain. Lastly, the specimen cross sectional area might not be a perfect square due to the manufacturing process.

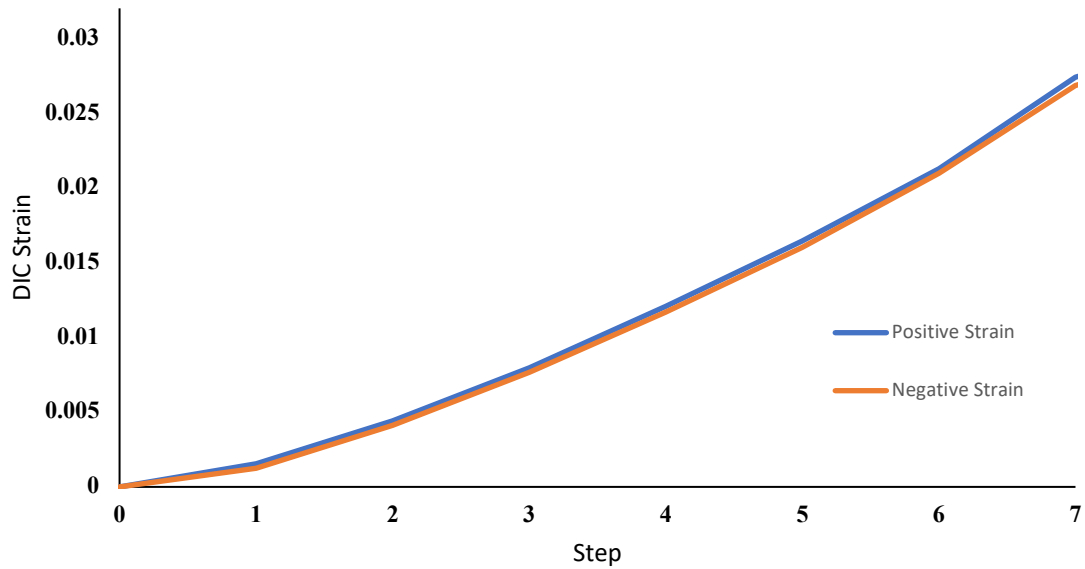


Figure 29. Magnitude of maximum positive and negative strain vs step.

According to the table, specimen 3 has a high RMS strain error. This is because the DIC data contains lots of noise from experimental testing as illustrated in figure 30. The positive strain region mid-way between the loading nose and the support on specimen has an uneven strain distribution which caused an unusual high RMS strain error. On the other hand, specimen 5 has more even strain distribution, resulting in less RMS error. Therefore, the result from specimen 3 was not used in averaging the material shear parameters.

