Torque transducer sensitivity study

HARI NARAYANAN SOUNDARARAJAN

KTH Industrial Engineering
and Management

Master of Science Thesis
Stockholm, Sweden 2014

Torque transducer sensitivity study

Hari Narayanan Soundararajan

Master of Science Thesis MMK 2014:01 MKN 099
KTH Industrial Engineering and Management
Machine Design
SE-100 44 STOCKHOLM

Examensarbete MMK 2014:01 MKN 099

Känslighetsstudie av momentgivare

KTH Industriell teknik och management

Hari Narayanan Soundararajan

Godkänt 2014-01-10	Examinator Ulf Sellgren	Handledare Ulf Sellgren
	Uppdragsgivare Atlas Copco Industrial Technique	Kontaktperson Per Forsberg

Sammanfattning

Momentgivare är små sensorer som används för att mäta och registrera vridmomentet på en roterande axel. Momentgivare baseras vanligtvis på trådtöjningsmätare för att mäta vridmomentetet kring axeln de är fixerade på.

Det skall noteras att en ideal momentgivare endast bör mäta den yttöjning som uppkommer på grund av ett vridmoment. Den belastning som uppkommer på grund av axelns böjning kompenseras bort med hjälp av en Wheatstonebrygga. Men på grund av geometriska toleranser och monteringsfel kan kompensationen bli felaktig och då påverkas det uppmätta momentet även av böj och axialbelastningar vilket är oönskat.

En analytisk modell har utvecklats med hjälp av Matlab och denna rapport undersöker de olika lastfallens bidrag till momentfelet. Användaren kan ange området där yttöjningen skall beräknas och utifrån de beräkningarna bestämma var trådtöjningsgivarna bör placeras. Inledningsvis är beräkningarna baserade på en av Atlas Copcos momentgivare och sedan har generaliserade resultat utvecklats. De teoretiska beräkningarna verifieras med hjälp av programmet ProEngineer Mechanica.

Användaren kan ange vilka belastningar som axeln känner samt de geometriska toleransvärdena och modellen beräknar då ytspänningen för den valda regionen. Det huvudsakliga syftet med denna avhandling är att skapa en bättre förståelse av ytspänningen som uppkommer på grund av vridning, böjning och axiella belastningar och även geometriska imperfektioner. Modellen kan också användas för att göra beräkningar som visar var töjningsgivare bör placeras på axeln för maximal precision. Slutligen har de introducerade felet från de olika möjliga konfigurationerna jämförts och en slutsats har dragits baserat på faktorförsök.

*Keywords: faktorförsök**,** geometriska toleranser, känslighet, töjningsgivare, ytspänning*

KTH Industrial Engineering and Management	**Master of Science Thesis MMK 2014:01 MKN 099**
	Torque transducer sensitivity study
	Hari Narayanan Soundararajan

Approved 2014-01-10	Examiner Ulf Sellgren	Supervisor Ulf Sellgren
	Commissioner Atlas Copco Industrial Technique	Contact person Per Forsberg

Abstract

A torque transducer or a torque sensor is a device for measuring and recording the torque on a rotating system. Torque transducers usually employ strain gauges to measure the torsional moment applied to a rotating shaft.

It is to be noted that for an ideal torque transducer, it should measure only the strain that is caused by a torque. Strain due to bending load should be compensated as per the Wheatstone bridge arrangement. However, because of geometrical tolerances and assembly errors, the compensation doesn’t occur and the measured strain is a resultant of bending loads and axial loads which are undesired to measure the torque associated with the system.

An analytical formulation has been developed using Matlab and this thesis gives the generalized indication of the strain due to all the associated loads. The user shall also entire the region where the strain needs to be computed and this knowledge can be useful for placing the strain gauges in the shaft accordingly. Initially, the formulation is based on a standard Torque Transducer used at Atlas Copco and then, a generalized result has been developed. The theoretical formulation is verified using the ProEngineer Mechanica software.

The end user shall enter the different loads (if any) along with the geometrical tolerance values and the output will be an indication of the strain at point, strain at a region and sensitivity. The main intention of the thesis is to create a better understanding of the strain associated with the twisting, bending and axial loads and also the geometrical imperfections. The user can also make a decision on the location of strain gauges on a shaft for maximum accuracy. Finally, the differences in error from different possible configurations are compared and a conclusion has been made based on factorial design pertaining to design of experiments.

Keywords: design of experiments, engineering strain, geometrical tolerances, sensitivity, strain gauges,

FOREWORD

I would like to thank Atlas Copco for giving me an opportunity to carry out my master thesis at their Research & Development division at Nacka, Sweden. I must say it was a memorable experience carrying out my thesis work at their head office.

I wouldn't have completed this thesis without the help and guidance of my supervisor Per Forsberg, who is a mechanical engineering designer at Atlas Copco Industrial Technique, Stockholm.

Big thanks goes to Andris Danebergs, Ingemar Sjors, Johan Nasell, Hakan Lindstrom and Erik Persson, who were instrumental in providing me with inputs and suggestions throughout this thesis work at Atlas Copco.

I would like to thank my supervisor at KTH, Ulf Sellgren, from whom I have learnt a lot in my Masters. Thank you Sir! It was a wonderful learning experience at KTH.

Tack så mycket Arvid Blom, för din hjälp med att översätta min abstract på svenska.

Lastly, I would like to thank Parents (very rarely I do) and my new friends in Sweden, who were instrumental in encouraging me, motivating me and supporting me throughout my stay in Sweden!

And yes, thanks to all the good people around the world! Live long and Prosper _\\V/

Hari Narayanan Soundararajan

Stockholm, January, 2014

Notations

Symbol	*Description*
A	Area of the object (mm^2)
D	Diameter of the outer shaft (mm)
d	Diameter of the inner hole (mm)
d'	Perpendicular distance between the two axes (mm)
dA	An element of small area (mm^2)
E	Young´s modulus (Pa)
e	Offset between the inner and the outer diameter (mm)
F, P	Axial load (N)
F_a	Axial gage factor
F_t	Transverse gage factor
I_x	Second moment of the inertia about x axis (mm^4)
$I_{x'}$	Second moment of the inertia about x' axis (mm^4)
I_y	Second moment of the inertia about y axis (mm^4)
$I_{x'}$	Second moment of the inertia about y' axis (mm^4)
I_{zz}	Moment of Inertia in z plane (mm^4)
J	Polar moment of inertia (mm^4)
M_z	Bending moment (Nmm)
n_e	Error in strain due to misalignment (με)
Q	First moment of small area about the neutral axis of the entire body (mm^3)
S_p	Polar section modulus (mm^3)
T	Torsion (indicated in Nmm)
t	thickness across the interested point (mm)
Unit - μE	Refers to micro-strain or $\mu\varepsilon$, $\mu\epsilon$
V	Supply Voltage (V)
V_{bridge}	Voltage across the bridge (V)
W, V_y	Bending Load (N)
x	distance between the intended axis and the centroid (mm)
$Xbar$	Centroid x (mm)
x_{offset}	Eccentricity in x direction (mm)
y	Vertical distance from the bending load towards the interested point (mm)
$Ybar$	Centroid y (mm)

y_{offset}	Eccentricity in y direction (mm)
y_Q	y distance between the intended axis and the centroid (mm)
$\gamma_{torsion}$, γ_{xy}	Shear strain due to torsion (με)
ε	Engineering strain (με)
$\varepsilon_{1,2}$	Maximum and minimum principal strain (με)
ε_1, ε_2, ε_3, ε_4	Strain in different regions (με)
ε_a	Strain parallel to gage axis, or the gridlines in the gage (με)
ε_{angle}	Strain along desired angle (με)
ε_t	Strain perpendicular to the gage axis, or the gridlines in the gage (με)
ε_{tt}	Strain at phi plane along phi direction (με)
ε_{zz}	Strain at z plane along z direction (με)
θ_p	Principal plane angle (degrees)
ν	Poisson's ratio
v_o	Manufacturer's gauge factor
$\sigma_{1,2}$	Maximum and minimum principal stress (MPa)
σ_{axial}, σ_a	Normal stress due to axial load (MPa)
$\sigma_{bending}$, σ_b	Normal stress due to bending load (MPa)
$\tau_{bending}$, τ_b	Shear stress due to bending load (MPa)
$\tau_{max,min}$	Maximum and minimum shear stress (MPa)
$\tau_{torsion}$	Shear stress due to torsion (MPa)
$\tau_{torsion}$, τ	Shear stress due to torsional load (MPa)
ϕ	Twist per unit length of the shaft (degrees)
$\frac{\Delta R}{R}$	Change in resistance to resistance in the Wheatstone bridge

Abbreviations

AO	Angle Orientation
AI	Angle Inclination
CAD	Computer Aided Design
DOE	Design of Experiments
FEA	Finite Element Analysis
GF	Gauge Factor
LD	Length Displacement
MOI	Moment of Inertia

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

1 INTRODUCTION

This chapter presents the background, purpose, delimitations and a brief description of the method used in this project.

1.1 Background

A torque transducer or a torque sensor is a device for measuring and recording the torque on a rotating system. Commonly, torque sensors or torque transducers use strain gauges applied to a rotating shaft or axle. With this method, a means to power the strain gauge bridge is necessary, as well as a means to receive the signal from the rotating shaft (Kumar, 2011).

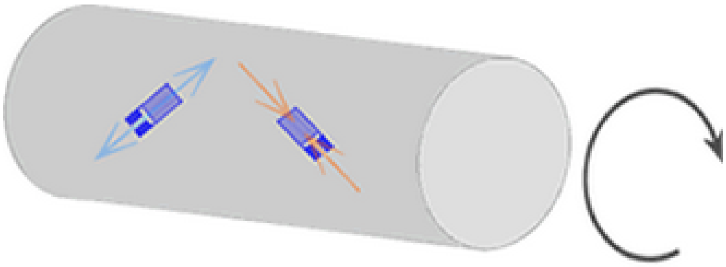


Figure 1 Torque Transducer

Many of Atlas Copco's products use torque transducers to measure the torque. Atlas Copco's products are getting smaller and that gives new problems to the torque measurement quality. To increase the knowledge in this area, a sensitivity study was needed that can correlate the relation between the measured strain in a strain gauge and the actual torque. The thesis also focuses on identifying parameters that can influence the engineering strain measurement and also numerically quantifies their effects.

The orientation of the strain gauge is necessary as this would have a significant impact in the measurement of the strain and subsequently the torque. A slight angular deviation might create a significant difference in measurement of the strain. Also the selection of optimal number of strain sensing elements can result in betting accuracy. But there is no clear understanding of these factors and their impact on the strain measurement.

The most important factor that can negatively influence the strain measurement is the offsets of the inner hole because of geometrical tolerances. Though it was physically observed that these eccentricities can create a significant output difference, there hasn't been any mathematical relation carried out previously that could state the impact of these unwanted errors.

Another important parameter that is considered in the thesis is the impact of bending and axial loads on the strain measurement. Normally, a bending and axial compensation is evident on the torque transducer. However, because of geometrical tolerances, these loads can create a significant difference in the strain gauge measurement and the output will be a result of these loads as well as torsion as opposed to the ideal case of measurement caused by torsion. An analysis of the stress that is caused by these loads was necessary to see their impact on the torque measurement.

In short, if there are no imperfections, then the strain gauge arrangement will give a sensitivity value which is worked further to obtain a torque value (which shall be the torque operating on the shaft). However, if there are imperfections, then the strain gauge arrangement will give a different sensitivity value which will show a different torque value from the actual torque applied on the shaft.

1.2 Purpose

An ideal torque transducer should measure only the strain that is caused by a torque. Strain caused by bending load or an axial load should be compensated as per the Wheatstone bridge arrangement (Schicher, 2002). However, because of geometrical tolerances and assembling errors, the compensation is not accurate and the measured strain is a resultant of bending loads and axial loads which are undesired to measure the torque associated with the system. Since Atlas Copco's transducers are getting smaller and smaller, there is a need for accuracy in these arrangements.

Conventionally, sensitivity computation was carried out for a perfect concentric shaft subjected to torsion alone. In this thesis, sensitivity can be computed for an ideal as well as imperfect shaft.

The purpose of this thesis is mainly concerned with making an analysis of strain measurement and also answering a few questions with reference to the tolerances, orientation accuracy of the strain gauges, number of strain sensing elements and the effect of bending and axial loads. A faster analytical model with a good accuracy can usurp the FEA because of the fact that a lot of time is spent in FEA for analyzing the strain. DOE was performed so that the effects can be numerically quantified as well.

1.3 Scope and Delimitations

The intended result from this thesis is that finally, a better understanding of strain gauges can be made with respect to different errors and unwanted effects. A numerical computation of the strain indication because of unwanted cases (like external bending load, axial load, and geometrical tolerances) is made using Matlab and simultaneous verification using FEM is made. The unwanted effects are also compared and their individual and combined effects are numerically categorized using DOE.

Therefore, the backbone of this thesis lies in solid mechanics and strength of materials. A lot of emphasis was made on the neutral axis and the corresponding centroids, neutral axis and the moments of inertia. Geometrical tolerance related errors are also analysed and a numerical calculator was scripted using Matlab. As a result, a certain offset shall bring in a different neutral axis and the entire computation was made assuming the fact that the load is always applied on the neutral axis. This shall be discussed in detail in Chapter 3.5.2.

1.4 Expected results and conclusions

Using the Matlab code, the end user shall find out the strain that would occur over any point or over any region depending upon his/her interest.

- ✓ A Matlab Program
- ✓ Verification with ProE Mechanica (FEA Software)
- ✓ The Matlab program should show the following
    - o Sensitivity and Strain output at different regions – with/without imperfections
    - o Plots showing the strain over a region where the strain gauges are to be placed
- ✓ Conclusion stating the optimal location of Strain gauges in the transducer
- ✓ Factorial design considering major parameters in assembly errors and their influence using DOE

1.5 Method

The basic approach for this thesis was to finally end with a generalized model that can hold good for almost all the torque transducers at Atlas Copco.

The first step involves writing a Matlab Script that can find the stress and corresponding strain at any point on the outer diameter's surface of the torque transducer. It is quite direct to calculate the stresses for a shaft subjected to only torsion as the stress will be the same at all points. But for the same shaft, when it is subjected to axial and bending load, it requires analytical computations to calculate the stresses at different points. The highest complexity arises, when the inner diameter is slightly offset from the outer diameter. In this case, many parameters like the moment of inertia, centroid and area moment of inertia changes. A Matlab script considering all these parameters is written with the output being the stress and strain at the interested point. Then, strain over an interested region is calculated and the verification is carried out using ProEngineer Mechanica. If the results match, then conclusions are to be made out. Then using Matlab, Stresses and Strain over a region is plotted against various parameters such as sensitivity, loads, offsets, etc. Factorial design is also carried out in order to numerically quantify the effects of various parameters with the output being the sensitivity caused because of those unwanted parameters. This shall give a clear understanding of the various parameters and their impact against the ideal case.

Once the above steps are calculated, then a correlation between the strain (obtained from the Matlab script) and the output sensitivity that a Wheatstone bridge shall show for different configurations is obtained. Finally the results are to be compared with an existing torque transducer and the conclusions are made.

