PROGRESS AND CHALLENGES IN NON-LINEAR CONSTITUTIVE MODELING FOR COMPOSITE MATERIALS

by

Eric Jason Booth

A professional paper submitted in partial fulfillment
of the requirements for the degree

of

Master of Science

in

Mechanical Engineering

MONTANA STATE UNIVERSITY
Bozeman, Montana

April 2007

APPROVAL

of a professional paper submitted by

Eric Jason Booth

This professional paper has been read by each member of the thesis committee and has been found to be satisfactory regarding content, English usage, format, citations, bibliographic style, and consistency, and is ready for submission to the Division of Graduate Education.

Dr. Douglas S. Cairns

Approved for the Department of Mechanical Engineering

Dr. Christopher H.M. Jenkins

Approved for the Division of Graduate Education

Dr. Carl A. Fox

STATEMENT OF PERMISSION TO USE

In presenting this professional paper in partial fulfillment of the requirements for a master's degree at Montana State University, I agree that the Library shall make it available to borrowers under the rules of the Library.

If I have indicated my intention to copyright this thesis by including a copyright notice page, copying is allowable only for scholarly purposes, consistent with "fair use" as prescribed in the U.S. Copyright Law. Requests for permission for extended quotation from the reproduction of this thesis in whole or in parts may be granted on by the copyright holder.

Eric Jason Booth

April, 2007

TABLE OF CONTENTS

LIST OF FIGURES

ABSTRACT

In this paper is outlined a technique for determining the non-linear constitutive properties of an orthrotropic laminated material such as a fiber reinforced composite. The Characterization of Composites is difficult because of the anisotropic stiffness properties of the material and also because of the complex manner in which they accumulate internal damage and eventually fail.

Presented here are techniques for determining the mechanical properties of a material by solving what is known as the Inverse Problem. In such a problem, the response of a material system to an external stimulus is measured, then, this information is used to divine the corresponding material properties of that system. Progress in solving the Inverse Problem for Non-Linear Material Properties is presented as well as challenges that remain.

INTRODUCTION

In recent years fiber reinforced composite materials have found increasing use in primary structural components for a wide variety of applications. From high cost aerospace vehicles to low cost wind turbine blades, composites are increasingly finding a niche because of their unique combination of high strength, excellent stiffness, and light weight [5]. The analysis techniques used to take advantage of the attributes have also improved in recent years. However, the complex manner in which composite materials fail and accrue damage is not yet fully understood. Additionally, no comprehensive technique exists for designing structures in the context of accruing damage. Current methodologies simply can not engage the complexities of damage evolution in composite materials and therefore the engineer cannot design parts that take full advantage of the properties that composites possess. In this paper is summarizes the research efforts of the author and others in attempting to address this issue.

BACKGROUND

Composite materials typically consist of bundles of relatively stiff fibers, usually carbon or glass, embedded in a polymer matrix, typically epoxy or polyester [5]. When subjected to mechanical strains these materials initials exhibit linear elastic behavior. As strains increase, small discontinuities appear in the material and eventually coalesce. We observe this micro scale damage accumulation as material softening when looking at the macro scale.

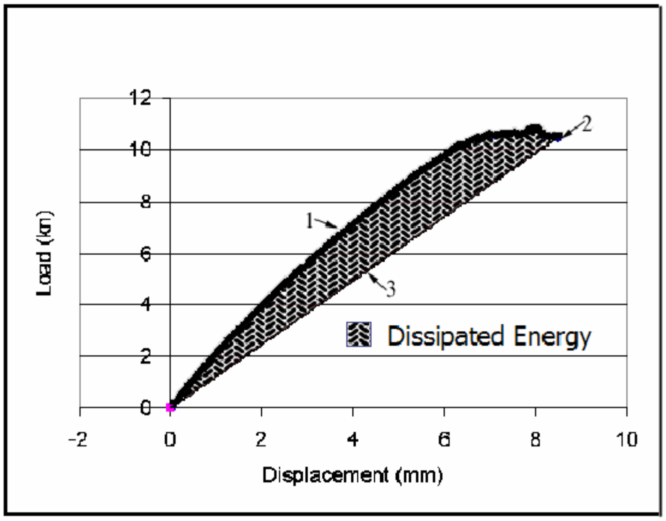

Figure 1: Typical Load-Displacement Curve for Fiber Reinforced Composite Material

In Figure 1 a typical load-displacement curve for a composite is shown. This curve is approximately linear at low displacements but rounds off (softens) at higher displacements. If the composite were unloaded at any point during a test, the data would travel back to the origin in approximately a straight line. The area enclosed by this load-unload path corresponds to the energy dissipated by the damage process. Subsequent Reloading will follow up the linear unloading line to point 2 on the figure, where it will rejoin the previous loading path and energy dissipation will continue.

Dissipated Energy Density was shown to be metric for characterizing material damage first by Mast at the Naval Research Lab [1] and later by Ritter at Montana State University [2]. Both of these researchers used a numeric optimization scheme to find the coefficients of a function that predicts dissipated energy density as a function of in-plane strains. In other words, an inverse problem can be solved that allows the magnitude and location of energy dissipation (damage formation) to be predicted knowing only the in-plane strains within a composite. For a compact tension sample typical results are shown in figure 2.