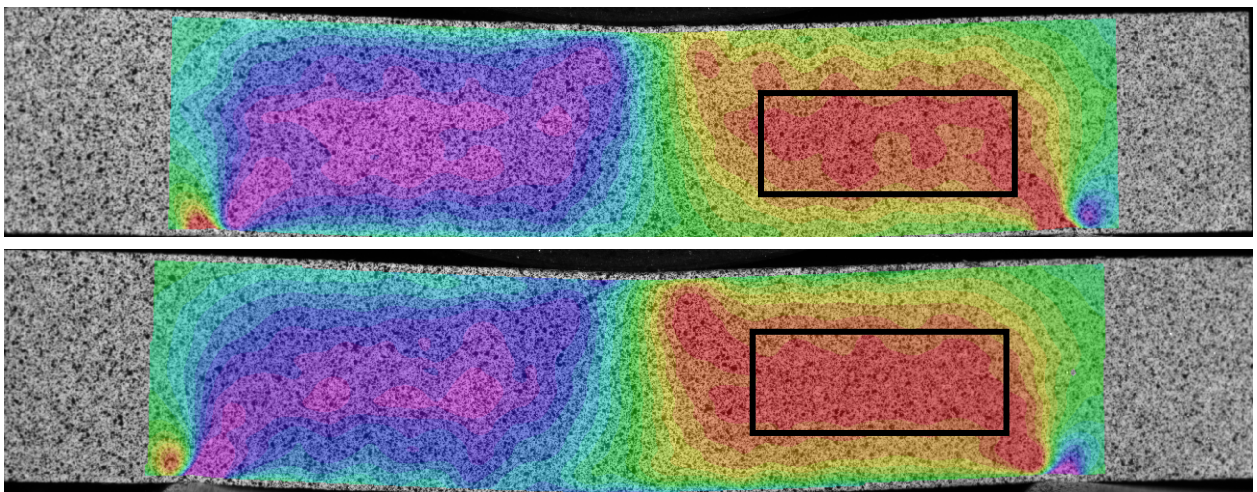


Figure 30. Top: Uneven strain distribution on specimen 3. Bottom: Even strain distribution on specimen 5.

The FEMU tend to provide an accurate result on a large value parameter. For example, the average value of G_{13} is 416,024 has the coefficient of variation of 3.4%, while the value of n_{13} which is 0.163 has the coefficient of variation of 8.44%. Therefore, the FEMU will be more accurate when optimizing large value of parameter.

4.2.4 Conclusion on FEMU with DIC Data

Five unstitched NCF composite specimens were used in this work for the characterization of the material shear properties. All five specimens used SBS experiment for a DIC strain measurement and the FEMU process to characterize shear properties. The average shear parameters from the 5 specimens are $G_{13} = 416,024$, $K_{13} = 19,581$ and $n_{13} = 0.1609$ psi. Several points can be concluded from the study, which are,

1. Getting a good DIC data interpolation on the FE grid is very important. The FE grid must contain enough node points for an accurate data interpolation that really represents the DIC test data.
2. The strain measurement from the DIC system may contain some noise which occurred during experimental testing. Therefore, the weight matrix of equation (15) should be implemented instead of equation (6). The optimization using the weight matrix from equation (15) gives a slightly better result in terms of RMS strain error.
3. The SBS FE model used for the FEMU with DIC data contains 15,840 elements. If more material parameters were to be optimized, the numerical Jacobian may not be the best option. For example, if all 3 direction of shear properties need to be determined, 9 constitutive shear properties will need to be optimized. Which means that 10 Abaqus jobs are needed to be compiled in order to update the material parameters for 1 iteration using numerical Jacobian. On the other hand, the analytical Jacobian can update all 9 parameters in 1 iteration using only 1 Abaqus job with slightly less accuracy. However,

a few more iterations of the optimization will make up for the accuracy. Therefore, when working with real data, analytical Jacobian may be the best option if the model contains a lot of elements, and more parameters were to be optimized.

- The RMS error from all specimens did not go to 0 even the solution is already converged. Unlike, FE generated data where the RMS error will always go to 0. Figure 31 compares the RMS strain error from the FE generated data and the DIC data. This is because the data from the experimental testing contain some noises as well as other factors such as, imperfect specimen and asymmetric negative and positive shear strain in the specimen.