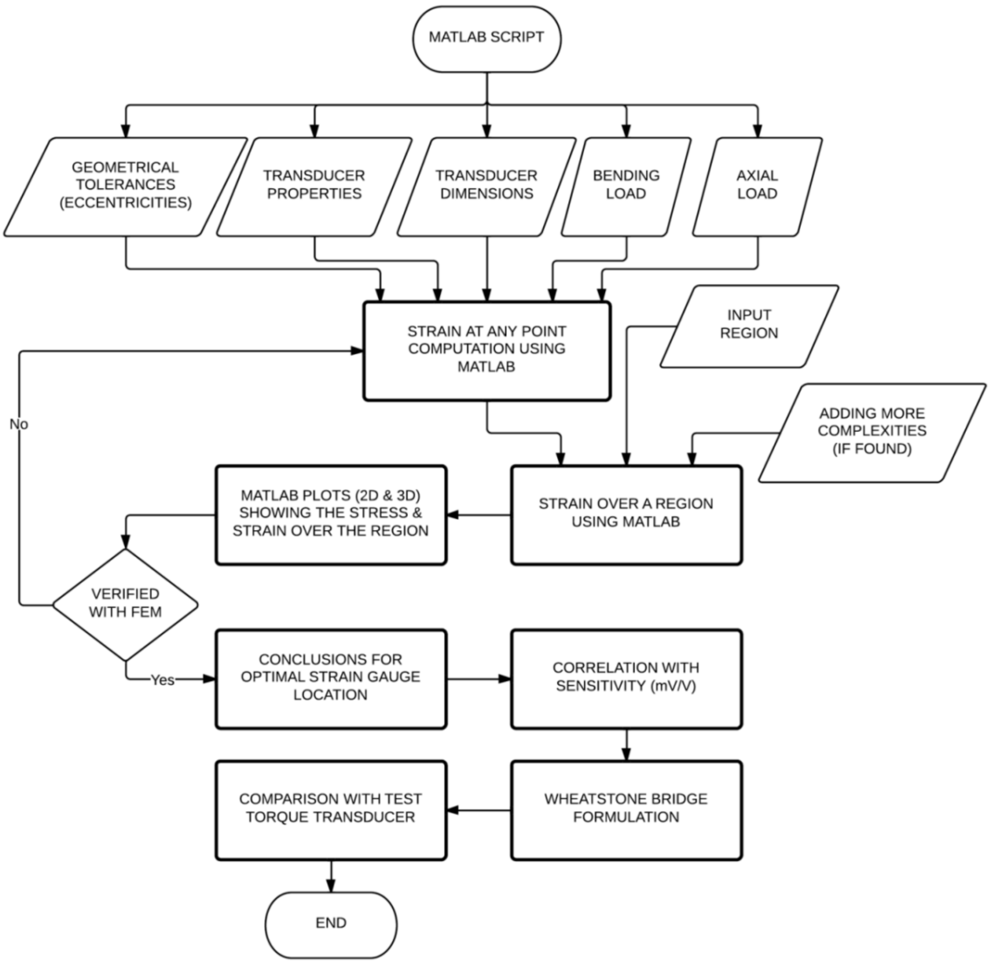


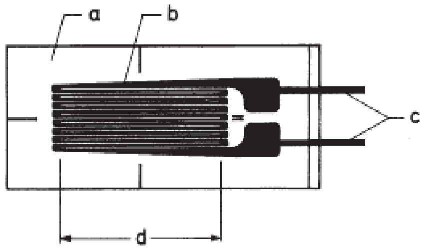
Figure 2 Flow Chart indicating Method of the Thesis

The reference frame is a summary of the existing knowledge and former performed research on the subject. This chapter presents the theoretical reference frame that is necessary for the performed research.

2.1 Strain gauges and measurement using strain gauges

Engineering strain is defined as the ratio of the change in length to the initial unstressed reference length. A strain gage (Figure 3) is the element that senses this change and converts it into an electrical signal. When wire is stretched, its cross-sectional area decreases and therefore its resistance increases. Figure 3 shows a strain gauge (Hoffman, 1989).

The metallic strain gauge consists of a very fine wire or a metallic foil arranged in a grid pattern (Muftah, 2010). The grid pattern maximizes the amount of metallic wire or foil subject to strain in the parallel direction (shown as the effective grid length). The cross sectional area of the grid is minimized to reduce the effect of shear strain and Poisson strain.



a carrier material

b measuring grid

c connections

d effective grid length

Figure 3 Strain Gauge

2.2 Wheatstone bridge and sensitivity

A Wheatstone bridge is an electrical circuit used to measure an unknown electrical resistance by balancing two legs of a bridge circuit, one leg of which includes the unknown component. If all four resistor values and the supply voltage (V_{EX}) are known, the voltage across the bridge (V_O) can be found by working out the voltage from each potential divider and subtracting one from the other (National Instruments, 1998).

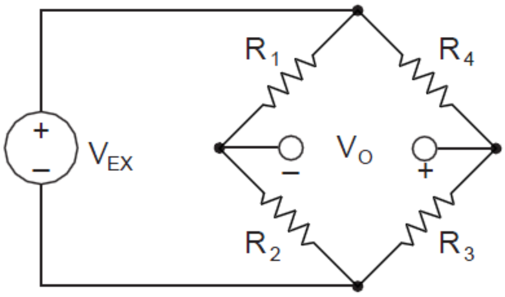


Figure 4 Wheatstone bridge

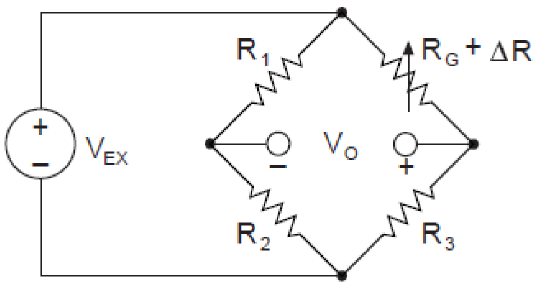
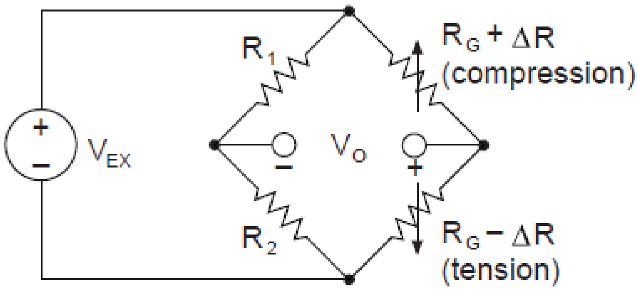
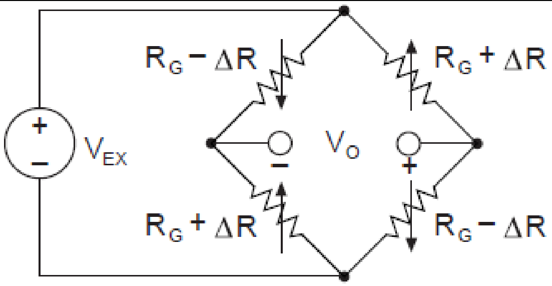
$$V_o = \left(\frac{R_3}{R_3+R_4} - \frac{R_2}{R_1+R_2}\right) \times V_{EX} \quad (1)$$

Strain gauge transducers usually employ four strain gauge elements electrically connected to form a Wheatstone bridge circuit. The sensor can occupy 1/2/3/4 arms of the bridge, depending on the application. So when R1, R2, R3 & R4 are balanced there in no V_o (National Instruments, 1998). If R4 acts as element and is strained, V_o is obtained usually in mV, which is an indirect measurement of the Strain.

Some significant results of Wheatstone bridge are as follows:

- For constant supply voltage V_{EX} and constant strain gage factor, axial strain at the location of the strain gage is a linear function of the output voltage from the Wheatstone bridge circuit.
- For known values of Strain Gauge Factor and V_{EX}, the actual value of the strain can be calculated from the equation (1) after output voltage Vo is known.

Table 1 Wheatstone bridge configuration (National Instruments, 1998)

Connection	Figure	$\frac{V_O}{V_{EX}} =$
Quarter Bridge		$-\frac{GF \times \varepsilon}{4}\left(\frac{1}{1 + GF\frac{\varepsilon}{2}}\right)$
Half Bridge		$-\frac{GF \times \varepsilon}{2}$
Full Bridge		$-GF \times \varepsilon$

2.3 Strain gauge Measurement system

The engineering strains measured with strain gages are normally very small. Consequently the changes of resistance are also very small and cannot be measured directly with a device like an ohmmeter. The strain gage must therefore be included in a measurement system where precise determination of the strain gage's change of resistance is possible (Hoffman, 1989).

The first component in the system is formed by the strain gage itself. It converts the mechanical strain into a change in the electrical resistance.

The second component in the system is a measuring circuit, shown here as a Wheatstone bridge having the strain gage as one arm. Energy must be passed to them to obtain a useful signal. This auxiliary energy is taken from a separate source. When the strain gage's resistance changes because of a strain, the bridge circuit loses its symmetry and becomes unbalanced. A bridge output voltage is obtained which is proportional to the bridge's unbalance.

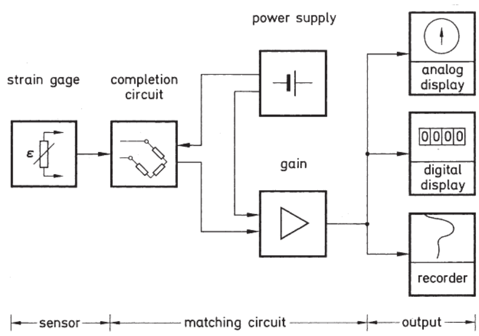


Figure 5 Strain gauge measurement system (Hoffman, 1989)

An amplifier is included in the measuring system as the third component which amplifies the bridge output voltage to a level suitable for indicating instruments. Sometimes amplifiers are designed to give an output current proportional to the bridge output voltage, but some models can provide either voltage or current outputs. With a linear amplifier the output voltage or output current is proportional to the amplifier input voltage which is also the bridge output voltage and this is in turn proportional to the measured strain.

The fourth component in the measuring system is the display mainly for user interface. In the simplest case, the measurement is displayed by the indicating scale of a voltmeter or ammeter or the figures on a digital measuring device. If the change of strain with time is needed as in a dynamic process, recording instruments are better suited than indicating ones. Many amplifiers enable the connection of both types of instrument, either as an alternative or in parallel connection

2.4 Solid-Mechanics terms

The backbone of this thesis lies in solid mechanics. Some of the terms that are used throughout the thesis are given below. A detailed explanation is given in Chapter 3.

- Centroid

The point at which we assume the area concentrated is called the centroid and the point at which the mass is assumed to be concentrated is called the center of gravity. (Timoshenko, 1940).

- Neutral Axis

The neutral axis is an axis in the cross section of a beam (a member resisting bending) or shaft along which there are no longitudinal stresses or strains. If the section is symmetric, isotropic and is not curved before a bend occurs, then the neutral axis is at the geometric centroid. For an ideal hollow shaft, the centroid is located at the geometric center of the hollow cylinder.

- Statistical moment of Area or the First Moment of Inertia

The static or statical moment of area, usually denoted by the symbol Q, is a property of a shape that is used to predict its resistance to shear stress. (Vable, 2009).

$$Q = \int y \, dA \tag{2}$$

Q - First moment of the small area about the neutral axis of the entire body
dA - an elemental area of small area;
y - The perpendicular distance to the element dA from the neutral axis or the Centroid.

- Second Moment of Inertia

The second moment of area, also known as moment of inertia of plane area, is a geometrical property of an area which reflects how its points are distributed with regard to an arbitrary axis.

$$I_x = \iint y^2 dxdy \tag{3}$$

$$I_y = \iint x^2 dydx \quad (4)$$

I_x - Second moment of the inertia about X axis
x – Distance between the intended axis and the centroid
I_y - Second moment of the inertia about Y axis
y – Distance between the intended axis and the centroid

- Perpendicular Axis Theorem

The perpendicular axis theorem (or plane figure theorem) can be used to determine the moment of inertia of a rigid object that lies entirely within a plane, about an axis perpendicular to the plane, given the moments of inertia of the object about two perpendicular axes lying within the plane. The axes must all pass through a single point in the plane. J is the polar moment of inertia, which passes through an axis perpendicular to I_x and I_y (Khurmi, 2008).
Polar Moment of Inertia $J = I_x + I_y$

- Parallel Axis Theorem

Parallel axis theorem or Huygens–Steiner theorem can be used to determine the second moment of area or the mass moment of inertia of a rigid body about any axis, given the body's moment of inertia about a parallel axis through the object's centroid and the perpendicular distance (r) between the axes (Khurmi, 2008).

$$I_{x'} = I_x + (A \times d^2) \quad (5)$$

$I_{x'}$ - Second moment of the inertia about x′ axis
I_x - Second moment of the inertia about x axis (along its centroid)
d – Perpendicular distance between the two axes
A – Area of the object

2.5 Combined loading

The following figure represents a case where an ideal shaft is subjected to Twisting, Bending and Axial Load (Vable, 2009). Figure 6 shows a shaft with four points A,B, C and D. Along the point A, the bending load acts which will be result in a tensile stress whereas on the point B, the bending load will create compression. The axial load is acting along the centroid (or the neutral axis) of the hollow shaft. The twisting moment acts in an anticlockwise rotation. Table 2 shows the stresses that are acting on the four points.

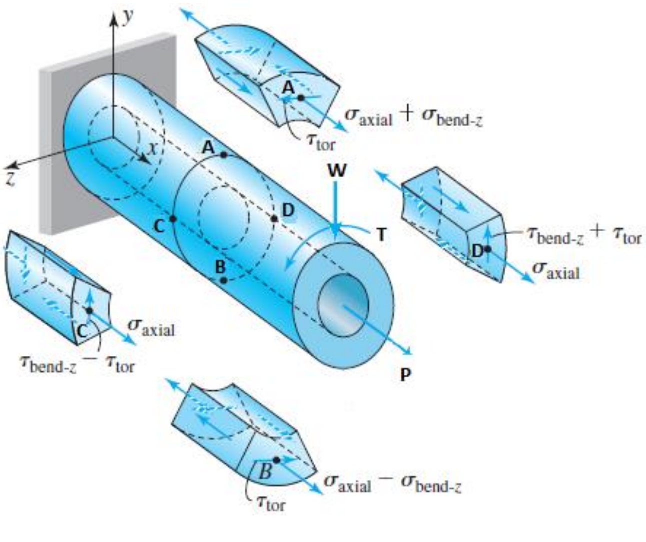


Figure 6 Combined loading in shafts

Table 3 shows the normal and the shear stresses for a combined load. The axial load is acting along the x direction while the bending load is applied along the y direction. Figure 6 shows a shaft when subjected to axial, bending and twisting load.

Table 2 Normal and Shear Stresses at different points on the shaft

Point	Normal Stresses	Shear Stresses
A	$\sigma_{\text{axial}} + \sigma_{\text{bending}}$	τ_{torsion}
B	$\sigma_{\text{axial}} - \sigma_{\text{bending}}$	τ_{torsion}
C	σ_{axial}	$\tau_{\text{bending}} + \tau_{\text{torsion}}$
D	σ_{axial}	$\tau_{\text{bending}} - \tau_{\text{torsion}}$

Table 3 Normal and Shear stresses formulas in combined loading (Vable, 2009)

Load	Normal Stress	Shear Stress
Axial	$\sigma_{xx} = \frac{N}{A}$; $\sigma_{yy} = 0$; $\sigma_{zz} = 0$	$\tau_{xy} = 0$; $\tau_{yz} = 0$; $\tau_{xz} = 0$
Torsion	$\sigma_{xx} = 0$; $\sigma_{yy} = 0$; $\sigma_{zz} = 0$	$\tau_{x\theta} = \frac{T \times \rho}{J}$; $\tau_{yz} = 0$
Bending (about Z Axis)	$\sigma_{xx} = -\frac{M_z * y}{I_{zz}}$; $\sigma_{yy} = 0$; $\sigma_{zz} = 0$	$\tau_{xs} = -\frac{V_y \times Q_z}{I_{zz} \times t}$; $\tau_{yz} = 0$

σ_{xx} is the normal stress that is associated with the axial load and is equal to the axial load divided to its area. Since in the real case too, there is a possibility of axial load only along the x axis and hence σ_{zz} *and* σ_{yy} are zero. Also, it is to be noted that because of the axial load acting along the neutral axis, there are no shear stresses.