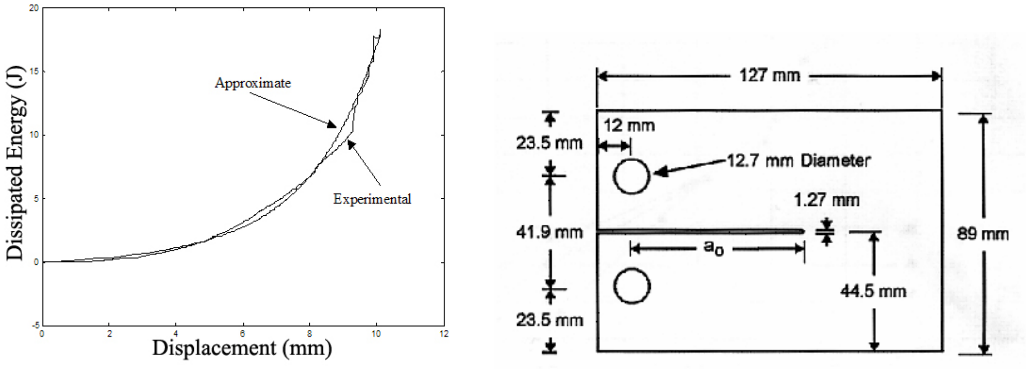

Figure 2: Dissipated Energy for Compact Tension Sample [2]

DED allows one to generate a map of where and to what degree energy dissipation has occurred within any geometry of part. It has been proposed that

DED can be used as a metric for designing composite structures. One question that is left unanswered is ‘what is the structural response of a composite that has been damaged?’. DED uses a linear elastic Finite Element Model both for solving the inverse problem as well as for any subsequent DED analysis. In summary, Dissipated Energy Density predicts where and to what degree a composite has been damaged but does not predict the way a composite structure with damage behaves when subject to structural loads.

The determination of In-Situ Linear Elastic material properties was explored by Huang also at Montana State University[3]. Huang used an Ansys finite element model with embedded optimization routine to find elastic properties for a composite based on experimental data. In this approach composite samples were loaded in MSU’s In-Plane-Loader as well as with a conventional Instron tensile testing machine. Data from these tests were fed into the Ansys code and elastic properties were adjusted virtually by numerical optimization until a best fit was found. Initially, convergence issues were encountered but these were solved by carefully bounding the material stiffness in the fiber direction.

It was shown that elastic ply properties could be divined from experimental data by solving what is known as the Inverse Problem. In such a problem the governing equations (or coefficients of the governing equations) of a system are not know, in this case the elastic moduli. The response of the system to some external loading can be measured experimentally. The inverse problem, then, is to determine the coefficients of the governing equations that best match the experimental data. In other words, given the known loads and displacements of some physical test, what are the material properties?

Armed with a knowledge of dissipated energy methodologies and also experience with in-situ linear elastic property determination it seem likely that these successes could be extended into the non-linear regime.

PROBLEM FORMULATION

We can think of the constitutive properties of a composite lamina as being elastic and exhibiting strain softening behavior. The In-Plane Elastic Moduli are assumed to be functions of three in-plane strains. Formulation in terms of stresses is also possible, in principle, but stresses are not necessarily monotonically increasing with increasing displacement. So this approach is more difficult.

$$\boxed{\begin{gathered} E_1 = E_1(\varepsilon_1,\varepsilon_2,\gamma_{12}) \\ E_2 = E_2(\varepsilon_1,\varepsilon_2,\gamma_{12}) \\ G_{12} = G_{12}(\varepsilon_1,\varepsilon_2,\gamma_{12}) \end{gathered}}$$

Figure 3: Elastic Properties as Functions of In-Plane Strains

At some point we need to choose a mathematical form for the stiffness functions shown above. Experience with dissipated strain energy density prompted the choice of a three dimensional interpolation element as illustrated in Figure 4. Ritter used this approach successfully in his work with dissipated strain energy density [2].

At any strain point within the element the stiffness value can be found by linearly interpolation from the known values at the nodes, in this diagram numbered 1-8. In this manner the non-linear constitutive response can be found by solving for the stiffness values at the nodes.

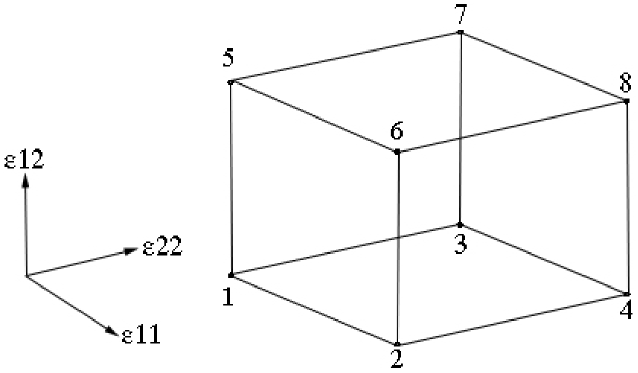

Figure 4: Three Dimensional Interpolation Element

Of course, a simple linear interpolation model is of limited value because only a linear relationship can be represented. Multiple elements can be assembled into an array as shown in figure 5. The nodal values are allowed to take whatever value is required. By doing this more complicated stiffness function can be constructed but the simple interpolation procedure within any particular node is retained.

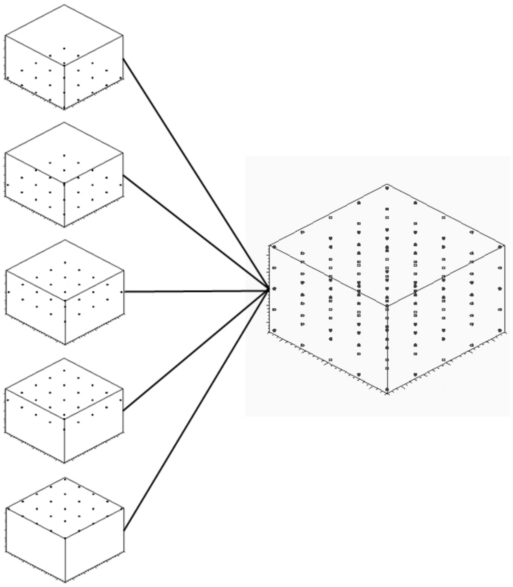

Figure 5: Array of 125 Interpolation Elements

Now, the behavior of the composite lamina will be represented by three of these 125 node interpolation arrays, one for each of the stiffness parameters (E_1,G_2,G_{12}) in ply principal coordinates. The entire non-linear constitutive relationship is contained within. The task that remains is to create an algorithm to determine the nodal values of these interpolation elements.

Like Dissipated Strain Energy Density methods, the domain of our composite coupon is discretized in Abaqus. A typical mesh and coupon geometry is shown in Figure 6.