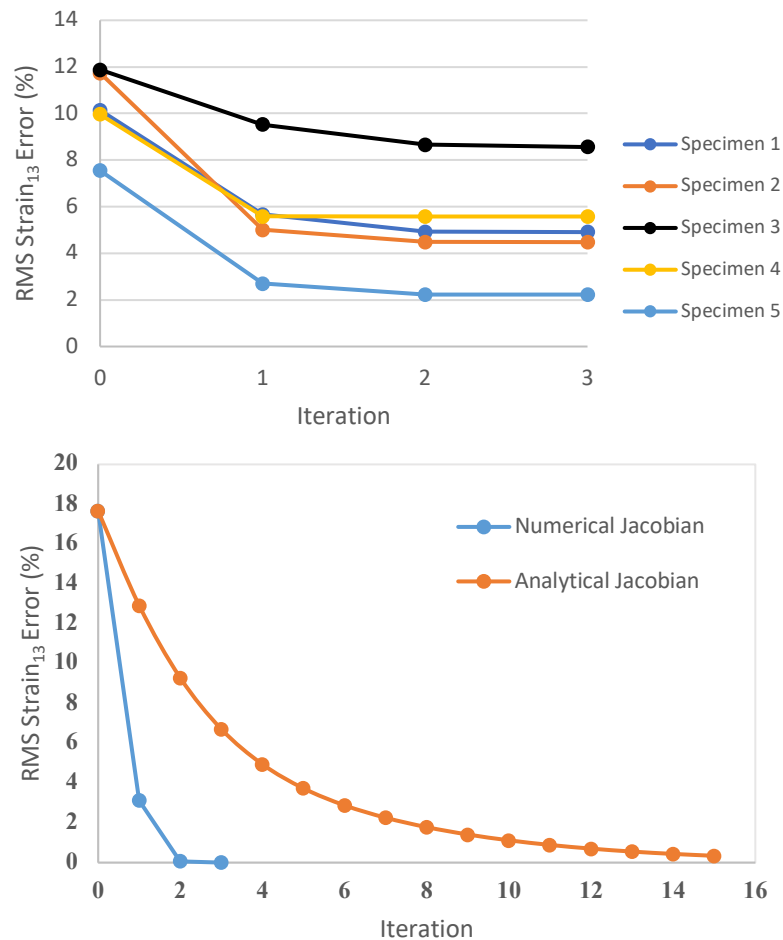


Figure 31. Top: RMS strain error of five specimens. Bottom: RMS strain error from virtual testing.

5. The FEMU provides more accurate result when optimizing large value of material parameter such as shear modulus and young modulus. As shown in table 10, the value of G_{13} has less coefficient of variation than n_{13} which has a very small value. However, normalizing the value of material parameter in FEMU process can help reducing the coefficient of variation and provides more accuracy in small value parameters.

Chapter 5

Streamline the Process

After the SBS experimental testing, a series of output files have to be processed in a VIC3D 7 software. These processes include setting DIC coordinate system, calculating strain and exporting the output into CSV files for further data analysis in Python. Figure 32 shows the VIC3D 9 software interface.