Due to Torsion, there are no normal stresses. The only shear stress component is $\tau_{x\theta}$, where T is the Twisting Moment and ρ is the outer radius and J is the Polar Moment of Inertia (Young, 1989).

Due to Bending along Z axis, the only normal stresses is σ_{xx} *as* $\sigma_{yy} = 0 = \sigma_{zz}$. M_z is the bending moment while y is the vertical distance from the neutral axis. I_{zz} is the second moment of inertia of the entire shaft. The corresponding shear stress caused by bending is τ_{xs}. The vertical shear force (in most cases being the bending load) is represented by V_y and first moment of inertia being represented by Q_z while t represents the thickness which is equal to difference between the diameters (Young, 1989).

2.6 Stresses and strains

The extreme values of normal stresses on shear-free planes are called the Principal Stresses and the planes on which the principal stresses act are called the principal planes. In two-dimensional cases, there are two principal stresses, namely the major principal stress and the minor principal stress which are defined as the maximum and minimum values of the normal stresses respectively (Khurmi, 2008).

$$\sigma_{x'} = \left(\frac{\sigma_x+\sigma_y}{2}\right) + \left(\frac{\sigma_x-\sigma_y}{2}\right)\cos 2\theta + \tau_{xy}\sin 2\theta \quad (6)$$

$$\sigma_{y'} = \left(\frac{\sigma_x+\sigma_y}{2}\right) - \left(\frac{\sigma_x-\sigma_y}{2}\right)\cos 2\theta - \tau_{xy}\sin 2\theta \quad (7)$$

$$\tau_{xy'} = -\left(\frac{\sigma_x-\sigma_y}{2}\right)\sin 2\theta + \tau_{xy}\cos 2\theta \quad (8)$$

$$\sigma_{1,2} = \frac{\sigma_x+\sigma_y}{2} \pm \sqrt{\left(\frac{\sigma_x-\sigma_y}{2}\right)^2 + \tau_{xy}{}^2} \quad (9)$$

$$\tau_{max,min} = \pm\sqrt{\left(\frac{\sigma_x-\sigma_y}{2}\right)^2 + \tau_{xy}{}^2} \quad (10)$$

$$\theta_p = \frac{1}{2}\tan^{-1}\left(\frac{2\tau_{xy}}{\sigma_x-\sigma_y}\right) \quad (11)$$

In case of a shaft subjected to twisting, bending and axial load, the above equations can be alternatively expressed as follows (Young, 1989)

$$\sigma_{1,2} = \frac{\sigma}{2} \pm \frac{1}{2}\sqrt{\sigma^2 + 4\tau^2} \quad (12)$$

$$\tau_{max,min} = \pm\frac{1}{2}\sqrt{\sigma^2 + 4\tau^2} \quad (13)$$

In the equations (12) & (13), σ represents the normal stresses due to all the loads combined while τ represents the shear stresses due to all the loads combined.

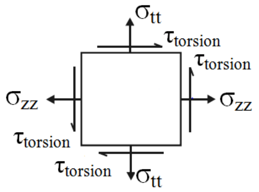


Figure 7 Stress plane

Figure 7 shows the stress directions that act along different directions for a considered region. The stress σ_{zz} refers to stress acting on the z plane along z direction. σ_{tt} refers to stress acting on phi plane along phi values. Theses stresses are extended to get strain (using Young's modulus).
If the value of ε_{zz}, ε_{tt} and $\gamma_{torsion}$, are calculated, the next step is to calculate the strain at 45 degrees and also at other angles. Using strain transformation, strain at any angle can be found out using the formula,

$$\varepsilon_{angle} = \left(\frac{\varepsilon_{zz}+\varepsilon_{tt}}{2}\right) + \left(\frac{\varepsilon_{zz}-\varepsilon_{tt}}{2}\right)\cos 2\theta + (\gamma_{xy}/2)\sin 2\theta \quad (14)$$

Since the principal strain would not be at 45 degrees (because of the presence of bending load), there is no use of calculating the maximum and minimum principal strains. Hence using the above expression, strain at any angle can be computed.

By this method, engineering strain along any direction can be calculated provided the three values are found out.

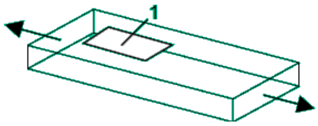
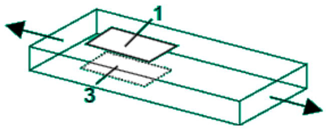
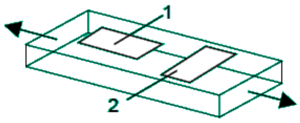
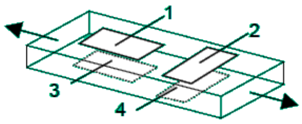
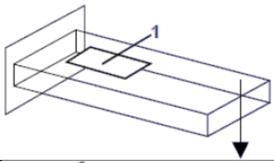
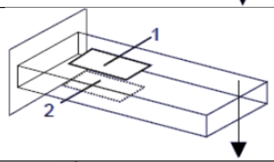
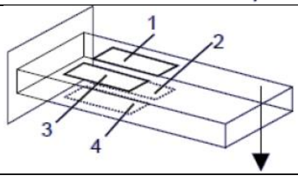
2.7 Article by T.A. Wilson

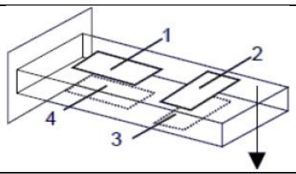
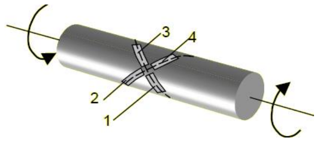
T.A.Wilson in 1954 published an article titled *"The Eccentric Circular Tube"* in the book Aircraft Engineering by 1942. In the article, he described the mathematical method of computing the shear stress at its maximum position along an eccentric shaft. This article forms the backbone of this thesis. Wilson calculated the shear stress only for the position corresponding to maximum shear stress. His expression is further extrapolated in order to find the maximum shear stress at all the locations of the shaft. Details regarding this are explained in Chapter 3.4.3.

2.8 Positioning of strain gauges

Positioning of Strain Gauges is necessary for optimal measurements of the load associated with it. Table 4 summarizes the various configurations of different loads and lists out the best possible configuration for each type. As a general rule of thumb, in a Wheatstone bridge, equal changes (e.g. temperature) in adjacent arms will cancel out each other (Hoffmann, 1989).

Table 4 Different configurations of positioning strain gauges (Hoffmann, 1989)

TYPE	CONFIGURATION	NOTES
Axial Strain		Must use dummy gauge in adjacent arm (2 or 4) to achieve temperature configuration
		Rejects bending strain but no temperature compensation. Must add dummy gauges in arms 2&4 to compensate for temperature
		Temperature compensated, but sensitive to bending strain
		Best configuration for axial loads as it compensates for temperature and rejects bending strain.
Bending Strain		Also responds equally to axial strains, must use dummy gauge in an adjacent arm (2 or 4) to achieve temperature compensation
		Half bridge configuration. Rejects axial strain and is temperature compensated. Dummy resistors in arms 3&4 can be in strain indicator
		Maximum sensitivity to bending rejects axial strains and temperature compensated. Best Configuration for bending loads

		Adequate configuration but not as good as the previous configuration. Compensates for bending and rejects axial strain.
Torsional Strain		Half bridge configuration. Gages at 45 degrees to center line sense principal strains which are equal and opposite for pure torsion, bending or axial force induces equal strains and is rejected and arms are temperature compensated
		Full bridge configuration and the best configuration for torsional strains. Rejects axial and bending strain and temperature compensated.

2.9 Torque Measurement

It is to be noted that $\sigma_{1,2} = \tau$ in case of pure torsion as there are no normal stresses associated with pure torsion. The principal stresses (maximum and minimum normal stresses) occur at an angle 45° to the cylindrical planes (lines running parallel to the longitudinal axis of the shaft). As a result, placement of the strain gauges along 45 degrees will be an indication of the principal strain corresponding to the principal stresses. Using that shear stress τ can be computed form which the torque can be found out. (Hoffmann, 1989)

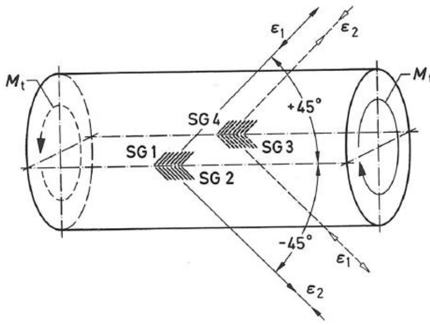


Figure 8 Torsion shaft with X-rosette gauges mounted in the principal strain directions

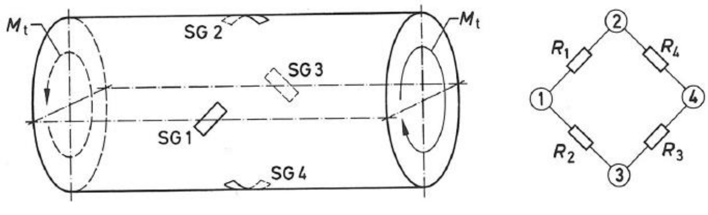


Figure 9 Torque measurement and Wheatstone bridge circuit

$$\sigma_1 = \frac{E(\varepsilon_1 + v\varepsilon_2)}{1 - v^2} \tag{15}$$

$$\sigma_2 = \frac{E(\varepsilon_2 + v\varepsilon_1)}{1 - v^2} \tag{16}$$

ε_1 & ε_2 are the measured strain from the strain gauges and v being the Poisson's Ratio. E is the Young's modulus of the shaft material. (Hoffmann, 1989)

When the shaft is twisted, $\varepsilon_1 = -\varepsilon_2$

For a half bridge connection,

$$\sigma_{1,2} = \pm \frac{E\varepsilon_i(1-v)}{2(1-v^2)} \quad (17)$$

$$\tau_{max} = \varepsilon_i \times G \quad (18)$$

Similarly, for a full bridge connection,

$$\sigma_{1,2} = \pm \frac{E\varepsilon_i(1-v)}{4(1-v^2)} \quad (19)$$

$$\tau_{max} = \frac{\varepsilon_i}{2} \times G \quad (20)$$

Once τ_{max} is calculated, the Torque can be computed using the formula

$$T = \tau_{max} \times S_p \quad (21)$$

S_p is the Polar Section Modulus, which can also be termed as the ratio between Polar Moment of Inertia and the Outer Diameter.

2.10 Poisson's ratio and Gauge Factor

When a sample object is stretched (or squeezed), to an extension (or contraction) in the direction of the applied load, it corresponds to a contraction (or extension) in a direction perpendicular to the applied load. The ratio between these two quantities is the Poisson's ratio (Muftah, 2011).

$$\nu = -\frac{\varepsilon_t}{\varepsilon_a} \quad (22)$$

Strain gauge is a device used to measure the strain of an object. As the object is deformed, the foil is deformed, causing its electrical resistance to change. This resistance change, usually measured using a Wheatstone bridge, is related to the strain by the quantity known as the gauge factor (Muftah, 2011).

$$\text{Gauge Factor } \lambda = \frac{\frac{\Delta R}{R}}{\frac{\Delta L}{L}} \quad (23)$$

If the Wheatstone bridge connection is present, then

$$\frac{V_o}{V_{EX}} = \tau \times \lambda \times \left(\frac{1+v}{E}\right) \quad (24)$$

2.11 Error due to misalignment

When a gage is bonded to a test surface at a small angular error with respect to the intended axis of strain measurement, the indicated strain will also be in error due to the gage misalignment. Magnitude of misalignment error depends upon three factors. (Vishay Precision Group, 2010)

- The ratio of the algebraic maximum to the algebraic minimum principal strain, $\varepsilon p / \varepsilon q$.
- The angle φ between the maximum principal strain axis & intended strain measurement axis
- The angular mounting error, β, between the gage axis & intended axis of strain measurement.

The error in measurement caused by angular misalignment n is given by,

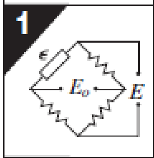
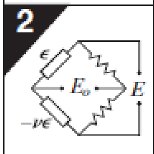
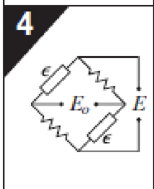
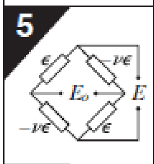
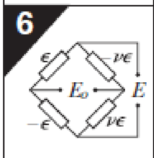
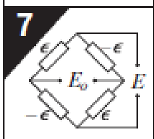
$$n = \varepsilon_{(\varphi \pm \beta)} - \varepsilon_{\varphi} \quad (25)$$

$$n = \frac{\varepsilon_1 - \varepsilon_2}{2}\left[\cos 2(\varphi \pm \beta) - \cos 2\varphi\right] \quad (26)$$

2.12 Corrections for Wheatstone bridge nonlinearity

The output voltage obtained from the unbalanced Wheatstone bridge is a function of the amount of unbalance, and is therefore directly related to the strain applied to the strain gage. However, under certain conditions frequently encountered in actual practice, the bridge output voltage is, as noted earlier, a nonlinear function of the resistance change in the bridge arms. When this occurs, the strain readings will be somewhat in error (Vishay Precision Group, 2010). Table 5 shows the corrections associated with different configurations.