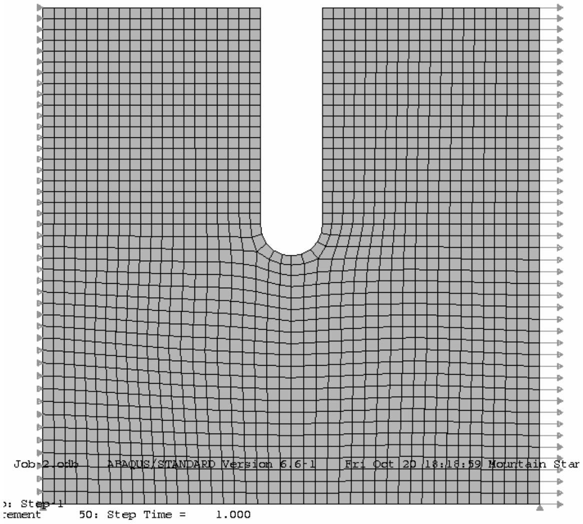

Figure 6: Finite Element Mesh for Composite Coupon

Much like Dissipated Strain Energy Density and In-Situ Elastic Properties, an equilibrium equation can be constructed which compares the nodal values of our inverse problem solution with a physically measured quantity, i.e. experimental data. This error norm must take the form shown in Equation 1.

$$\boxed{[\Sigma_{elements}\ \textit{Interpolation Functions}]\ {*}\{\textit{Nodal Values}\} = \{\textit{Experimental Values}\}}$$

Equation 1: Equilibrium Equation Form

For Non-Linear Elastic Property Determination the most logical quantity to be conserved is Elastic Strain Energy. This conservation principle can be applied at every data point throughout a laboratory test so it is shown in matrix form in equation 2.

$$\boxed{\{\Sigma_{elements}\ \textit{Element Elastic Strain Energy}\} = \{\textit{Experimental Total Strain Energy}\}}$$

Equation 2: Conservation of Energy

At this point the available procedures can be separated into two categories, those with Whole Field Strain Data and those without.

With Whole Field Strain Data

If whole Field Strain Data is available this problem is relatively straight forward. The Experimental Strain Energy values on the right side of Equation 1 are known. At each data point The Interpolation Function on the left side of the equation are evaluated for each element at the strains of that particular element. Summation can be carried out and the problem is ready for optimization.

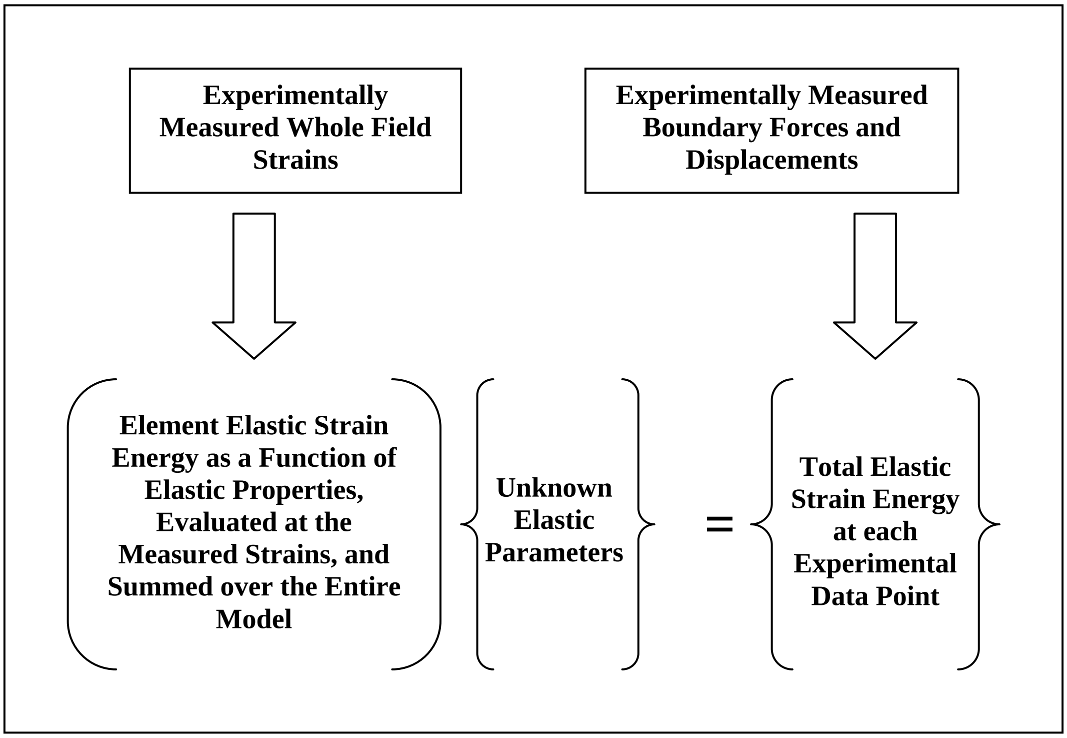

Figure 7: Optimization Problem with Whole field Stain Data

In the scenario with whole field strain data, solving of a Finite Element model is not necessary. The domain must still be discretized (divided into a suitable mesh) but the required strain values are found by experiment not by numeric optimization.

Without Whole Field Strain Data

If whole field strain data is not available the problem is far more difficult. Individual Element Strain values must be predicted at each data point by some method. A linear elastic model could predict strains much like Dissipated Energy Density methodology. But, using linear strains to solve for non-linear stiffness behavior makes little sense. The alternative is to use a non-linear material model with a set of assumed parameters and solve the inverse problem. Then the newly found material behavior could be used to rerun the analysis. In this circular fashion an acceptable set of non-linear parameters could eventually be found.

Deviation from a linear elastic material model introduces several difficulties. First, the principle of superposition no longer applies. With a linear material model a Finite Element model can be run for a unit displacement and then simply be scaled to any displacement that is desired. Also, a displacement involving two different loading directions can be created from adding the correct portion of a unit displacement in those two directions. A Non-linear material model requires that Finite Element data be generated for each data point within a laboratory experiment. Multiply this by the number of tests in any given data set and one can see that this approach becomes computationally intensive. It becomes obvious that automation of the FEA modeling becomes mandatory.