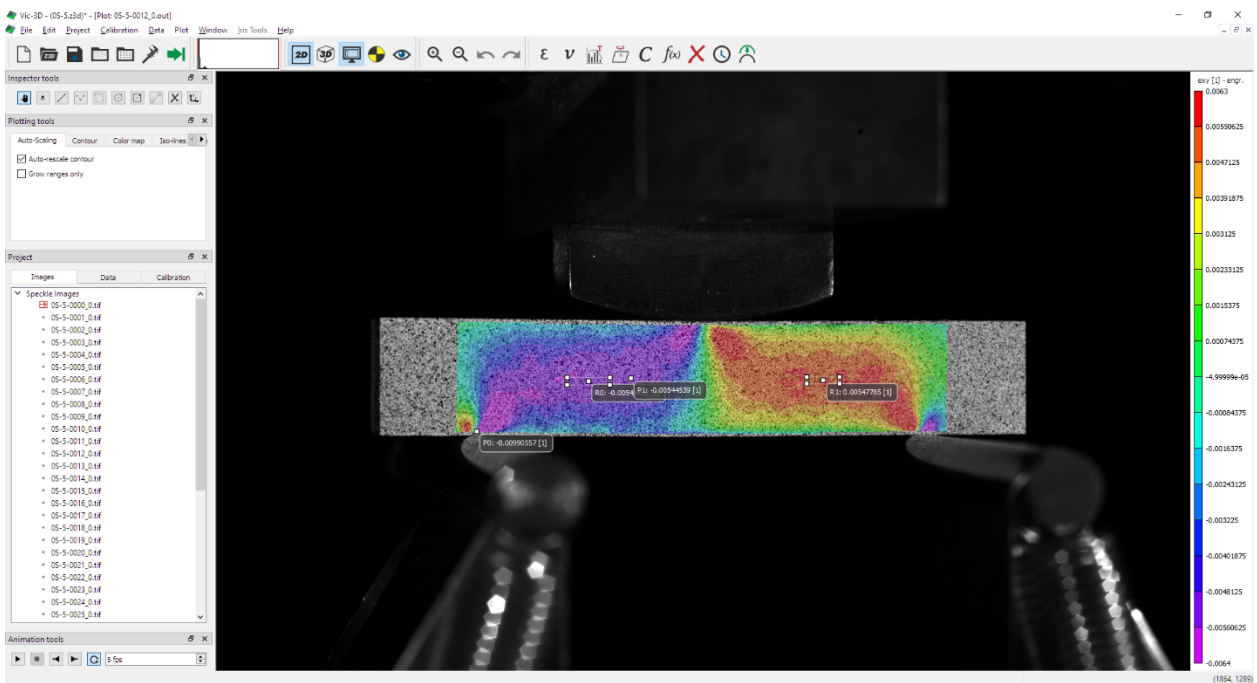
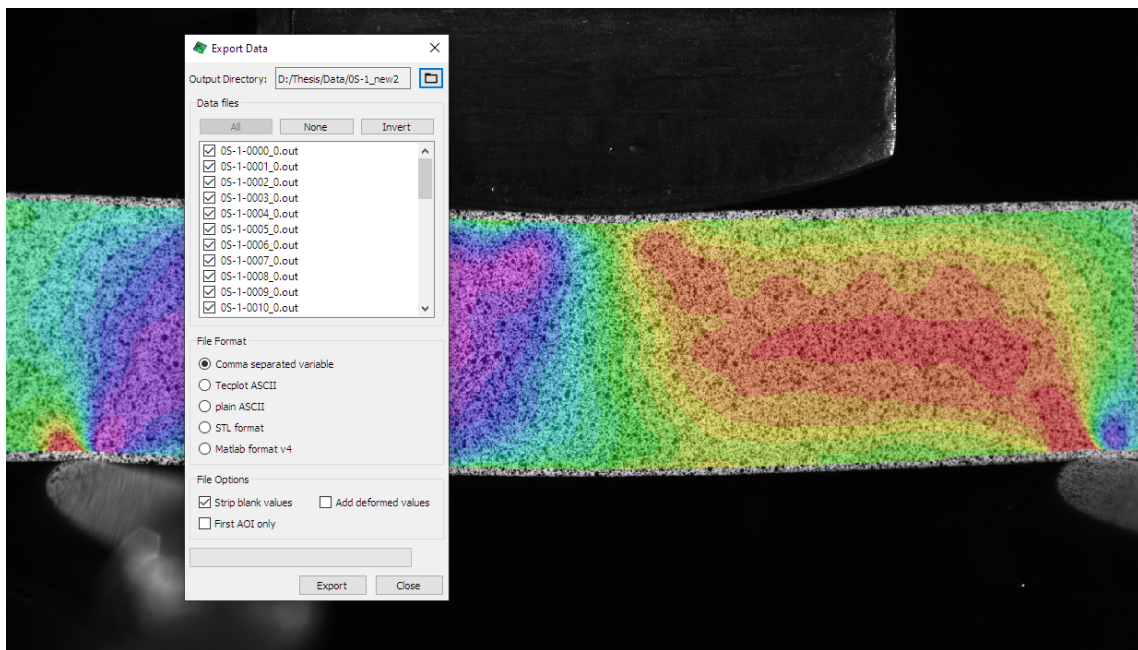


Figure 32. VIC3D 9 Interface.

These processes consume a considerable amount of time when working with several test specimens. However, with the introduction of VicPy Python module that comes with the latest VIC3D 9 software, all processes mentioned above can be done using Python which will save up a great deal of time. The VicPy Python module allowed loading, manipulating, and saving the VIC3D output file in Python, resulting in faster and more efficient data processing.

The module can access DIC .out file which contains several parameters including DIC coordinates and strains. This means that the process of exporting DIC data into CSV files can be eliminated and the whole DIC data processing can be automate using Python scripts. Figure33 shows the process of exporting .CSV file in VIC3D 9 and reading the value from .CSV file which consumes a lot of time. The old process can be replaced by a few lines of code as illustrated in figure 34.



```

file = Data_Directory.split("\\")
file = file[-1] + "-00"

for Id in range(len(data_ID)):
    with open(Data_Directory + "\\ " + file + data_ID[Id] + "_0.csv") as gg:
        reader = csv.reader(gg)
        gg.readline() # read header
        for line in reader:
            for i, j in enumerate(line):
                if i == 0:
                    x_DIC.append((float(j) / 25.4) + Shift[0])
                if i == 1:
                    y_DIC.append((float(j) / 25.4) + Shift[1])
                elif i == 8:
                    value_DIC.append(float(j))