Table 5 Wheatstone bridge nonlinearity corrections (Vishay Precision Group, 2010)

Bridge/Strain Arrangement (Note 1)	**Description**	**Bridge Output, E_o/E mV/V** (Notes 2, 3)	**Nonlinearity, η Where $E_o/E = K\epsilon \times 10^{-3}(1-\eta)$** (Notes 2, 3)	**Corrections** (Note 3, 4)
1 	Single active gage in uniaxial tension or compression.	$\frac{E_o}{E} = \frac{F\epsilon \times 10^{-3}}{4 + 2F\epsilon \times 10^{-6}}$	$K = \frac{F}{4}$ $\eta = \frac{F\epsilon \times 10^{-6}}{2 + F\epsilon \times 10^{-6}}$	$\epsilon = \frac{2\epsilon_i}{2 - F\epsilon_i \times 10^{-6}}$
2 	Two active gages in uniaxial stress field — one aligned with maximum principal strain, one "Poisson" gage.	$\frac{E_o}{E} = \frac{F\epsilon(1+\nu) \times 10^{-3}}{4 + 2F\epsilon(1-\nu) \times 10^{-6}}$	$K = \frac{F(1+\nu)}{4}$ $\eta = \frac{F\epsilon(1-\nu) \times 10^{-6}}{2 + F\epsilon(1-\nu) \times 10^{-6}}$	$\epsilon = \frac{2\epsilon_i}{2(1+\nu) - F\epsilon_i(1-\nu) \times 10^{-6}}$
3 	Two active gages with equal and opposite strains — typical of bending-beam arrangement.	$\frac{E_o}{E} = \frac{F\epsilon}{2} \times 10^{-3}$	$K = \frac{F}{2}$; $\eta = 0$	$\epsilon = \frac{\epsilon_i}{2}$
4 	Two active gages with equal strains of same sign — used on opposite sides of column with low temperature gradient (bending cancellation, for instance).	$\frac{E_o}{E} = \frac{F\epsilon \times 10^{-3}}{2 + F\epsilon \times 10^{-6}}$	$K = \frac{F}{2}$ $\eta = \frac{F\epsilon \times 10^{-6}}{2 + F\epsilon \times 10^{-6}}$	$\epsilon = \frac{2\epsilon_i}{4 - F\epsilon_i \times 10^{-6}}$
5 	Four active gages in unaxial stress field two aligned with maximum principal strain, two "Poisson" gages (column).	$\frac{E_o}{E} = \frac{F\epsilon(1+\nu) \times 10^{-3}}{2 + F\epsilon(1-\nu) \times 10^{-6}}$	$K = \frac{F(1+\nu)}{2}$ $\eta = \frac{F\epsilon(1-\nu) \times 10^{-6}}{2 + F\epsilon(1-\nu) \times 10^{-6}}$	$\epsilon = \frac{2\epsilon_i}{4(1+\nu) - F\epsilon_i(1-\nu) \times 10^{-6}}$
6 	Four active gages in uniaxial stress field — two aligned with maximum principal strain, two "Poisson" gages (beam).	$\frac{E_o}{E} = \frac{F\epsilon(1+\nu) \times 10^{-3}}{2}$	$K = \frac{F(1+\nu)}{2}$; $\eta = 0$	$\epsilon = \frac{\epsilon_i}{2(1+\nu)}$
7 	Four active gages with pairs subjected to equal and opposite strains (beam in bending or shaft in torsion).	$\frac{E_o}{E} = F\epsilon \times 10^{-3}$	$K = F$; $\eta = 0$	$\epsilon = \frac{\epsilon_i}{4}$

2.13 Transverse sensitivity

Transverse sensitivity in a strain gage refers to the behavior of the gage in responding to strains which are perpendicular to the primary sensing axis of the gage. Ideally, it would be preferable if strain gages were completely insensitive to transverse strains. In practice, most gages exhibit some degree of transverse sensitivity; but the effect is ordinarily quite small, and of the order of several percent of the axial sensitivity. (Vishay Precision Group, 2011)

In general, a strain gage actually has two gage factors, F_a and F_t, which refer to the gage factors as determined in a uniaxial strain field (not uniaxial stress) with, respectively, the gage axes aligned parallel to and perpendicular to the strain field. For any strain field, the output of the strain gage can be expressed as follows: (Vishay Precision Group, 2011)

$$\frac{\Delta R}{R} = \varepsilon_a F_a + \varepsilon_t F_t \tag{27}$$

$\varepsilon_a, \varepsilon_t$ = strains parallel to and perpendicular to the gage axis, or the gridlines in the gage.
F_a = axial gage factor.
F_t = transverse gage factor.

Poisson's Ratio $v_o = -\frac{\varepsilon_t}{\varepsilon_a}$ and Transverse Sensitivity coefficient $K_t = \frac{F_t}{F_a}$

Hence,

$$\frac{\Delta R}{R} = F_a \varepsilon_a (1 - v_o K_t) = F \varepsilon_a \tag{28}$$

where F = $F_a(1 - v_o K_t)$ which the manufacturer's gauge factor (Vishay Precision Group, 2011).

2.14 DOE

In statistics, fractional factorial designs are experimental designs which consist of a fraction of the full factorial design. The major inference that can be carried out from a factorial design is that it will give a clear view of the effects of the parameters on the experiment or the simulation (Box, 2005). Either fractional factorial design or full factorial design can be performed, depending on the number of runs. Fractional Factorial design also gives the main effects and the interaction effects of these parameters. Thereby the effects can be quantified and numerically compared. Two levels can be inflicted on these cases with – representing a low level and + representing a higher level. Table 66 shows a factorial design with three levels.

Table 6 Two level Factorial Design

Cases	A	B	C	D = ABC
1	-	-	-	-
2	+	-	-	+
3	-	+	-	+
4	+	+	-	-
5	-	-	+	+
6	+	-	+	-
7	-	+	+	-
8	+	+	+	+

A, B, C and D are the factors, while - and + refers to the lower level and the higher level respectively.

In this chapter the working process is described.

3.1 Different imperfections

As explained earlier, the main aim of this thesis is to calculate the sensitivity caused by different imperfections (if any) that occur commonly in the transducers. The following were listed out to be the major imperfections contributing to the sensitivity measurement system.

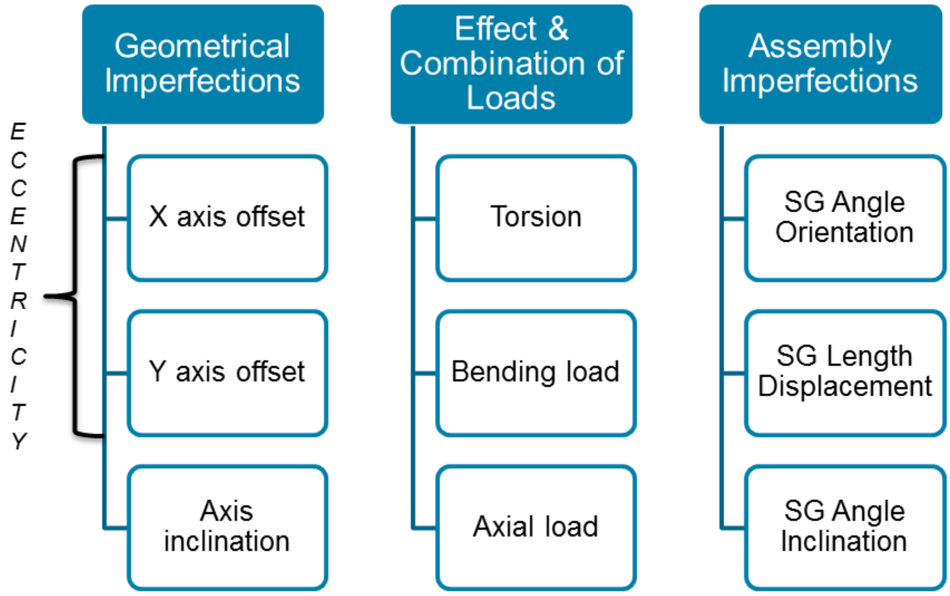


Figure 10 Different imperfections

3.1.1 Eccentricity

It is highly difficult to achieve zero concentricity between the outer diameter and the inner hole. Eccentricity refers to the effective distance between the centers of the two circles (the outer diameter and the hole). The X axis offset and the Y axis offset refers to the offsets in the x axis and y axis respectively. It gives the position of the inner hole at a particular instant. Figure 11 shows an exaggerated eccentric hollow shaft.

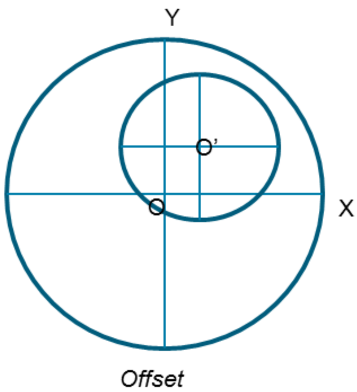


Figure 11 Eccentricity imperfection

3.1.2 Axis inclination

Axis inclination refers to the angle with which the inner axis is inclined to the outer axis. The inner axis corresponds to the hole and the outer axis corresponds to the shaft. This imperfection is closely related with run-out tolerance. Figure 12 shows an exaggerated shaft (for illustration purpose) with an imperfection occurring because of axis inclination.

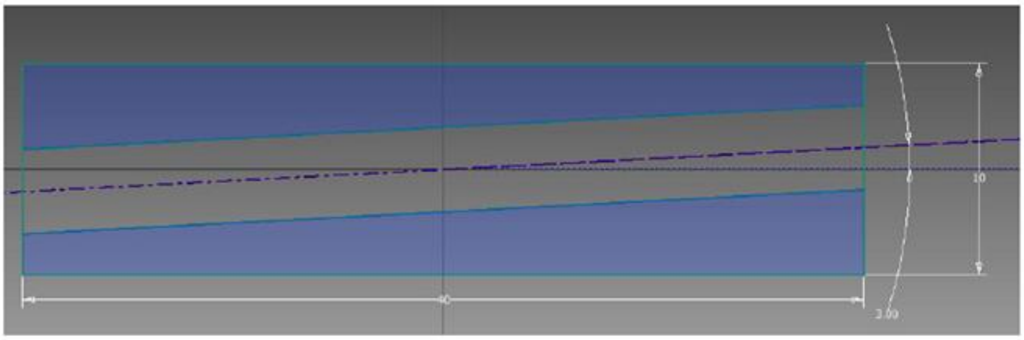


Figure 12 Axis inclination imperfection

3.1.3 Effect and combination of loads

Usually, only the torsion load is desired on the transducer body. For a concentric shaft, if there is a bending load, it gets compensated in the Wheatstone bridge arrangement. Thereby, it doesn't have any effect on the sensitivity measurement. However, it is unclear if there will be compensation for a geometrically imperfect shaft. Nevertheless, if there are other loads present, the indicated strain at any point will be different.

3.1.4 Strain gauge angle orientation

This is one of the commonly occurring assembly errors. Strain gauges are intended to be kept at 45 degrees to measure strain and subsequently compute torque, since the maximum principal strain occurs at these angles. If they are inclined at an angle other than 45, then that case refers to this imperfection. Figure 13 shows a typical imperfection.

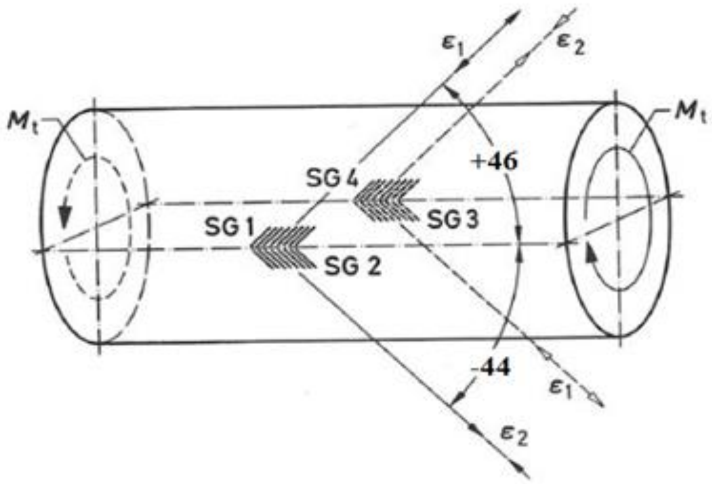


Figure 13 Strain gauge angle orientation imperfection

3.1.5 Strain gauge length displacement

The strain gauges are intended to be kept at 4 locations usually, symmetrically along the same length. If one of them is not properly aligned with respect to the length along the lateral sides of the shaft, then that case is considered to be this error. If there is just torque on the shaft, then this imperfection doesn’t have any effect on the output sensitivity, since the principal strain will be the same at all the points oriented at 45 degrees. However, if there are other loads, then there will be a change in sensitivity.

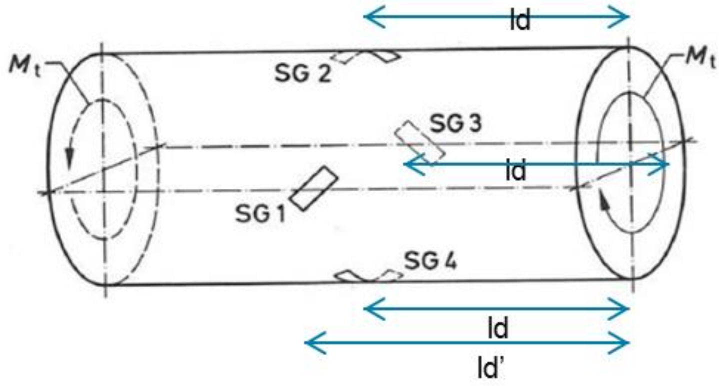


Figure 14 Strain gauge length displacement imperfection

3.1.6 Strain gauge angle inclination

Normally, 4 strain gauges are employed around the shaft at a common distance from the load end of the shaft. They are placed at an angle of 90 degrees to each other. If the angle between two successive strain gauges (in a 4 strain gauge system) is not 90 degrees, then that error is called strain gauge angle inclination imperfection.

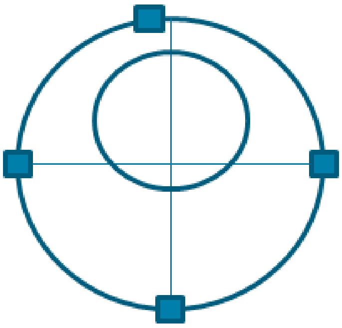


Figure 15 Strain gauge angle inlcination imperfection

3.2 Input parameters

There are many input parameters that can influence the measurement of strain and the torque measurement in a shaft. The major parameters that define a shaft are as follows:

- Outer diameter of the shaft
  This corresponds to the outer diameter of the shaft
- Inner diameter of the shaft
  This corresponds to the inner diameter of the shaft. For a hollow shaft, this value is zero.
- Length of the shaft
  This corresponds to the total length of the solid shaft.

- Twisting moment acting on the shaft
  Twisting moment corresponds to the turning moment that is acting on the shaft because of a Torque. For convenience, if the twisting is along the clockwise direction, a negative sign is intended to be specified preceding the input value of torque.
- Bending moment acting on the shaft
  Bending moment corresponds to the moment that is acting on one end of the shaft because of a Vertical load. The vertical load tends to deform the shaft.
- Axial load acting on the shaft
  Axial load corresponds to either the tensile or the compressive load that acts on the end of the shaft. In this thesis, it is assumed that the axial load is acting along the neutral axis and is perpendicular to the bending load.
- Young's modulus of the shaft
  It is a measure of the stiffness of an elastic material. According to Hooke's law, it is the ratio of the stress along an axis over the strain along that axis in the range of stress. Different materials have different Young's modulus. By changing the Young's modulus, the user can simulate the same for a particular material, provided the Young's modulus of the material is known.
- Poisson's ratio of the shaft
  Poisson's ratio is defined as the negative ratio of the transverse strain to axial strain. When a material is compressed in a direction, it usually tends to expand in the other two directions which are perpendicular to the direction of compression. Poisson's ratio is a measure of this effect, which is called Poisson's effect. Since the Poisson's ratio is different for different materials, it is viable to include Poisson's ratio as an input parameter.
- Offset in x-direction
  It is the offset of the inner diameter axis from the outer diameter axis in x axis when measured from the load end. This parameter, along with the offset in y-direction is usually the main contributor for the geometrical tolerance errors in manufacturing industry corresponding to hollow shafts.
- Offset in y-direction
  It is the offset of the inner diameter axis from the outer diameter axis in y axis when measured from the load end.
- Axis inclination
  This is inclination of the inner diameter axis from the outer diameter axis. It is closely related to run-out tolerance.

3.3 Matlab script brief

The Matlab Script is written with compatibility with Microsoft Excel as far as the input parameters are concerned. The Matlab script is split into eight sections (Appendix 2).

- The first section deals with the input parameters. An excel sheet is used in tandem with the Matlab code and the various parameters are entered. Apart from that, the coordinates of the strain gauge location are entered in different sheets.
- The next section deals with torsion computation. Once the input value is read by Matlab, the torsional shear stress is calculated along different angles. The user has an option to mention the grid size of the angles and the length.
- The third section deals with finding out the stress and strain correspondingly for bending and axial loads along different lengths and angles. For that matter, a double for loop is used
  - Section 3.1
    - It finds out the centroid, neutral axis and the interested point of consideration (for example, the coordinates of 360 points when the user is interested in 360 points)

- Section 3.2
    - It finds out with respect to neutral axis – the area above the neutral axis, xbar and ybar (centroids) of the inner circle
- Section 3.3
    - It finds out with respect to neutral axis – the area above the neutral axis, xbar and ybar (centroids) of the outer circle
- Section 3.4
    - Now that the outer and inner circles' geometrical properties are obtained, this section calculates the first moment of area with respect to the neutral axis
- Section 3.5
    - This section calculates the polar moment of inertia with respect to neutral axis
- Section 3.6
    - This section calculates second moment of inertia with respect to neutral axis
- Section 3.7
    - The basic idea is that each point has to be considered separately and thereby the geometrical properties with respect to that point would vary accordingly. If the point considered is in the top half (technically first or second quadrant), and if the point position with respect to the inner circle (vertical distance) is analyzed and the properties are calculated accordingly.
- Section 3.8
    - Similarly, if the inner circle is in third or fourth quadrant, then the different possible locations of the point on the outer diameter is analyzed and the suitable geometrical parameters are calculated on each case
- Section 3.9
    - Here the effect of the inner circle is negated temporarily and the geometrical parameters are calculated accordingly.
- Section 3.10
    - Here, the difference between the above two section is further extrapolated to calculate the First moment of area, polar moment of inertia, second moment of inertia and the vertical distance from the bending load.