A second and more profound difficulty is that we now have in essence an inverse problem within an inverse problem. We are still attempting to back out the material behavior that is experienced under a certain loading case. However, we have no way of determining the strain state and can’t directly construct an error norm. So we assume a material behavior and in a somewhat backhanded way iterate our way to a converged solution.

IMPLEMENTATION

Early in the course of this research the available whole field strain techniques were surveyed. Several potential technologies were identified including:

-Moiré Inferometry
-Direct Photographic Strain Measurement
-Photoelastics

Of these, photoelasticity was the most promising. Photoelasticity uses the birefringent properties of certain polymers and their interaction with polarized light to create fringe patterns which correspond to the strain of the birefringent coating. A photoelastic coating is bonded to the sample being analyzed and polarized light is reflected through the coating [4]. The resulting image is recorded by conventional or digital photographic methods. A typical image is shown in Figure 8.

A finite element model can be used to predict the fringe patterns of photoelastic images. In figure 9 is shown a photoelastic image overlayed with a FEA prediction of the fringe pattern at the same boundary displacements. This image is interesting because it shows a region near the stress concentrating notch where damage to the material has occurred. This region is seen as the blurred dog bone in the center where fringe lines are too close together to be seen individually. This corresponds to an area of material softening and consequently significantly increased strains. The surrounding fringe patterns show the load redistribution due to this local damage.

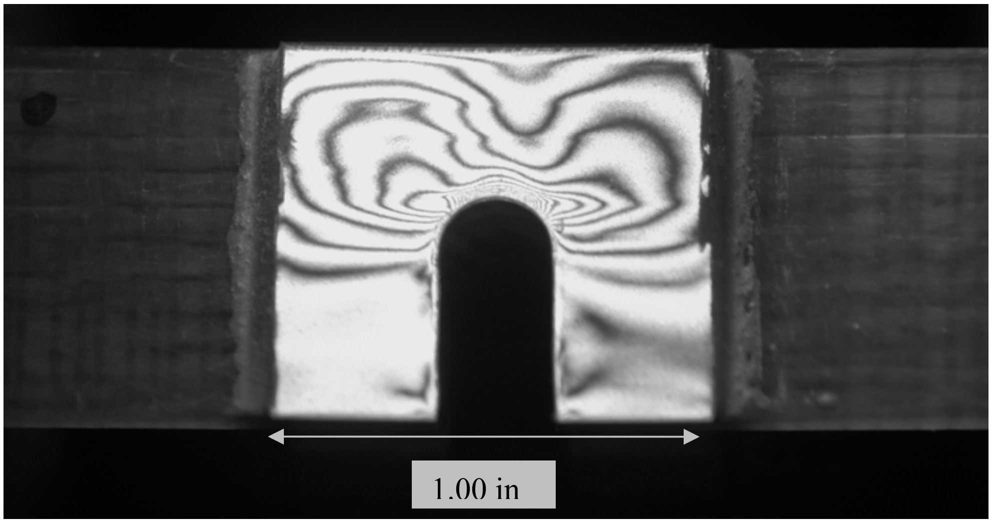

Figure 8: Photoelastic Image of Notched Composite Coupon

Much progress has been made in the field of digital photoelasticity; however, the image analysis necessary to convert these images into a quantitative strain field is not a trivial problem. At the time of this writing digital photoelasticity remains a topic of research and not a ready made solution for this project.

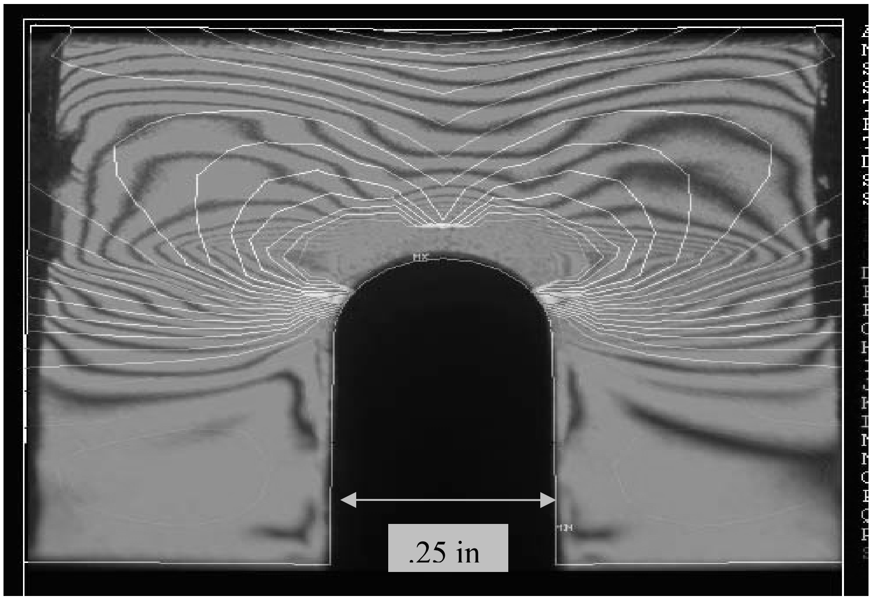

Figure 9: Photoelastic Image Over-Layed with Linear Elastic FEA Model Showing Strain Redistribution of Damaged Coupon

Moiré Inferometry also showed potential. But, much like photoelasticity, this technique was not mature enough for application to this research.

Direct Photographic Strain measurement (also called Digital Image Correlation) uses a high contrast, black and white pattern printed on the sample[7]. A series of grayscale digital images is captured as the sample is tested. Movements of the black and white pattern are measured by digital image analysis. The resulting displacement field is numerically differentiated to give the strain field. Resolution of this technology is no better 10X that of the digital camera being used. A 6 mega-pixel camera, the highest available when this work was completed, does not give high enough resolution to make this a viable alternative. However, recent progress in this technique may make it a viable option in the future.

In the absence of a viable whole field strain measurement technique the problem was approached from the perspective of an iterative procedure. An overview of this is shown in Figure 10.