```

Figure 33. Old Process: Exporting CSV file, reading CSV file and aligning coordinate system.

```

for Id in range(len(data_ID)):

    data = VicDataSet()
    data.load("0S-1-00" + data_ID[Id] + "_0.out")

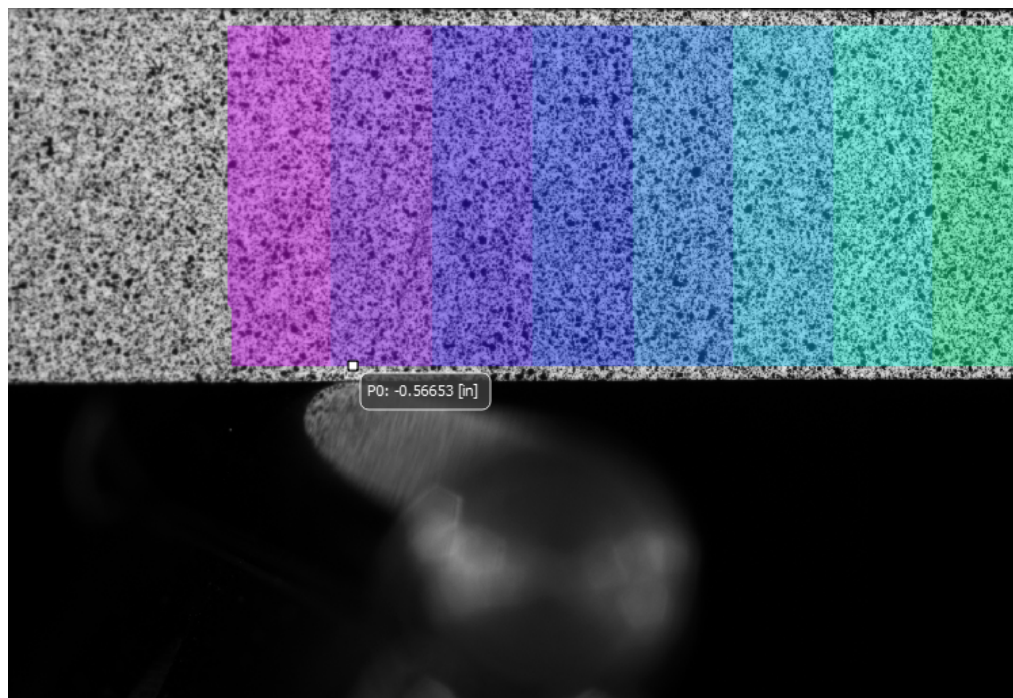
    t = RigidTransformation()
    t.setTranslation([Shift[0], Shift[1], 0.0])
    data.transform(t)
    data.computeStrain(tensorType="engineering", computePrincipalStrain=False)
    data.save("0S-1-00" + data_ID[Id] + "_0.out")

    v = data.asArray(['X', 'Y', 'exy', 'sigma'])
    mask = v['sigma'] >= 0
    v = np.extract(mask, v)
    nodes = np.column_stack((v['X'] / 25.4, v['Y'] / 25.4))
    strain = v['exy']

```

Figure 34. New process: Reading output file and aligning coordinate system using Python.

The VicPy module also allows the coordinate adjustment of DIC data directly from .out file using RigidTransformation class. The rotation of DIC data can also be done using setRotation and setAngles function. These allow user to easily align the DIC coordinate with the FE coordinate system. Figure 35 shows the result from translating the DIC x-coordinate in the left contact area which is -0.56653 to match with FE x-coordinate which is -0.6.