- Fourth section
  Here, the bending parameters are listed out and the thickness at the point of interest is calculated, which is needed for evaluating the bending stress.
- Fifth section
  The total bending stresses and the torsional stresses are super positioned to calculate the Maximum Shear stresses.
- Sixth section
  Here the different interesting plots are plotted
- Seventh section
  So far, strain at individual points is calculated. Now, they are clubbed together to calculate the average strain at a region. All the values of the strain at different points are indexed in an array. Using the input coordinates of the region and the program identifies and uses the necessary values calculated in section 3. Thus, the average strain in a region is calculated
- Eight section
  This section computes the sensitivity of the strain gauge using the strain values at a region obtained from the previous step.

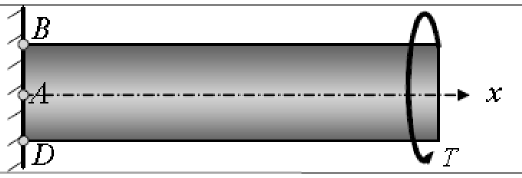
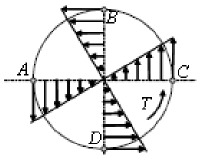
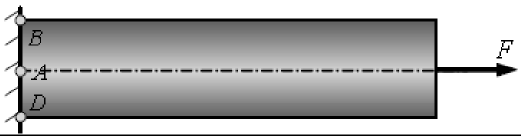
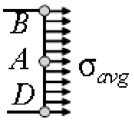
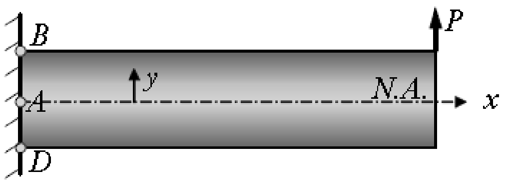
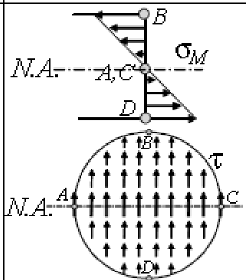
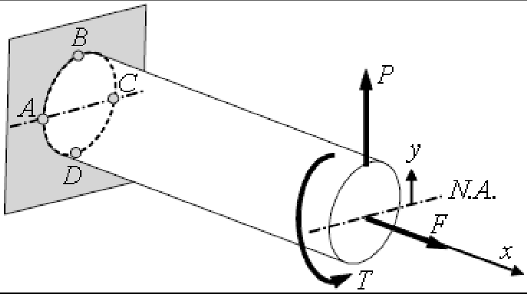
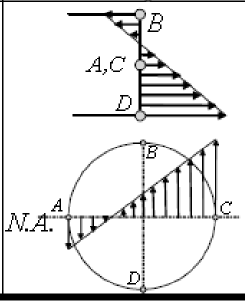
3.4 Different stresses

The following table shows the normal and shear stresses that are associated with different loading cases. By calculating each expression, the corresponding strain that shall be sensed by the strain gauge can be obtained.

Table 7 Normal and Shear stresses for different loads

Load	Normal Stress	Shear Stress
Axial	$\sigma_{axial} = \frac{N}{A}$	$\tau_{axial} = 0$
Torsion	$\sigma_{torsion} = 0$	$\tau_{x\theta} = \frac{T \times \rho}{J}$
Bending (about Z Axis)	$\sigma_{bending} = -\frac{M_z * y}{I_{zz}}$ $(perendicular\ to\ the\ bending\ load)$	$\tau_{xs} = -\frac{V_y \times Q_z}{I_{zz} \times t}$ $(opposite\ in\ direction\ to\ the\ bending\ load)$

Table 8 Direction of stresses

	Stresses Produced by Each Load Individually	Stress Distributions	Stresses
Torsional Load (Torque *T*)			Torsional shear stress $\tau_T = T\rho/J$
Axial Load (Force *F*)			Tensile average normal stress $\sigma_{avg} = F/A$
Bending Load (Transverse Force *P*)			Bending normal stress $\sigma_M = -My/I$ Transverse shear stress $\tau_V = VQ/It$
Combined Loads			Total normal stress $\sigma = F/A - My/I$ Total shear stress at *N.A.* $\tau = VQ/It \pm T\rho/J$

3.5 Calculation of parameters related to torsional moment

Whenever a shaft is subjected to torsional moment, there are no normal stresses. Only the shearing stresses occur due to Torsion. This shear stress will result in strain along 45 degree and -45 degree, which shall be sensed by the strain gauge. So, basically, it is adequate to calculate the shearing stresses caused by torsion.

To calculate the strain related to Twisting Moment, the calculation corresponding to a solid shaft and a concentric shaft are shown first. Then the complexities of the eccentric offsets are computed using St. Venant's torsion principle is illustrated.

3.5.1 Ideal solid shaft

This is the most common type of shaft that is used in many applications. When a shaft is subjected to a twisting load, deformation occurs because of shear stress. The twisting moment along the ends of the shaft, tends to cause shear along the surfaces of the shaft. For a solid shaft, the shear stress that is caused by the twisting moment is same along all the points/surfaces of the shaft. Since this thesis is mainly concerned with finding the strain that can be sensed by the strain gauge, the only formula that is interesting is that of the shear stress. Shear stress τ can be found out provided the Twisting Moment (T) and the diameter (D) are known (Vable, 2009). From that, shear strain can be found just by dividing it with Shear modulus G. Shear modulus is equal to Young's modulus, divided by a factor $2(1+\nu)$.

$$\tau_{torsion} = \frac{16\,T}{\pi D^3} \tag{29}$$

3.5.2 Ideal concentric shaft

Concentric shafts are those that have a circular hole along the center of the shaft. It can be typically related using the two parameters – outer and inner diameters. When the centers of the two diameters lie on the same point, it can be termed as a concentric shaft. It is to be noted that the term 'hollow shaft' generically means an ideal concentric shaft unless specified. The typical practical problem occurring with the concentric shaft is that it is difficult to precisely manufacture a concentric shaft. Although a shaft with concentricity 6 microns can be considered as very accurate, the small variation can tend to bring a change in the shear stress distribution along the shaft. Almost all of the calculations for shaft are either pervasively based on solid or concentric shafts. Shear stress τ can be calculated if the Twisting moment (t), Outer diameter (D) and inner diameter (d) are known (Vable, 2009).

$$\tau_{torsion} = \frac{16\,T}{\pi\left(\frac{D^4-d^4}{D}\right)} \tag{30}$$

3.5.3 Eccentric hollow shaft

Eccentric hollow shafts are typically observed in hollow shafts. Almost all of the hollow shafts that are manufactured in industries are eccentric shafts pertaining to the fact that it is highly unlikely to achieve zero concentricity. The effect of concentricity can be extrapolated from the work of T.S.Wilson, 1954.

The general St. Venant torsion problem may be reduced to the determination of a function φ, which satisfies the equation throughout the material section of the tube and which also satisfies the boundary condition $\psi = 0$ on the outer boundary and $\psi = K$ on the inner boundary

$$\nabla^2\psi + 2 = 0 \tag{31}$$

K is a constant and can be determined from the fundamental principle that the displacement of any point of the section must be a single valued function (Wilson, 1954). If the function ψ can be determined, then the shear stress at any point of the section can be found. The direction of the stress is tangential to that curve of the family ψ = constant which passes through the point and its magnitude is $E\phi\psi\frac{\delta\psi}{\delta n}$, where E is the coefficient of rigidity, ϕ is the twist per unit length of the shaft and $\frac{\delta\psi}{\delta n}$ represents the differentiation along the normal to that curve of the family ψ = constant, which passes through the point.

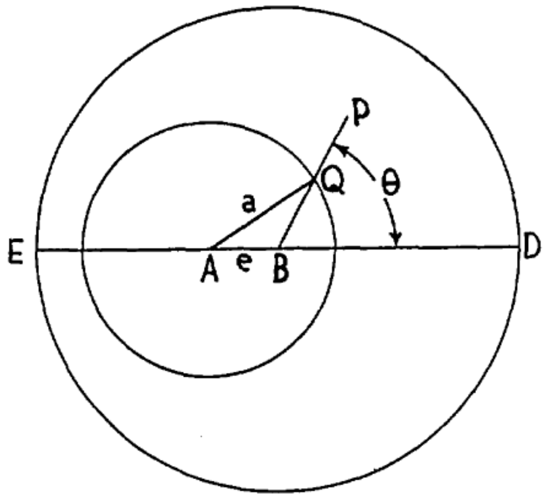


Figure 16 An eccentric circular shaft (Wilson, 1954)

From above figure, B is the center of the outer diameter and A is the centre of the inner diameter. Eccentricity is represented by e which is offset between the inner and the outer diameter. The outer diameter is represented by b and the inner diameter is represented by a.

Using polar coordinates (r, θ) equation 28 can be written as

$$\frac{1}{r}\frac{\delta}{\delta r}\left(r\frac{\delta\psi}{\delta r}\right)+\frac{1}{r^2}\frac{\delta^2\psi}{\delta\theta^2}+2=0 \tag{32}$$

The components of shear stress τ at any point (r, θ) will be as follows

$$\tau(radial) = \frac{E\phi}{r}\frac{\delta\psi}{\delta\theta} \tag{33}$$

$$\tau(tangential) = -E\phi\frac{\delta\psi}{\delta r} \tag{34}$$

- Any point on the outer boundary is represented by boundary condition being ψ = 0 and the equation being ψ = 0
- Any point on the inner boundary is represented by boundary condition ψ = K (a constant) and by equation (32)

$$r = -e\cos\theta + a\sqrt{1-\frac{e^2sin^2\theta}{a^2}} \tag{35}$$

$$\frac{\rho}{a} = -\rho\cos\theta + a\sqrt{1-\rho^2sin^2\theta} \tag{36}$$

Where $\rho = \frac{e}{a}$ and θ is the angle that the point P on the outer surface makes with the point D (Angle PBD).

In accordance with this method, a function ψ must be sought out such that it shall satisfy the equation (31) and also the two boundary conditions which are given below.

$$\psi = -\frac{r^2}{2} + a_0 + \sum_{n=1}^{\infty}\left(a_n r^n + \frac{\beta_n}{b^n}\right)\cos n\theta \tag{37}$$

Equation 9 is a solution of equation 4 where a's and β's are constants (Wilson, 1954)
Since, the main interest is to calculate the stress along the outer diameter, it can be hypothesized that the boundary conditions also satisfy when $\psi = 0$ for all values of θ.
Thus, for any point on the outer diameter, the boundary condition will be satisfied if

$$0 = -\frac{b^2}{2} + a_0 + \sum_{n=1}^{\infty}\left(a_n r^n + \frac{\beta_n}{b^n}\right)\cos n\theta \tag{38}$$

In order for the above equation to be zero, a_0 *should be equal to* $\frac{b^2}{2}$ and
β_n *should be equal to* $-a_n b^{2n}$, for the different values of n.

Wilson demonstrated that by using the expression of a_1 to a_4 as shown below, the above equation can be rewritten for torsion computation

$$a_1 = a_{11}\rho + a_{12}\rho^2 + a_{13}\rho^3 + a_{14}\rho^4 + \ldots \tag{39}$$

$$a_2 = a_{22}\rho^2 + a_{23}\rho^3 + a_{24}\rho^4 + \cdots \tag{40}$$

$$a_3 = a_{33}\rho^3 + a_{34}\rho^4 + \cdots \tag{41}$$

$$a_4 = a_{44}\rho^4 + \cdots \tag{42}$$

Similarly, on the inner boundary,

$$K = -\frac{b^2}{2} + a_0 + \sum_{n=1}^{\infty}\left(a_n r^n + \frac{\beta_n}{b^n}\right)\cos n\theta \tag{43}$$

$$0 = -K - \frac{r_i^2}{2} + a_0 + \sum_{n=1}^{\infty}\left(a_n r_i^n + \frac{\beta_n}{r_i^n}\right)\cos n\theta \tag{44}$$

Wilson found that r_i^n and $\frac{1}{r_i^n}$ could be expanded in Cosine series, leading to the expression

$$0 \equiv A_0 + \sum_{n=1}^{4} A_n \cos n\theta \tag{45}$$

Provided $A_0 \equiv A_1 \equiv A_2 \equiv A_3 \equiv A_4 \equiv 0$
Based on this, Wilson yielded a set of equations

$$a_0 - K - \frac{a^2}{2} - \frac{1}{2}a_1 a\rho + \frac{1}{2}a_2 a^2\rho^2 + \frac{1}{2}\frac{\beta_1}{a}(\rho + \rho^3) + \frac{1}{2}\frac{\beta_2}{a^2}(\rho^2) = 0 \tag{46}$$

$$a^2\left(\rho - \frac{\rho^3}{8}\right) + a_1 a\left(\rho - \frac{\rho^2}{8}\right) - a_2 a^2\rho + \frac{\beta_1}{a}\left(1 + \frac{7}{8}\rho^2\right) + \frac{\beta_2}{a^2}\rho = 0 \tag{47}$$

$$-\frac{1}{2}a^2\rho^2 - \frac{1}{2}a_1 ap + a_2 a^2 - \frac{3}{4}a_3 a^3\rho + \frac{1}{2}\frac{\beta_1}{a}(\rho + \rho^3) + \frac{\beta_2}{a^2}(1 + 2\rho^2) + \frac{3}{2}\frac{\beta_3}{a^3}\rho = 0 \tag{48}$$

$$\frac{1}{8}a^2\rho^3 + \frac{1}{8}a_1 a\rho^2 - a_2 a^2\rho + a_3 a^3 + \frac{1}{8}\frac{\beta_1}{a}\rho^2 + \frac{\beta_2}{a^2}\rho + \frac{\beta_3}{a^3} = 0 \tag{49}$$

$$\frac{1}{2}a_2 a^2\rho^2 - \frac{3}{2}a^3 a_3\rho + a_4 a^4 + \frac{1}{2}\frac{\beta_2}{a^2}\rho^2 + \frac{3}{2}\frac{\beta_3}{a^3}\rho + \frac{\beta_4}{a^4} = 0 \tag{50}$$