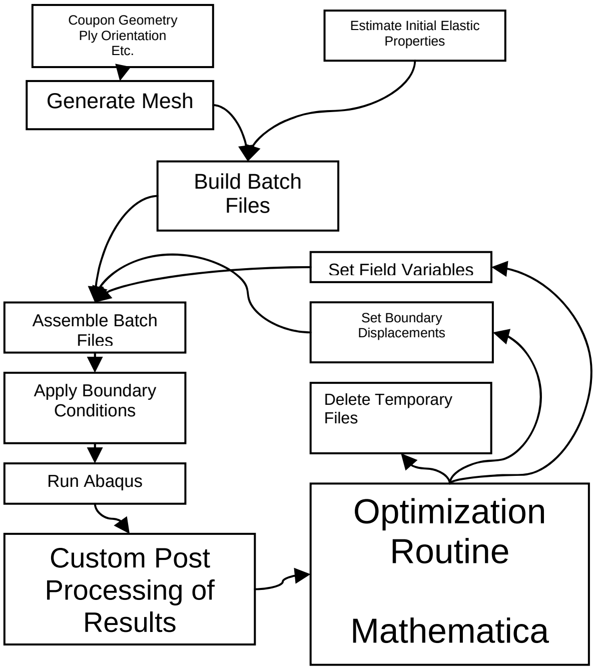

Figure 10: Iterative Procedure Overview

The mathematics program Mathematica has a powerful set of numeric and symbolic capabilities but it also contains fairly comprehensive file handling capabilities[6]. Mathematica was used as the framework and Abaqus FEA codes as well as compiled C++ programs were called from within Mathematica. A custom postprocessor, written in FORTRAN, and compiled as an executable (.exe) was used to collect the necessary finite element data. The Mathematica framework allowed the process to be fully automated. The iterative process was then performed without user intervention. An example of this Mathematica coding is shown in Figure 10.

```
In Order to define a function that executes an external command
one must use the Run[ ] command.

Post:=Run[Abaqus EXTRACT3];
FEA :=Run[Abaqus Job=juice3 interactive];
Build := Run["Abaqus Python BatchApp.py"];
Reset:= Run[Abaqus Python  name.py];
RunFEA:=Do[Reset;Build;FEA;Post,1];
In[36]:=
For[i=1,i<2,i++,Post;Print[i]]
```

Figure 11: Using Mathematica to Call External Programs

Mathematically speaking, many similarities exist between finding Dissipated Energy Density and finding non-linear constitutive behavior. Both can be written as an optimization problem in matrix form as illustrated in Equation 1 [1],[2].

$$[\Sigma_{\text{elements}} \text{ Interpolation Functions}] * \{\text{Nodal Values}\} = \{\text{Experimental Values}\}$$

Equation 1: Equilibrium Equation Form

$$\left| [\mu]\{C\} - \{\Phi\} \right| = \text{minimum}$$

Equation 3: Error Norm for Dissipated Energy Density

$$\left| [\mu]\{C\} - \{E\} \right| = \text{minimum}$$

Equation 4: Error Norm for Non-Linear Material Characterization

Rearranging Equation 1 gives an Error Norm that can be minimized as shown in Equation 3 and Equation 4. In the above equations $[\mu]$ is a matrix of interpolation functions evaluated at the individual element strains and summed over the entire Finite Element model, $\{\Phi\}$ is a vector of experimentally determined total dissipated energies at each data point, and {C} is a vector of the unknown coefficients, i.e., what we are solving for.

For Non-Linear Material Characterization the formulation is in terms of elastic strain energy so {E} is a vector of experimentally determined total strain energies. Minimizing the quantity in equation 3 and 4 gives the best fit for the over-constrained optimization problem.

Now, in general terms, consider the dimensions of this problem to understand why it is solvable. In order to have a tractable optimization problem we must have more equations to be satisfied than unknowns to solve for. If we use, for example, 50 data points in our experimental data the dimensions of the problem are as follows:

[μ]	50 X 125
{C}	125 X 1
{Φ}	50 X 1

In this case there are 125 unknowns and only 50 equations to solve. However, if several laboratory experiments are run with different boundary loading and hence different strain states, then this difficulty can be overcome. If 5 unique sets of experimental data from MSU's Inplane Loader are collected, then the dimensions of the problem become:

[μ]	250 X 125
{C}	125 X 1
{Φ}	250 X 1

This is now a solvable problem because there are 125 unknown variables and 250 equations to solve. We now have an over constrained optimization problem.

Non-Linear Material Characterization

The same mathematics apply to Non-Linear Material Characterization. The process is essentially the same as Dissipated Energy Density with a few variations. First, rather than optimizing for a single parameter we are now looking for the 4 parameters, the three in-plane material stiffnesses and the in-plane poisson's ratio.

$$\begin{gathered} E1\ (\varepsilon 1,\ \varepsilon 2,\ \gamma 12) \\ E2\ (\varepsilon 1,\ \varepsilon 2,\ \gamma 12) \\ G12\ (\varepsilon 1,\ \varepsilon 2,\ \gamma 12) \\ \nu 12\ (\varepsilon 1,\ \varepsilon 2,\ \gamma 12) \end{gathered}$$

If the Poisson's ratio is assumed constant then there are three parameters. The next difference between the DED and the Non-Linear Constitutive formulations is that a linear elastic FEA model cannot be used to predict the element strains within a coupon. This should be obvious because the strains depend on the elastic properties which are non-linear. The result of this conclusion is that the matrix of interpolation functions cannot be assembled before hand and must be done along the way. Recall that this difficulty was circumvented by employing an iterative scheme and rerunning the Finite Element Models with each iteration.

Now consider the dimensions of this problem. Keep in mind that we are looking for three parameters now instead of the single DED parameter.

[μ] 50 X 375
{C} 375 X 1
{E} 1 X 50

This is not solvable with 375 unknowns and 50 equations. If the five unique laboratory experiments are added, the dimensions of the problem are improved but not adequately.

[μ] 250 X 375
{C} 375 X 250
{E} 250 X 1

This remains an under constrained problem with 250 equations and 375 unknowns. Keep in mind that this is the best case where the [μ] is well populated. If the interpolation function matrix is not well populated then the dimensions or the problem are even less favorable than they appear.