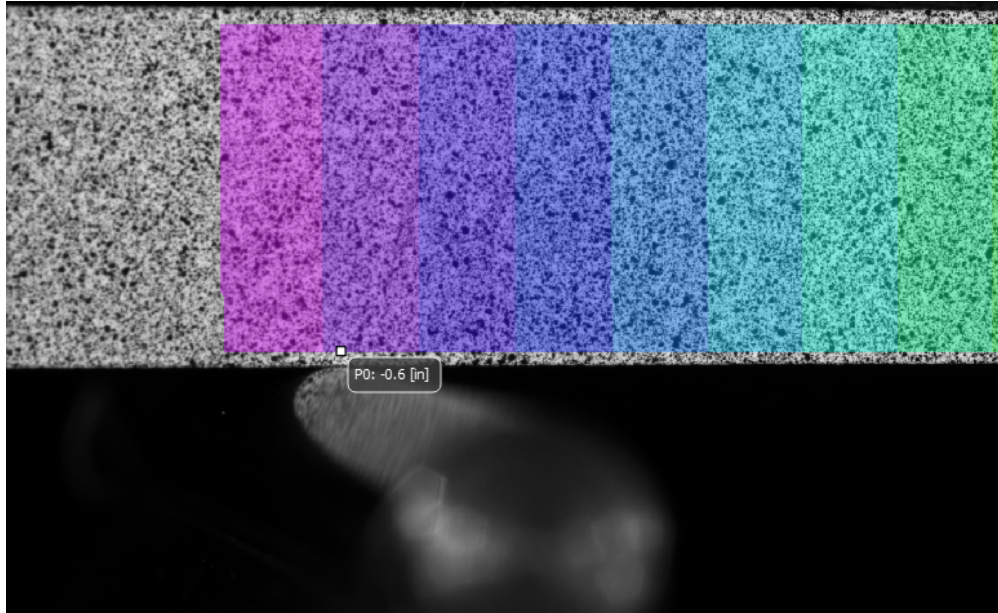


Figure 35. Top: DIC x-coordinate before the adjustment. Bottom: DIC x-coordinate after the adjustment.

The VicPy module is built for both Python 2 and 3. But in order to integrate the module with Abaqus, the built for Python 2.7 must be implemented. Some of the Python scripts shown below are reading DIC coordinate, shifting coordinate, and calculating strains.

1. Reading the DIC .out File

The script reads DIC X-Y coordinate and strain 1-3 directly from DIC .out file.

```

from VicPy import*
import numpy as np

data = VicDataSet()
data.load("file_name.out")
v = data.asArray(['X', 'Y', 'exy'])
coordinate = np.column_stack(v['X'], v['Y'])
strain = v['exy']

```


2. Aligning Coordinate System

The script can be implemented to the current workflow for aligning DIC coordinate system to FE coordinate system. For the translation in X, Y and Z, the input is in inch. For the rotation, the input is a half angle in degree.

```
from VicPy import*
import numpy as np

data = VicDataSet()
data.load("file_name.out")

# Translation
#      X      Y – in inch
Shift = [-1.23 * 25.4, 0.1526 * 25.4]
# Align DIC coordinate system with FE coordinate system
t = RigidTransformation()
t.setTranslation( Shift[0], Shift[1], 0.0)
data.transform(t)

# Re calculate Strain
data.computeStrain(tensorType="engineering", computePrincipalStrain=False)
data.save("file_name.out")
```

```
# Rotation
#   alpha(x), beta(y), gamma(z) - double angle in degree
Angle = [0.0, 15.0, 0.0]
t = RigidTransformation()
r = Rotation()
r.setAngles(Angles[0], Angles[1], Angles[2], radians=False)
t.setRotation(r)
data.transform(t)

# Re calculate Strain
data.computeStrain(tensorType="engineering", computePrincipalStrain=False)
data.save("file_name.out")
```

For DIC data rotation, it is important to re calculate the shear strain in every transformation.

From here, the DIC output strain data is ready to be interpolated and transferred onto FE grids in Abaqus without exporting several CSV files. The script will speed up the process when working with several test specimens. More information regarding class definition of VicPy module can be found at <http://www.correlatedsolutions.com/supportcontent/VicPy/>. [11]

Chapter 6

Conclusion

The Short-Beam Shear experiment developed at the AMSL provided an accurate measurement of a shear strain in the 1-3 direction. The method takes advantage of a full-field non-contact strain measurement to capture shear strain that can be transferred into the FE model for an FEMU process. Several factors that affect the accuracy of material inputs generated were studied in this work which included mesh density, accuracy of Jacobian matrix, initial approximation of material parameters, tau value and the weight matrix. The FEMU process was tested with a virtual experiment to verify the accuracy of the algorithm. After that, the FEMU was used with DIC data from a SBS experiment to characterize material shear properties.