Using the boundary conditions and the above set of equations, the expression can be further reduced in order to determine an expression for a_{11} to a_{44}

$$a^2\left(\rho-\frac{\rho^3}{8}\right)+a_1\left[\left(a-\frac{b^2}{a}\right)-\frac{1}{8}\rho^2\left(a+\frac{7b^2}{a}\right)\right]-a_2\rho\left(a^2+\frac{b^4}{a^2}\right)=0 \quad (51)$$

$$\frac{1}{2}a^2\rho^2-a_1\left[\frac{1}{2}\rho\left(a+\frac{b^2}{a}\right)+\frac{1}{2}\rho^3\frac{b^2}{a}\right]+a_2\left[\left(a^2-\frac{b^4}{a^2}\right)-2\rho^2\frac{b^4}{a^2}\right]-\frac{3}{2}a_3\rho\left(a^3+\frac{b^6}{a^3}\right)=0 \quad (52)$$

$$\frac{1}{8}a^2\rho^3+\frac{1}{8}a_1\rho^2\left(a-\frac{b^2}{a}\right)-a_2\rho\left(a^2+\frac{b^4}{a^2}\right)+a_3\left(a^3-\frac{b^6}{a^3}\right)=0 \quad (53)$$

$$\frac{1}{2}a_2\rho^2\left(a^2-\frac{b^4}{a^2}\right)-\frac{3}{2}a^3\rho\left(a^3+\frac{b^6}{a^3}\right)+a_4\left(a^4-\frac{b^8}{a^4}\right)=0 \quad (54)$$

The above equations are simplified and can be rewritten using the values of a_1, a_2, a_3 and a_4

$$a_{11}=\frac{a^3}{b^2-a^2} \quad (55)$$

$$a_{12}=0 \quad (56)$$

$$a_{13}=\frac{2a^7b^2}{(b^2-a^2)^2(b^4-a^4)} \quad (57)$$

$$a_{14}=0 \quad (58)$$

$$a_{22}=\frac{-a^4b^2}{(b^2-a^2)(b^4-a^4)} \quad (59)$$

$$a_{23}=0 \quad (60)$$

$$a_{24}=\frac{a^8b^2(3a^8-a^6b^2+3a^4b^4-3a^2b^6-2b^8)}{(b^2-a^2)^2(b^4-a^4)^2(b^6-a^6)} \quad (61)$$

$$a_{33}=\frac{a^5b^2(a^4+b^4)}{(b^2-a^2)(b^4-a^4)(b^6-a^6)} \quad (62)$$

$$a_{34}=0 \quad (63)$$

$$a_{44}=\frac{-a^6b^2(b^{10}+2b^6a^6+2b^4a^6+a^{10})}{(b^2-a^2)(b^4-a^4)(b^6-a^6)(b^8-a^8)} \quad (64)$$

On the periphery (r = b), $\frac{\delta\psi}{\delta\theta}=0$

$$\frac{\delta\psi}{\delta r}=-b+\left(a_1-\frac{\beta_1}{b^2}\right)\cos\theta+2\left(a_2b-\frac{\beta_2}{b^3}\right)\cos 2\theta+\left(a_3b^2-\frac{\beta_3}{b^4}\right)\cos 3\theta+\left(a_4b^3-\frac{\beta_4}{b^5}\right)\cos 5\theta+\cdots \quad (65)$$

$$\tau(tangential)=-E\phi\frac{\delta\psi}{\delta r} \quad (66)$$

Substituting the above values in Equation 38, we can get the value of $\frac{\delta\psi}{\delta r}$

$$\tau=-E\phi\frac{\delta\psi}{\delta r}=-E\phi(-b+\left(a_1-\frac{\beta_1}{b^2}\right)\cos\theta+2\left(a_2b-\frac{\beta_2}{b^3}\right)\cos 2\theta+\left(a_3b^2-\frac{\beta_3}{b^4}\right)\cos 3\theta+\left(a_4b^3-\frac{\beta_4}{b^5}\right)\cos 5\theta\ ...) \quad (67)$$

Where θ will be the angle from the reference horizontal axis

Thus, the expression of the Shear stress because of torsion is calculated (Wilson, 1954).

Thus, a Matlab program with inputs eccentricity, outer and inner diameter and the torque can yield a shear stress distribution along the entire lengths of the shaft. This shear stress value can be further extended to get a strain value, which should be compared with the strain value from that of the FEA for verification.

3.6 Calculation of parameters related to bending moment

As far as bending moment is calculated, two disparate stresses are to be obtained. One is the normal stress caused by bending and the other one is the shear stress caused by bending. The calculation of these two parameters for the ideal and the offset cases are shown.

3.6.1 Ideal shaft

Shear stresses due to bending and normal stresses due to bending are the two parameters that are to be calculated. The shear stress at all points will always be opposite in direction to the bending load. The normal bending stress will be perpendicular to the bending load. The direction of the torsional shear stresses and the normal stresses are shown above.

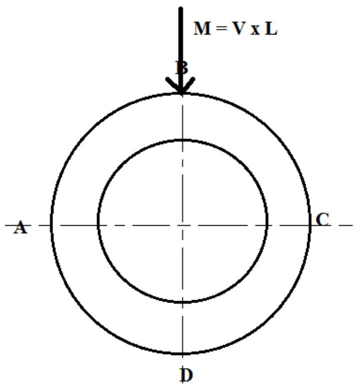


Figure 17 Ideal shaft bending application

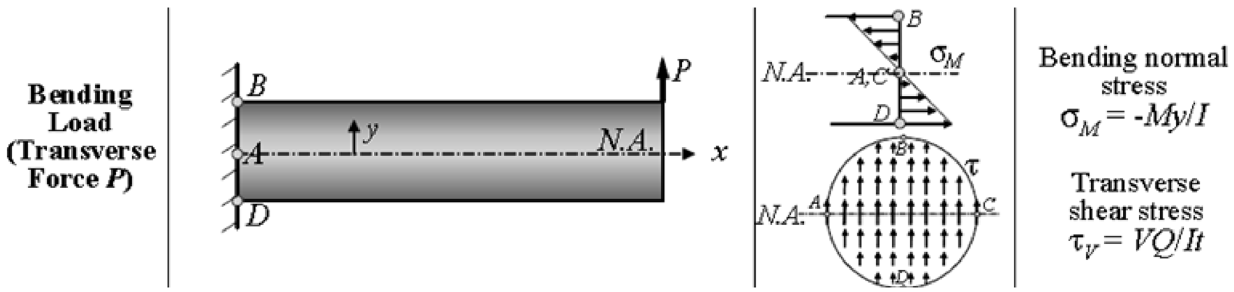


Figure 18 Direction of normal bending stresses and shear bending stresses

Normal bending stress can be found out using the formula $\sigma_b = \frac{-M\times y}{I_{zz}}$ (68)

Shear stress caused by bending can be found out using the formula $\tau_b = \frac{-V\times Q}{I_{zz}\times t}$ (69)

It is to be noted that only the parameters at the outer circumference is intended to be calculated. Along each point, the value of the stress will be different. The bending normal stresses always act perpendicular to the application of the bending load, whereas the bending shear stress is always in direction to that of the bending load application. The direction at each of the locations is shown in the figure. The normal strain will be the normal stress divided by the Young's modulus.

From the above expression, the values changing along the circumference for a particular circular slice (One circular cut at a length l from bending load end) are y and Q. y is the vertical distance of each point from the Centroid of the shaft. Thus, we see that the magnitude of the bending normal stress is maximum at points B and D, whereas it is zero at all the points in the horizontal neutral axis. For the region above the neutral axis, the normal bending stress is always towards the bending load, whereas it is away from the bending load if the region is below the horizontal neutral axis.

In case of shear stress owing to bending, the only term which is complex to find is the First moment of area Q. Q is the First moment of the small area about the neutral axis of the entire body, A is the area. So if the point is between B and C, then A is the area between the point of application of the load B and the point considered.

3.6.2 Eccentric hollow shaft

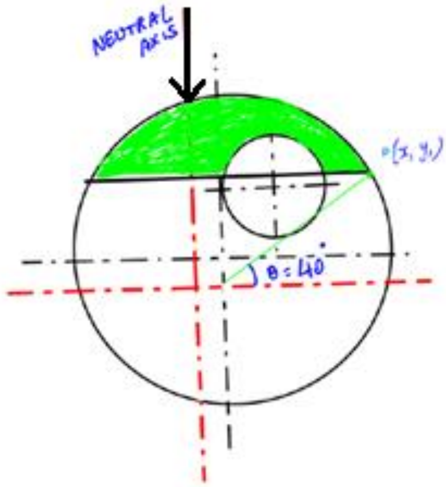


Figure 19 Area moment of inertia for an eccentric shaft at an arbitrary point

Normal bending stress can be found out using the formula $\sigma_b = \frac{-M \times y}{I_{zz}}$ (70)

Shear stress caused by bending can be found out using the formula $\tau_b = \frac{-V \times Q}{I_{zz} \times t}$ (71)

Here, an eccentric shaft is considered. As a result, the neutral axis is changed and is indicated in the above figure. Now if the two stresses are to be computed at point P, then it shall be as follows:

Here y is the distance from centroid, which can be calculated once the centroid of the entire shaft is known (which is along the neutral axis). I_{zz} is the Second moment of inertia and the procedure for calculating it is already mentioned in the second chapter. First moment of inertia along point (0, 0) just considering the outer circle is found. Using parallel axis theorem, Moment of inertia for the outer circle along the neutral axis if found. Similarly, MOI for the inner circle along the neutral axis is found after knowing the MOI of the inner circle along the central point. Thus, effective MOI along the neutral axis Izz is found for the hollow shaft. Once Izz, y and M are known, normal bending stress can be calculated.

For shear bending stress, V is the bending load in N, t is thickness at each point of consideration and Q is the first moment of area. For the point P in the above figure, the area to be considered is highlighted. Q will be product of the highlighted area and the distance vertical between the centroid of the highlighted portion and the centroid of the entire shaft. Thus, using this way, bending stress along each point along the outer circumference can be calculated.

3.7 Calculation of parameters related to axial load

There is no Shear stress associated with an axial load. The direction of the normal stresses will be in same direction to that of the axial load.

The normal stress owing to axial load is $\sigma_a = \frac{P}{A}$, where P is the load in N and A is Area of the shaft in mm^4.

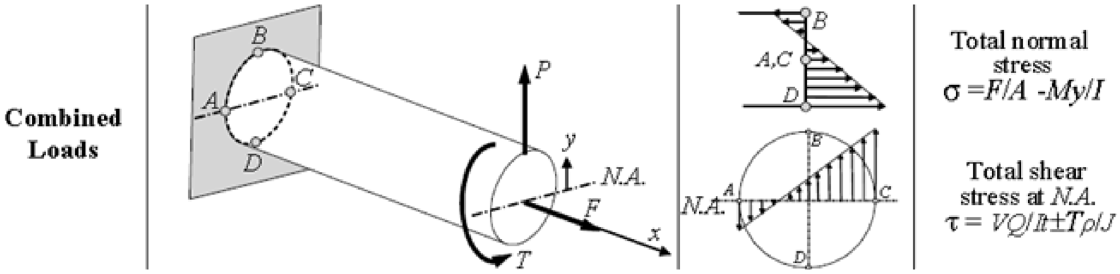


Figure 20 Axial load combinations

3.8 Application of strength of materials in strain gauges

So far, the following parameters are found

- Normal stress due to axial load
- Normal stress due to bending load
- Shear stress due to bending load
- Shear due to Torsion.

The strain gauges will be placed along the sides of the transducer. For illustration purposes, an eccentric shaft with a strain gauge placed on it is considered. It is to be noted that the eccentricity is exaggerated primarily for illustration.

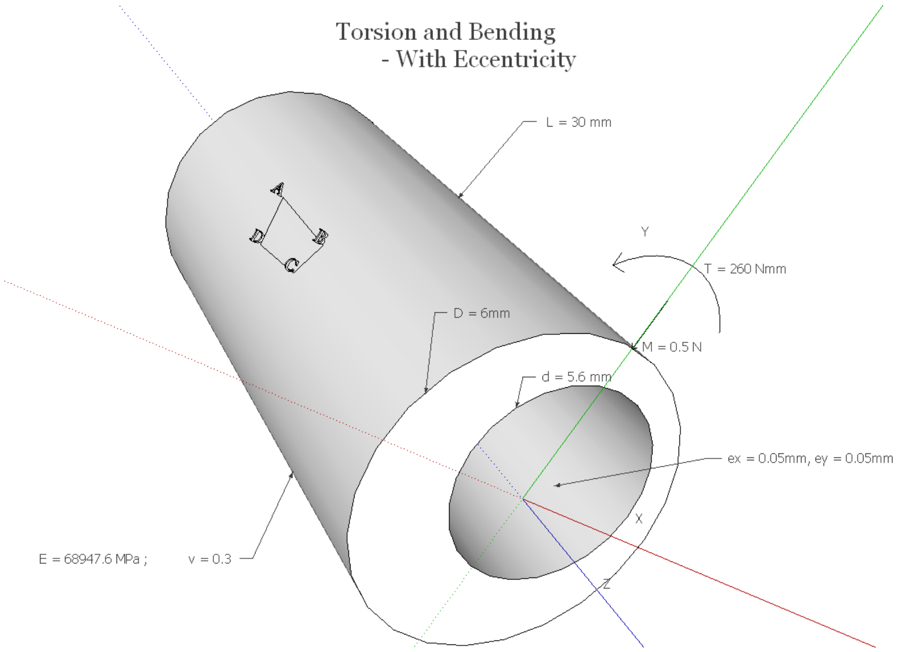


Figure 21. Strain gauge location identification

The strain gauges are always placed along the Z direction (from the above figure) for measuring torque. As a result, all the stresses except the shear stress caused by bending will have an effect in the measurement of the strain. The shear stress caused by bending will act along the negative y direction (as shown in the figure). This stress doesn't have any influence in the deformation of the strain gauge represented by the area ABCD as they are in different plane (Hoffman). As a result, only three of the parameters are to be used in computing the strain that will be sensed by the strain gauge which is the normal stress caused by axial load, normal stress caused by bending load and the Shear stress caused by torsion. Nevertheless, shear stress because of bending load is computed in Matlab in Appendix 2, just for showing their magnitude.

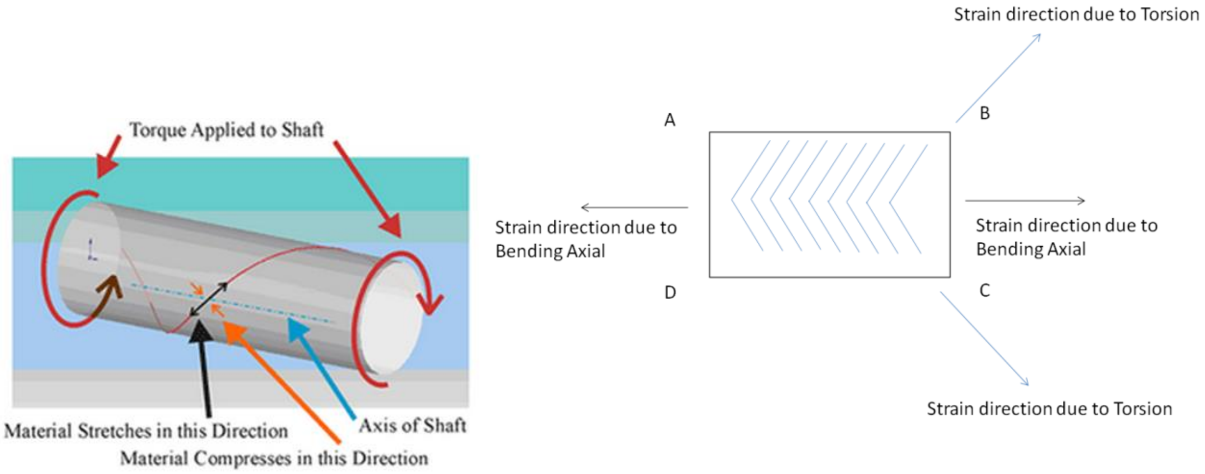


Figure 22. Strain gauge sensing strain due to torsion

3.9 Calculation of strain at a point

The important step is to analyse the strain at different directions and subsequently to calculate the strain that a strain gauge will sense.