CONCLUSIONS AND FUTURE WORK

Despite much work and anxiety on this topic it is the conclusion of the author that a suitable optimization scheme cannot be assembled with the current assumptions, because the dimensions of the problem simply will not allow it. The iterative technique cannot converge to a satisfactory solution.

However, important progress has been made on this topic. The degree of automation required to solve this problem was demonstrated. The computational technology is available and the necessary programming has been demonstrated.

Whole field strain measurement techniques, while not yet mature, are quite promising and it may not be long before they are commercially available.

It is concluded that Non-Linear Material Characterization, at least as presented by this research, can be done but only when whole-field strain data are available.

It is suggested that future work concentrate on two areas. The first is an experimental technique for measuring whole field strains throughout a material test. What is needed is a way to measure the surface strains at discrete point during a test along with boundary forces and displacements as are currently measured.

The second area for future work is to develop the algorithm that takes the strain data as an input and solves the inverse problem to reveal the material constitutive response. Recall, that an overview of this is shown in Figure 7.

Many of the techniques presented in this body of research can be adapted. In fact, from a mathematical perspective, the procedure outlined in figure 7 is a bit easier to implement than the iterative approach that was attempted. Because strains are directly measured, computationally intensive FEA is not required to find the material strains nor is the repeated optimization necessary to include the non-linearity of the problem. The non-linearity is automatically introduced because the non-linear strains are measured directly then used for analysis. In summary, with a more complete data set, the determination of non-linear material properties can be streamlined to yield a tractable problem.

REFERENCES

[1] Mast, P.W., Nash, G.E., Michopoulos, J., Thomas, R.W, Badaliance, R., and Wolock, I., "Experimental Determination of Dissipated Energy Density as a Measure of Strain-Induced Damage in Composites," Naval Research Laboratory report **NRD/FR/6383-92-9369**, April 17, 1992.

[2] Ritter, William, J., "Application of Energy Methods to Modeling Failures in Composite Materials and Structures," M.S. Thesis, Department of Mechanical Engineering, Montana State University, 2004.

[3] Huang, Dong Fang., "In-Situ Material Properties for Composite Materials ," M.S. Thesis, Department of Mechanical Engineering, Montana State University, 2003.

[4] Ramesh,K. "Digital Photoelasticity: Advanced Technologies and Applications," Springer-Verlag Berlin and Heidelberg GmbH & Co. K; Bk&CD-Rom edition (Mar 2000), ISBN-10: 3540667954

[5] Barbero, Ever J. "Digital Introduction to Composite Materials Design," Taylor & Francis, 325 Chestnut Street, Suite 800, Philadelphia, PA 19106. ISBN: 1-56032-701-4, copyright 1999.

[6] Mathematica is symbolic and numerical mathematics software available from Wolfram Research, 1-800-WOLFRAM (965-3726)

www.wolfram.com

[7] Eberl, Christoph."Digital Image Correlation and Tracking," Matlab Central, online database.

www.mathworks.com/matlabcentral/fileexchange/loadFile.do?objectId=12413&objectType=FILE

APPENDIX A:

Sample Codes

Python Scripts

This is a Python code which is used to assemble several batch files and several data files into an input deck that can be run by Abaqus.

```
def BatchApp():
# filesIN: list of input files
# fileOut: output file
# writeOrappend: 'w' for new file to write
#           'a' to append to existing file
      fileOut='juice3.minp'
      filesIN=['batch1.txt','fieldset1.txt','fieldset2.txt','fiel
dset3.txt','batch2.txt']
      writeOrappend='w'
      fidOut=open(fileOut, writeOrappend)
      for file in filesIN:
            fidIN=open(file,'r')
            lines=fidIN.readlines()
            fidIN.close()
            fidOut.writelines(lines)
      fidOut.close()
BatchApp()
```