A virtual experiment was conducted to simulate the SBS experimental testing and FE generated data were used for the FEMU process. The study from a virtual experiment found that, mesh density has little effect on the convergence of the solution since a very coarse mesh model provided the same result as a very fine mesh model. This proves that the FEMU algorithm is very robust and accurate. The study also found that the numerical Jacobian matrix provides more accurate result than the analytical Jacobian. With FE generated data, the Numerical Jacobian always used less iteration and therefore less time for the solution to converge. Lastly, the best initial approximation of material parameters should lie between 40% lower to 40% higher than the actual material parameters for an optimal convergence.

The FEMU algorithm is then used with a full-field strain measurement data from a DIC system to characterize nonlinear shear parameters of five unstitched NCF composite specimens. The captured shear parameters were $G_{13} = 416,024$, $K_{13} = 19,581$ and $n_{13} = 0.1609$ psi with the coefficient of variation of 3.40%, 7.38% and 8.44% respectively. Contrary to the FE generated

data, the RMS strain error from FEMU of DIC data never reached 0. This is because there are several factors that affect the convergence of the solution including, noise during experimental testing and imperfect specimens.

With the DIC data, the study found that even though numerical Jacobian provided more accurate result, it is too time consuming and computational expensive. The result from analytical Jacobian was a bit less accurate according to the RMS strain error. However, it only required one third of a time the numerical Jacobian used. With more material parameters to optimize, analytical Jacobian seems to be the best option for FEMU. Also, the amount of node points on the FE grids has to be high enough to get a good DIC data interpolation. With low mesh density on the SBS, the amount of node will not be enough for an accurate data interpolation which will result in an accuracy of the resulting parameters.

Lastly, with the new version of VIC3D DIC software which support the VicPy python module, the entire working process can be streamlined using Python. The process of exporting the CSV file can be totally eliminated and the process of aligning the DIC coordinate with FE coordinate can be automated using Python, which save a great deal of time.

In conclusion, the SBS experiment and the FEMU process developed at the AMSL provide convenience and accuracy in characterizing nonlinear material properties of composite materials which are vital in a virtual design platform for predicting failure and deformation. The method could potentially become a standard testing method in measuring nonlinear shear properties of fiber-reinforced composite materials.

References

- [1] Maria Girardi, Cristina Padovani, Daniele Pellegrini, Margherita Porcelli and Leonardo Robol “Finite Element Model Updatung For Structural Applications”
- [2] American Society for Testing and Materials, "ASTM D2344".
- [3] Bastiaan Corstiaan Wouter Van Der. Vossen, "Nonupdating Methods for the Characterization of Nonlinear Stress-Strain and Delamination Fracture Properties of Fiber Reinforced Composites," 2018.
- [4] Andrew Makeev, Guillaume Seon, Julia Cline, Brian Shonkwiler "In quest of methods for measuring 3D mechanical properties of composites," *Composites Science and Technology*, p. 3, 2014.
- [5] Robert M. Jones, *Mechanics of Composite Materials*, Taylor & Francis Group, 1998.
- [6] K. Vallons, "Fatigue of non-crimp fabric," Leuven, Belgium, p. 276.
- [7] Guillaume Seon, Andrew Makeev, Brian Shonkwiler, Yuri Nikishkov, "An Integrated Approach for Characterizing Non-Crimp Fabric Composites with Interlaminar Stitching Reinforcement," p. 2, 2021.
- [8] Wikipedia, "Jacobian matrix and determinant," Wikipedia, [Online]. Available: https://en.wikipedia.org/wiki/Jacobian_matrix_and_determinant. [Accessed 20 3 2021].
- [9] Henri P. Gavin, "The Lavenberg-Marquardt algorithm for nonlinear least squares curve-fitting problems," 2020.

- [10] Hans Bruun Nielsen, "Damping Parameter in Marquardt's Method," Technical University of Denmark, Lyngby - Denmark, 1999.
- [11] Correlated Solution, [Online]. Available:
<http://www.correlatedsolutions.com/supportcontent/VicPy/>. [Accessed 27 June 2021].