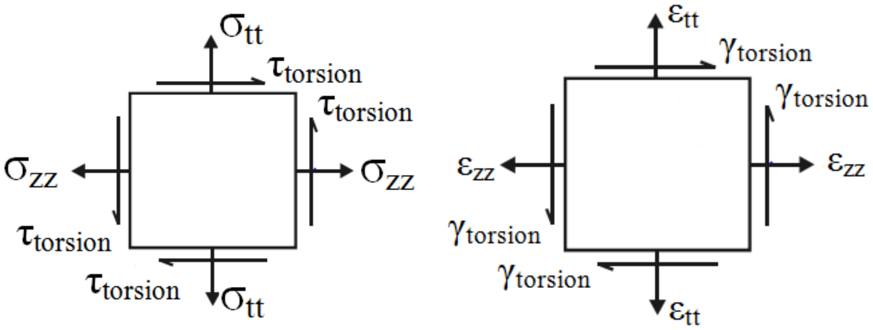


Figure 23. Plane stress system

The above figure shows the stress directions that act along different directions for a considered region (say ABCD from figure). The only shear stress that will have an impact on the strain gauge is the Torsional shear stress (which gives rise to principal strains along 45 degrees). The normal strain acts along the z direction as shown in the figure and their subsequent strain will be ε_{zz}. However, there will be a strain acting along the tt direction (i.e. along the ϕ direction around the surface). This strain is due the transverse sensitivity of the strain measurement and this strain is represented by the product of the strain along z direction multiplied by the poison's ratio.

So, if the value of ε_{zz}, ε_{tt} and $\gamma_{torsion}$, are calculated, the next step is to calculate the strain at 45 degrees and also at other angles. Using strain transformation, strain at any angle can be found out using the formula,

$$\varepsilon_{angle} = \left(\frac{\varepsilon_{zz}+\varepsilon_{tt}}{2}\right) + \left(\frac{\varepsilon_{zz}-\varepsilon_{tt}}{2}\right)\cos 2\theta + (\gamma_{xy}/2)\sin 2\theta \quad (72)$$

Since the principal strain would not be at 45 degrees (because of the presence of bending load), there is no use of calculating the maximum and minimum principal strains. Hence using the above expression, strain at any angle can be computed.

By this method, strain along any direction can be calculated provided the three values are found out.

3.10 Calculation of strain over a region

Strain at any point along any direction can be found out using the previous section. The next step is to calculate the strain over a region. Strain at a region can be calculated using the average strain over that region. In Matlab, at a particular angle, the strain acting at all points along the specified direction of the entire shaft is found out.

Each strain point calculated will have two parameters for indexing. The first one is the length from the load end, while the second one is the angle from the horizontal x axis in the xy plane. In Excel, the coordinates of different points of the interested region is mentioned. For example, the following figure shall be considered. Here, the coordinates of points A, B, C and D shall be mentioned in the Excel sheet. The length coordinate is distance of the particular point from the load end. The theta coordinate is the angle with respect to the XY plane measuring from the horizontal x axis. Thus the length coordinate and theta coordinate of the four points are to be mentioned in the input excel sheet. Apart from this, the resolutions of the length and theta coordinates are to be mentioned.

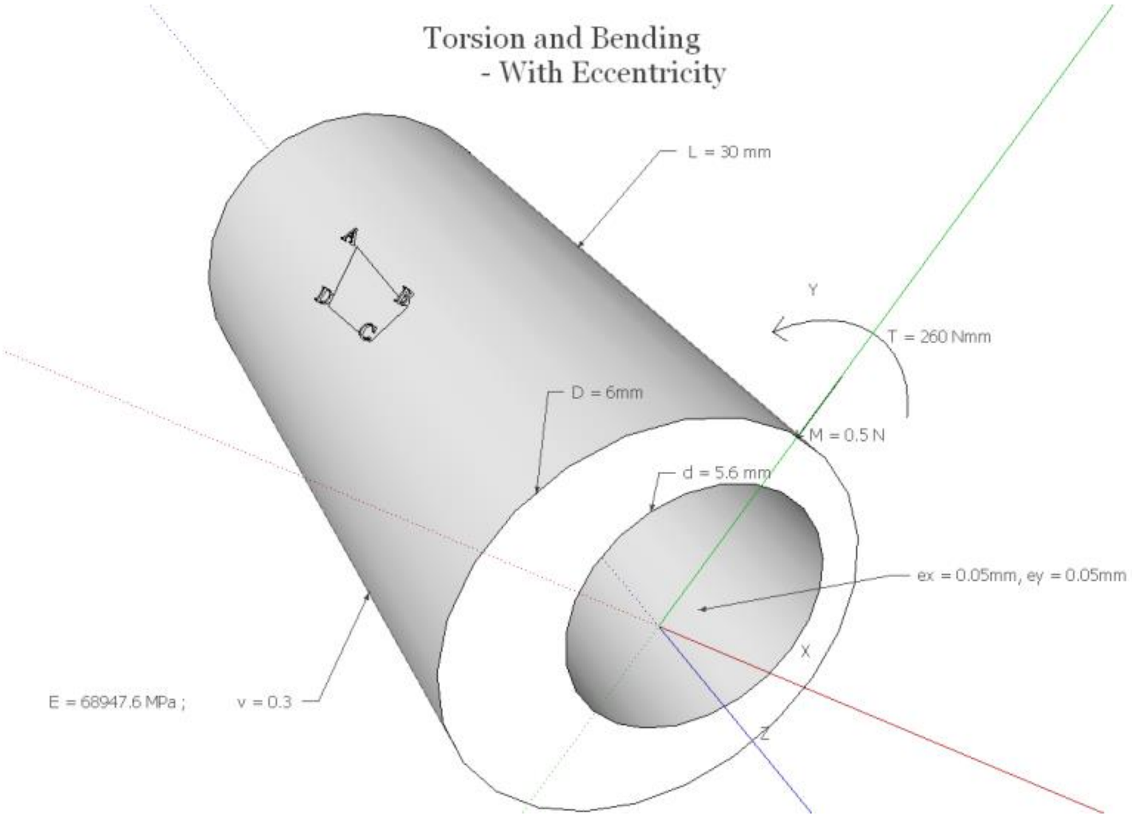


Figure 24 Strain gauge location

The lower the resolution, the more accurate the results will be, but at the expense of time. A lowest resolution of 0,2mm for lengths and 0,1 degrees for angle is recommended. A resolution of 1degree and 1mm shall fetch the result with a minute, and thereby it is recommended for quick checks.

Consider in this case, a shaft of 30mm length with appropriate loads with a resolution of 1 degree and 1 mm length. The array of the entire strain points will look like this.

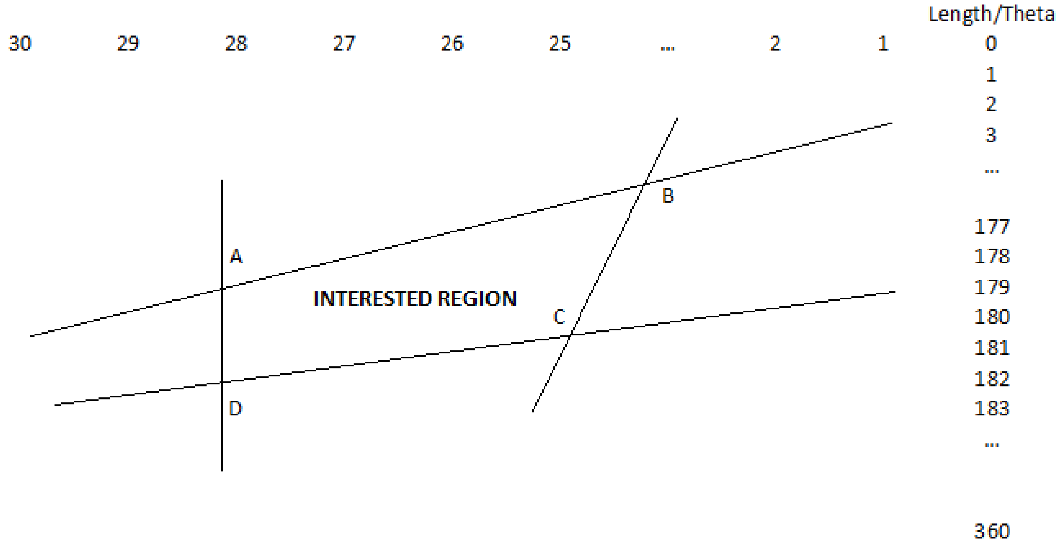


Figure 25 interested region in strain array

Once the four points are located according to the length and theta coordinates, the next step is to find the average strain at the interested region. The above figure shows an uncommon strain region which the user is interested. Once the four points are marked, then the equations of the four lines are expressed in the Matlab code. After that the slope of the four lines are found out. All the points below line AB will be marked. Then all the points above line DC will be marked. Then, the points to the left of the line BC and to the right of the line AD will be marked. Then, the points with the region ABCD will be the intersection of the points of the four lines. Thus, all the interested points will be known using the theta and length coordinates.

Thus the average strain over a particular region can be found out. As far as the strain gauge is concerned, average strain over two angles (usually +45 and 315) will have to be found out. The same procedure will have to be repeated for the different strain regions which the user is interested in.

3.11 Wheatstone bridge and sensitivity

Sensitivity of a strain gauge refers to how sensitive a strain gauge will be to different conditions. The sensitivity of a strain gauge can be further worked upon to give the torque that the torque transducer shall indicate. However, sensitivity can be used as a relative term to measure the sensitivity with an ideal case. Usually sensitivity is indicated in mV/V.

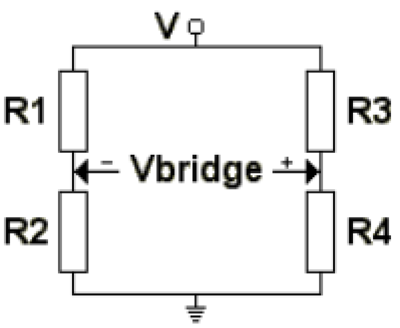


Figure 26 Wheatstone bridge circuit

$$Sensitivity = \frac{V_{bridge} * 1000}{V}$$

The resistance values are nothing but the change in resistances because of the change in deformation.
Resistance and Strain are related through the following expression

$$Gauge\ Factor = \frac{\frac{\Delta R}{R}}{\epsilon}$$

$$\frac{\Delta R}{R} = \frac{V_{bridge}}{V} = GF \times \varepsilon$$

Thus, another way of computing the Sensitivity is using the change in deformation and Gauge Factor

$$Sensitivity = Gauge\ Factor * \left(\frac{\varepsilon_1 + \varepsilon_2 + \varepsilon_3 + \varepsilon_4}{4}\right)$$

ε_1 – Absolute value of tensile strain along region 1 (subjected to increase in deformation)
ε_2 - Absolute value of compressive Strain along region 2 (subjected to decrease in deformation)
ε_3 - Absolute value of tensile strain along region 1 (subjected to increase in deformation)
ε_4 - Absolute value of compressive Strain along region 2 (subjected to decrease in deformation)

In this thesis work, the final output is considered the Sensitivity of the strain gauge. The sensitivity can be further worked upon to obtain the torque that shall be indicated by the measurement system, but it is not done since the control signals are considered outside the scope of the thesis.

Thus, from now on, sensitivity that an ideal shaft displays is compared with the sensitivity than an imperfect shaft (because of geometrical tolerances) shows is compared.

In the results chapter the results that are obtained with the methods described in the method chapter are compiled, and analyzed and compared with the existing knowledge.

4.1 Verification of geometrical properties

Centroid verification was carried out using Autodesk Inventor Professional 2014 (Student version).

4.1.1 Ideal shaft geometrical properties

For a shaft of 25mm outer diameter and 15 mm inner diameter, the geometrical properties from Inventor and from Matlab are as follows:

Table 9 Geometrical properties of ideal shaft

	Inventor	Matlab
Area (mm^2)	314.159	314.159
Centroid X (mm)	0	0
Centroid Y (mm)	0	0
I_{xx} about Neutral Axis (mm^4)	16689.711	16689.711
I_{yy} about Neutral Axis (mm^4)	16689.711	16689.711
Polar Moment of Inertia (mm^4)	33379.422	33379.422

4.1.2 Eccentric shaft geometrical properties

For an eccentric shaft of 25mm outer diameter and 16 mm inner diameter with offsets 1mm and 1mm in both directions at a particular instant, the geometrical properties from Inventor and from Matlab are as follows:

Table 10 Geometrical properties of eccentric shaft

	Inventor	Matlab
Area (mm^2)	289.812	289.812
Centroid X (mm)	-0.6938	-0.6938
Centroid Y (mm)	-0.6938	-0.6938
I_{xx} about Neutral Axis (mm^4)	15276.66	15276.66
I_{yy} about Neutral Axis (mm^4)	15957.77	15957.77
Polar Moment of Inertia (mm^4)	31234.43	31234.43

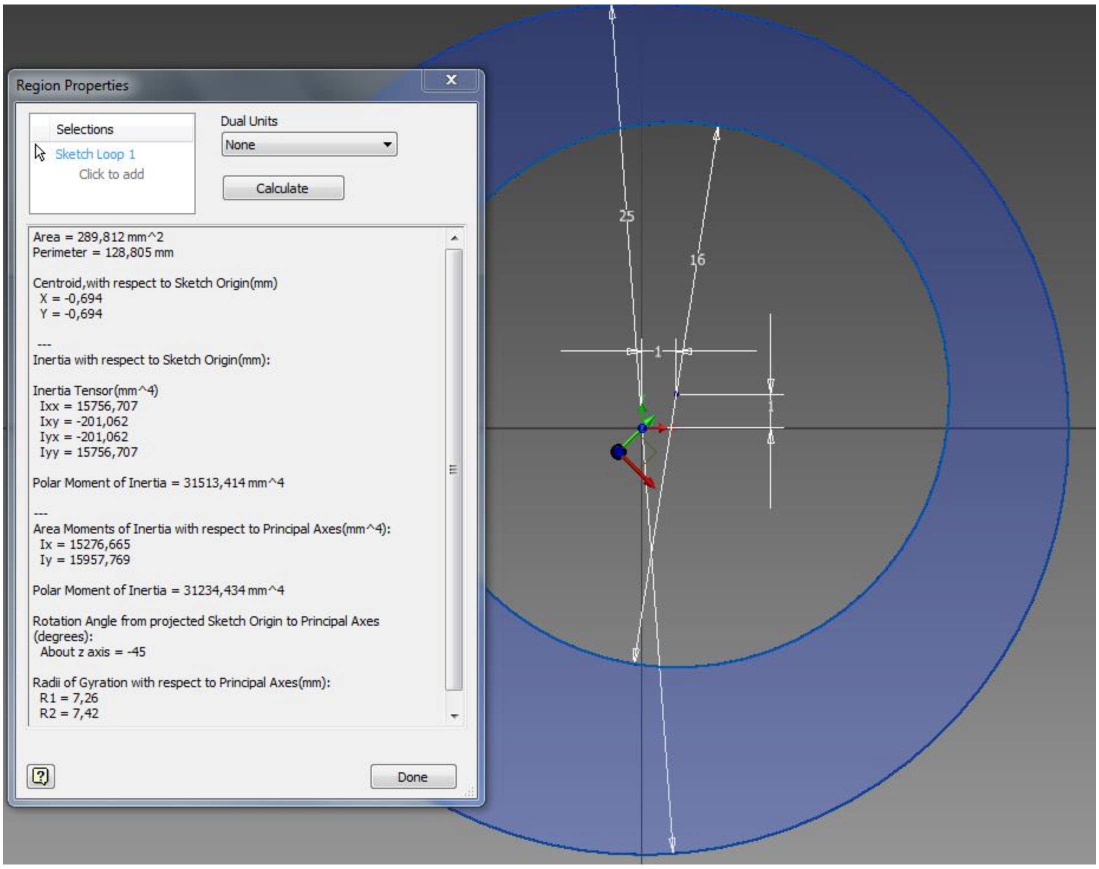


Figure 27. Eccentric shaft in inventor for geometrical properties

4.1.3 Hemispherical shaft geometrical properties (for first moment of area)

For a hemispherical eccentric shaft of 25mm outer diameter and 15 mm inner diameter with offsets 2 mm and 2 mm in both directions at a particular instant, the geometrical properties from Inventor and from Matlab are as follows