Abaqus Input File

This is an example an Abaqus input file that is solved with each iteration. Some Node and Element definitions have been removed because they fill many pages of text.

```
*Heading
 linear elastic run
** Job name: juice2 Model name: Model-1
*Preprint, echo=NO, model=NO, history=NO, contact=NO
**

*Node, nset=NALL
      1,        -12.7,        -12.2
      2,    -4.914533,    -2.376212
      3,      -7.9125,        3.825
      4,        -12.7,        3.825
      5,    -1.940969,     1.375864
      6,       -3.125,        3.825
      7,         12.7,        -12.2
      8,     4.914533,    -2.376212
      9,           0.,      -4.0875
     10,           0.,        -12.2
     11,     1.940969,     1.375864
     12,           0.,          0.7
     13,      -7.9125,         12.2
     14,       -3.125,         12.2
 *Element, type=CPS4R
 1,   1,  22, 147,  39
 2,  22,  23, 148, 147
```

```
 3,  23,  24, 149, 148
 4,  24,  25, 150, 149
 5,  25,  26, 151, 150
 6,  26,   2,  27, 151
 7,  39, 147, 152,  38
 8, 147, 148, 153, 152
 9, 148, 149, 154, 153
10, 149, 150, 155, 154
11, 150, 151, 156, 155
12, 151,  27,  28, 156
13,  38, 152, 157,  37
14, 152, 153, 158, 157
15, 153, 154, 159, 158
*Elset, elset=_PickedSet2, generate
   1,  290,    1
** MATERIAL ORIENTATION
*ORIENTATION,NAME=ZERO
0.,1.,0.,-1.,0.,0.
3,0.0
*Solid Section, elset=_PickedSet2, material=E-glass,
ORIENTATION=ZERO
1.,
*Nset, nset=_PickedSet19
   1,   4,  21,  36,  37,  38,  39, 139, 140, 141, 142, 143
*Elset, elset=_PickedSet19
   1,   7,  13,  19,  25, 261, 262, 263, 264, 265, 266
*Nset, nset=_PickedSet21
   7,  17,  18, 104, 105, 106, 107, 108, 131, 132, 133
*Elset, elset=_PickedSet21
 161, 162, 163, 164, 165, 166, 212, 218, 224, 230
**
** MATERIALS
**
*Material, name=E-glass
**Elastic, type=LAMINA
**38260.,9960.,  0.3,4230.,4230.,4230.


*ELASTIC,TYPE=LAMINA,DEPENDENCIES=3
38.26E3,9.96E3,0.30,4.23E3,4.23E3,4.23E3,0.,0,
0,0
3000,9.96E3,0.30,4.23E3,4.23E3,4.23E3,0.,1,
0,0
38.26E3,1.000e0,0.30,4.23E3,4.23E3,4.23E3,0.,0,
1,0
3000,1.000E0,0.30,4.23E3,4.23E3,4.23E3,0.,1,
1,0
38.26E3,9.96E3,0.30,1.000E0,1.000E0,1.000E0,0.,0,
0,1
3000,9.96E3,0.30,1.000E0,1.000E0,1.000E0,0.,1,
0,1
38.26E3,1.000E0,0.30,1.000E0,1.000E0,1.000E0,0.,0,
1,1
3000,1.000E0,0.30,1.000E0,1.000E0,1.000E0,0.,1,
1,1
```

```
**normally there would be one field variable assignment for each
**element in the model.
**most have been removed for conciseness.
*INITIAL CONDITIONS,TYPE=FIELD,VARIABLE=1
"1.     0.001\n\n2.     0.002\n\n3.     0.003\n\n4.
0.004\n\n5.    \
   0.005\n\n6.     0.006\n\n7.     0.007\n\n8.     0.008\n\n9.
\
   0.009\n\n10.    0.01\n\n11.    0.011\n\n12.    0.012\n\n13.
\
   0.013\n\n14.    0.014\n\n15.    0.015\n\n16.    0.016\n\n17.
\
   0.017\n\n18.    0.018\n\n19.    0.019\n\n20.    0.02\n\n21.
\
   0.021\n\n22.    0.022\n\n23.    0.023\n\n24.    0.024\n\n25.
\
   0.025\n\n26.    0.026\n\n27.    0.027\n\n28.    0.028\n\n29.
\"
*INITIAL CONDITIONS,TYPE=FIELD,VARIABLE=2
fieldset2
*INITIAL CONDITIONS,TYPE=FIELD,VARIABLE=3
fieldset3
**
** BOUNDARY CONDITIONS
**
** Name: left side Type: Displacement/Rotation
*Boundary
_PickedSet19, 1, 1
_PickedSet19, 2, 2
_PickedSet19, 6, 6
** --------------------------------------------------------------
--
**
** STEP: Load
**
*Step, name=Load, nlgeom=YES
*Static
1., 1., 1e-05, 1.
**FIELD,USER
**NALL,
**FIELD,USER,VARIABLE=2
**NALL,
**FIELD,USER,VARIABLE=3
**NALL,
**
** BOUNDARY CONDITIONS
**
** Name: right side Type: Displacement/Rotation
*Boundary
_PickedSet21, 1, 1, 1.01
_PickedSet21, 2, 2
_PickedSet21, 6, 6
**
** OUTPUT REQUESTS
```

```
**
*Restart, write, frequency=1
**
** FIELD OUTPUT: F-Output-1
**
*Output, field, variable=PRESELECT
**
** HISTORY OUTPUT: H-Output-1
**
*Output, history, variable=PRESELECT
**Node Print, freq=1
*EL PRINT, FREQUENCY=1
E11,S11,FV1,FV2,FV3
*EL FILE, FREQUENCY=1, POSITION=INTEGRATION POINTS
S,E,FV
**NODE FILE, FREQ=1
*End Step
```

Post Processing

This is an example of the custom post processor used to get certain data from an Abaqus Data file. This script is written in FORTRAN and must be compiled to an .exe file before it can be used.

```
      SUBROUTINE HKSMAIN
C=================================================================
C This program must be compiled and linked with the command:
C     abaqus make job=EXTRACT5
C Run the program using the command:
C     abaqus EXTRACT5
C=================================================================
C=============================================================
C      THIS PROGRAM EXTRACTS THE VALUES OF STRESS AND STRAIN
C      FROM THE .fil FILE
C=============================================================
C
C
       INCLUDE 'aba_param.inc'
       DIMENSION ARRAY(513), JRRAY(NPRECD,513), LRUNIT(2,1)
       EQUIVALENCE (ARRAY(1), JRRAY(1,1))
C
       DIMENSION S(6), E(6), FIELD(3)
       CHARACTER FNAME*80
       CHARACTER STRESS*80
       CHARACTER STRAIN*80
       CHARACTER ELEMENT*80
       CHARACTER FIELDV*80
C
C=============================================================
C      START A LOOP TO COUNT THROUGH ALL 7 FEA MODELS
C=============================================================
       DO 1002 INDEX = 1, 7
C=============================================================
C      SET THE NAME OF THE RESULTS(.FIL) FILES HERE
C      ALSO SET THE NAMES OF THE OUTPUT FILES
C=============================================================
       IF (INDEX .EQ. 1) THEN
       FNAME = 'juice1'
       STRESS = 'stress1.dat'
       STRAIN = 'strain1.dat'
       ELEMENT = 'element1.dat'
       FIELDV = 'field1.dat'
       ELSE IF (INDEX .EQ. 2) THEN
       FNAME = 'juice2'
       STRESS = 'stress2.dat'
       STRAIN = 'strain2.dat'
       ELEMENT = 'element2.dat'
       FIELDV = 'field2.dat'
       ELSE IF (INDEX .EQ. 3) THEN
       FNAME = 'juice3'
       STRESS = 'stress3.dat'
       STRAIN = 'strain3.dat'
       ELEMENT = 'element3.dat'
       FIELDV = 'field3.dat'
       ELSE IF (INDEX .EQ. 4) THEN
       FNAME = 'juice4'
       STRESS = 'stress4.dat'
       STRAIN = 'strain4.dat'
```

```
      ELEMENT = 'element4.dat'
      FIELDV = 'field4.dat'
      ELSE IF (INDEX .EQ. 5) THEN
      FNAME = 'juice5'
      STRESS = 'stress5.dat'
      STRAIN = 'strain5.dat'
      ELEMENT = 'element5.dat'
      FIELDV = 'field5.dat'
      ELSE IF (INDEX .EQ. 6) THEN
      FNAME = 'juice6'
      STRESS = 'stress6.dat'
      STRAIN = 'strain6.dat'
      ELEMENT = 'element6.dat'
      FIELDV = 'field6.dat'
      ELSE IF (INDEX .EQ. 7) THEN
      FNAME = 'juice7'
      STRESS = 'stress7.dat'
      STRAIN = 'strain7.dat'
      ELEMENT = 'element7.dat'
      FIELDV = 'field7.dat'
C
C
C
      ENDIF
C=============================================================
C     OPEN THE OUTPUT FILES
C=============================================================
      OPEN(UNIT=9,FILE=STRESS,STATUS='REPLACE')
      OPEN(UNIT=3,FILE=STRAIN,STATUS='REPLACE')
      OPEN(UNIT=7,FILE=ELEMENT,STATUS='REPLACE')
      OPEN(UNIT=4,FILE=FIELDV,STATUS='REPLACE')
      NRU = 1
      LOUTF = 0
      LRUNIT(1,1) = 8
      LRUNIT(2,1) = 2
C
      CALL INITPF(FNAME,NRU,LRUNIT,LOUTF)
C
      JUNIT = 8
C
C     CALL DBRU(JUNIT)
C=============================================================
C     READ RECORDS FROM THE RESULTS FILE(.fil) AND PROCESS
C     THE DATA.  COVER A MAXIMUM OF 10 MILLION RECORDS
C=============================================================
      DO 1000 K100 = 1, 100
      DO 1000 K1 = 1, 99999
             CALL DBFILE(0,ARRAY,JRCD)
             IF (JRCD .NE. 0) GO TO 1001
             KEY = JRRAY(1,2)
C
C=============================================================
C     GET THE HEADING (TITLE) RECORD.
C=============================================================
C     IF (KEY .EQ. 1922) THEN
C            WRITE(9,1100) (ARRAY(IXX),IXX=3,12)
C 1100       FORMAT(1X,10A8)
C
C=============================================================
C     GET THE CURRENT STEP AND INCREMENT NUMBER
C=============================================================
C     ELSE IF (KEY .EQ. 2000) THEN
```

```
C            WRITE(9,1200)        JRRAY(1,8), JRRAY(1,9)
C 1200       FORMAT(1X,'** STEP ',I2,'    INCREMENT ',I3)
C
C============================================================
C     Get the element and integration point numbers, JELNUM and INTPN,
C     and the location of INTPN(0--at int. pt., 1--at centroid,
C     4--nodal average) and the number of direct and shear components
C     in the analysis
C============================================================
      IF (KEY .EQ. 1) THEN
             JELNUM = JRRAY(1,3)
             INTPN  = JRRAY(1,4)
             LOCATE = JRRAY(1,6)
             NDI    = JRRAY(1,8)
             NSHR   = JRRAY(1,9)
          NDIR1  = NDI + 1
             IF(LOCATE.LE.1) THEN
                    WRITE(7,*) JELNUM
C            ELSE IF(LOCATE.EQ.4) THEN
C                   WRITE(9,1191) JELNUM, NDI,NSHR
C 1191              FORMAT(2X,'NODE NUMBER = ',I8,5X,
C     1                                 'NDI/HSHR = ',2I2)
             END IF
C
C============================================================
C     THE STRESS TENSOR
C============================================================
      ELSE IF (KEY .EQ. 11) THEN
C
             DO 10 IXX = 1, NDI + NSHR
                    S(IXX) = ARRAY(IXX+2)
   10        CONTINUE
             WRITE(9,1203) (S(IZZ), IZZ = 1, NDI + NSHR)
 1203        FORMAT(4X,E12.5,' ',E12.5,' ',E12.5,' ',E12.5,'
',E12.5,E12.5)
C            DO 20 IYY = NDI + 1, NSHR + NDI
C                   S(IYY) = ARRAY(IYY+2)
C  20        CONTINUE
C            WRITE(9,1204) (S(IZZ), IZZ = NDI + 1, NSHR + NDI)
C 1204       FORMAT(4X,E12.5,E12.5,E12.5)
C=============================================================
C     GET THE STRAIN TENSOR
C=============================================================
      ELSE IF (KEY .EQ. 21) THEN
C            WRITE(3,2202)
C 2202       FORMAT(3X,'STRAINS:')
C
             DO 30 IXX = 1, NDI + NSHR
                    E(IXX) = ARRAY(IXX+2)
   30        CONTINUE
             WRITE(3,2203) (E(IZZ), IZZ = 1, NDI + NSHR)
 2203        FORMAT(4X,E12.5,' ',E12.5,' ',E12.5,' ',E12.5,'
',E12.5,E12.5)
C            DO 40 IYY = NDI + 1,  NSHR + NDI
C                   E(IYY) = ARRAY(IYY+2)
C  40        CONTINUE
C            WRITE(8,2204) (E(IZZ), IZZ = NDI + 1, NSHR + NDI)
C 2204       FORMAT(4X,'E12 = ',E12.5,'  E13 = ',E12.5,'  E23 = ',E12.5)
C
C===============================================================
C     GET THE 3 FIELD VARIABLES
C===============================================================
```

```
      ELSE IF (KEY .EQ. 9) THEN
             DO 50 IXX = 1, 3
                    FIELD(IXX) = ARRAY(IXX + 2)
   50        CONTINUE
             WRITE(4,2005) (FIELD(IZZ), IZZ = 1, 3)
 2005        FORMAT(4X,E12.5,' ',E12.5,' ',E12.5)
C===============================================================
C      SHUT THE LOOP DOWN
C===============================================================
      END IF
C
 1000  CONTINUE
 1001  CONTINUE
      CLOSE (UNIT=9)
      CLOSE (UNIT=3)
      CLOSE (UNIT=7)
      CLOSE (UNIT=4)
 1002  CONTINUE
C
      RETURN
      END
```