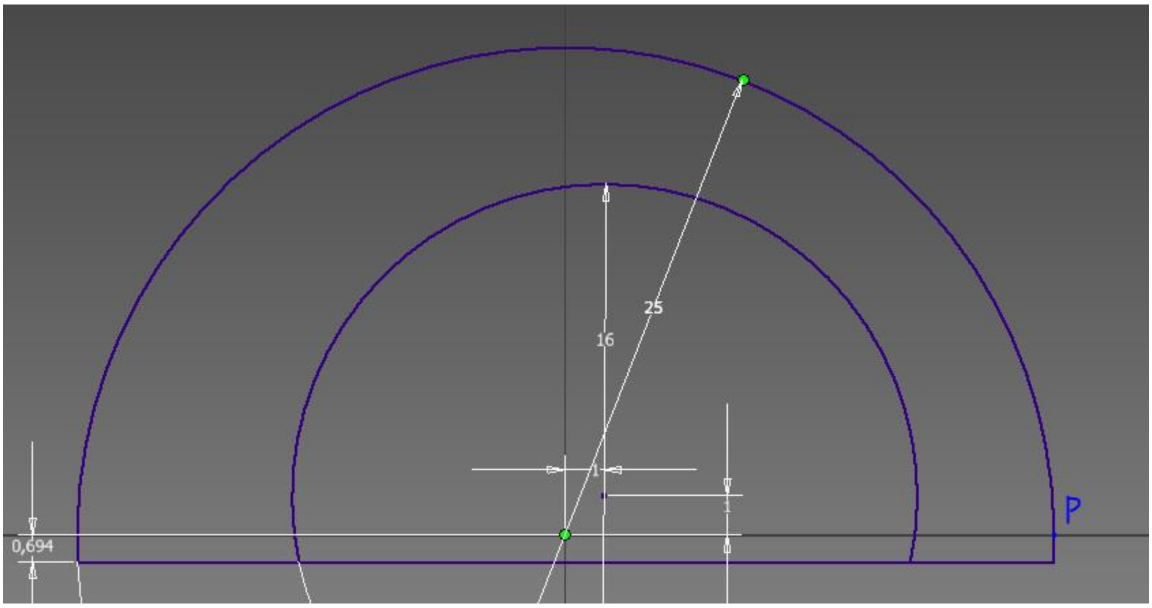


Figure 28 Hemispherical shaft specifications

The above figure is just to verify the values of the first moment of area, which is an important parameters in measuring the shear stress caused by bending. The first moment of area Q is defined as the product of the Area of the section at that considered point multiplied by the differences in centroid. Now, let the case for a point on the centroid be considered. Here, the area will be the highlighted portion as shown in the figure below and y shall be the distance between the centroid of the entire body and the centroid of the highlighted region , while area 'A' shall be area of semicircle. The value of A and y obtained from Matlab were the same, thus leading to the conclusion that geometrical properties are verified.

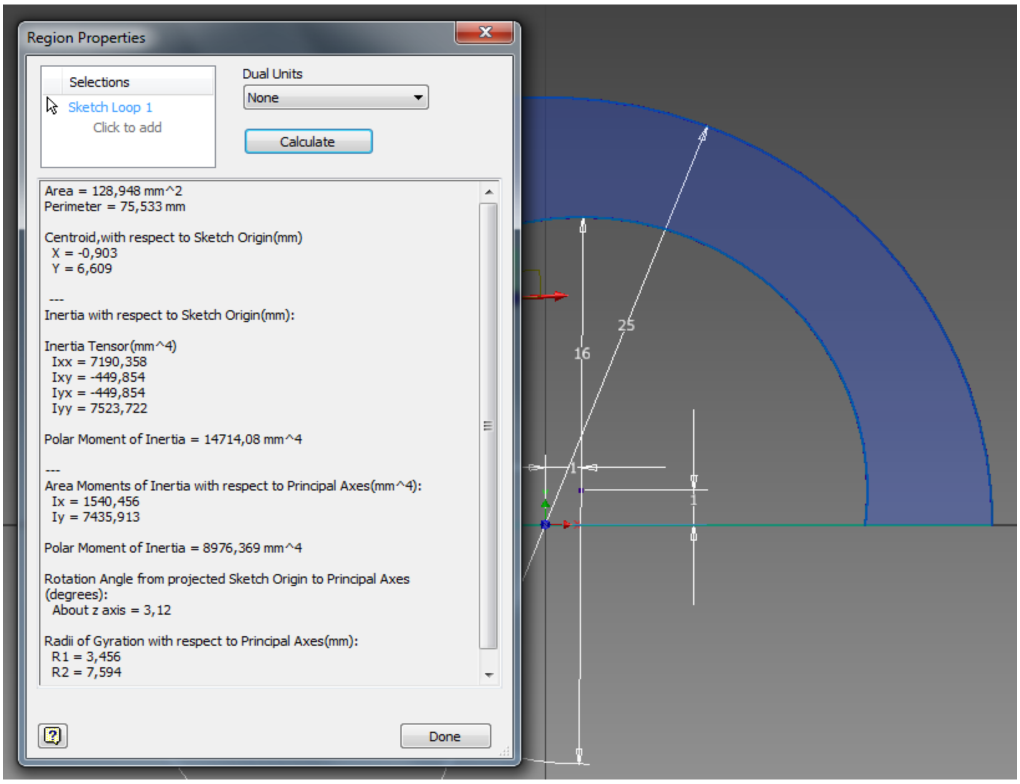


Figure 29 Hemispherical shaft in inventor for geometrical properties

4.2 Twisting moment verification with FEA

Verification of the strain values are carried out using Pro Engineer Mechanica (using p-type method). The polynomial order of minimum 3 and maximum 9 was used in the FEA Analysis. A fairly closer percentage convergence of 1 was used for the Multi-pass adaptive method in analysis. The convergence was measured along different geometry when required.

4.2.1 Solid shaft and concentric shaft

In case of a solid shaft, the shear stress value will be the same through all the points in the outer surface of the shaft. A typical example of an Aluminum shaft with 20 mm diameter and 40 mm length subjected to 100 Nmm Torque is considered. Figure 30 shows the maximum and minimum principal strain for the considered case. It can be seen that along the outer surface, the strain values are the same.

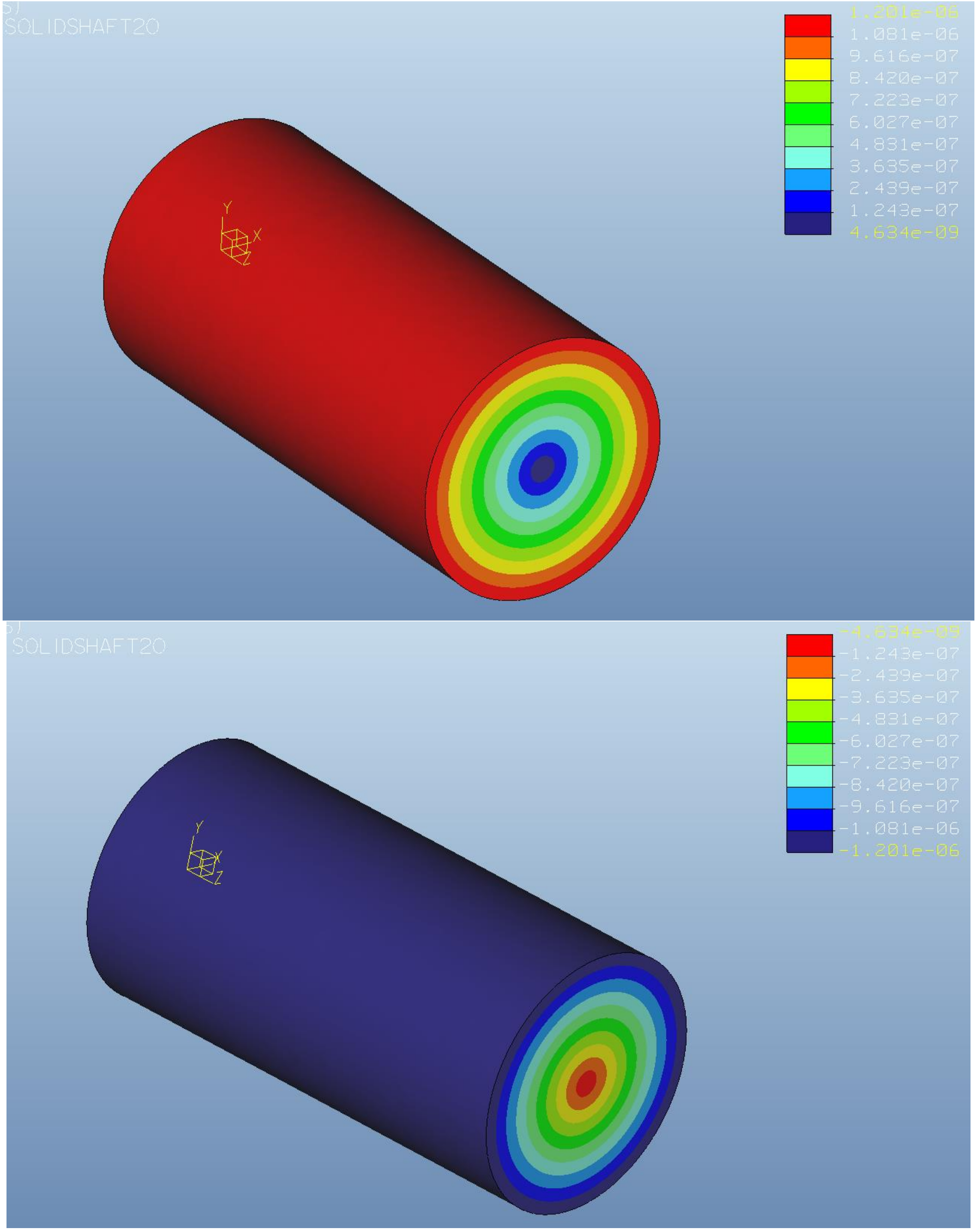


Figure 30 Maximum and minimum principal strain values for a solid shaft

From the ProE Mechanica FEM Analysis of a solid shaft, the maximum principal strain and the minimum principal strain are correspondingly constant throughout the outer surface of the shaft. The maximum and minimum principal values obtained are 1.201 με and -1.201 με at the outer surface respectively.

Using the analytical method (Matlab Code), the value of the maximum and minimum principal strains are 1.200 με and -1.200 με. It is safe to conclude that this simple model yields the same result in analytical and FEA method.

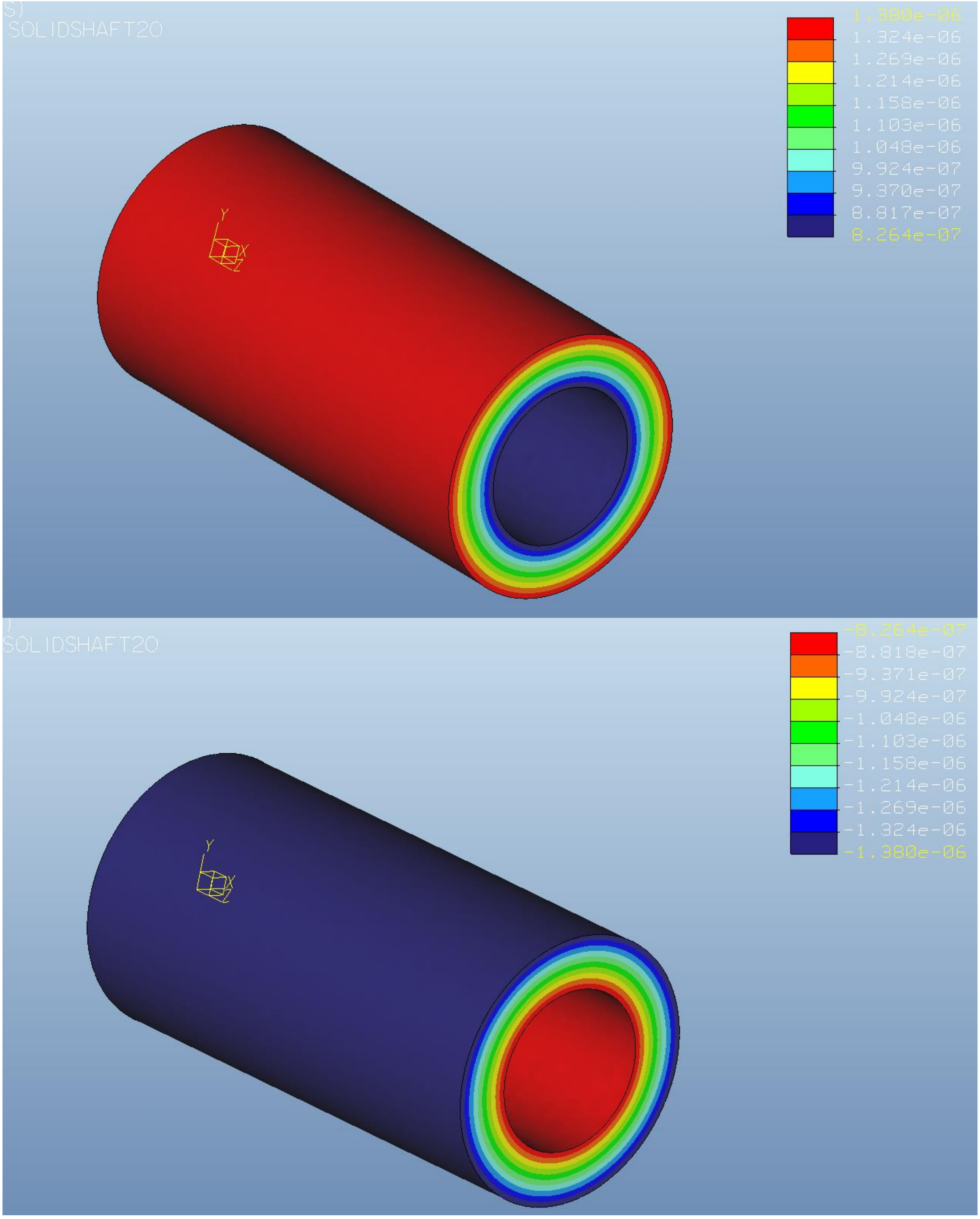


Figure 31 Maximum and minimum principal strain values for a concentric hollow shaft

Similarly, the analytical method yielded a maximum and minimum principal strain as 1.379 $\mu\varepsilon$ and -1.379 $\mu\varepsilon$ for a hollow shaft with 12 mm hole. These values are close to the maximum and minimum principal strain values (1.380 $\mu\varepsilon$ & -1.380 $\mu\varepsilon$ respectively) as observed from the FEA.

4.2.2 Eccentric shafts

An eccentric Aluminium shaft of outer diameter 20 mm and inner diameter of 12 mm is considered to illustrate the verification with FEA. The eccentricity value (viz, offset value) between the center of outer diameter and the center of inner diameter is considered as 0.3 mm. A twisting moment of 100 Nmm is applied at one end of the shaft. This twisting moment will cause a deformation along the outer surface of the shaft. The expression for calculating the shear stress is already mentioned in previous chapter. Maximum and minimum principal strains can be computed once the shear stress value is known as explained earlier.

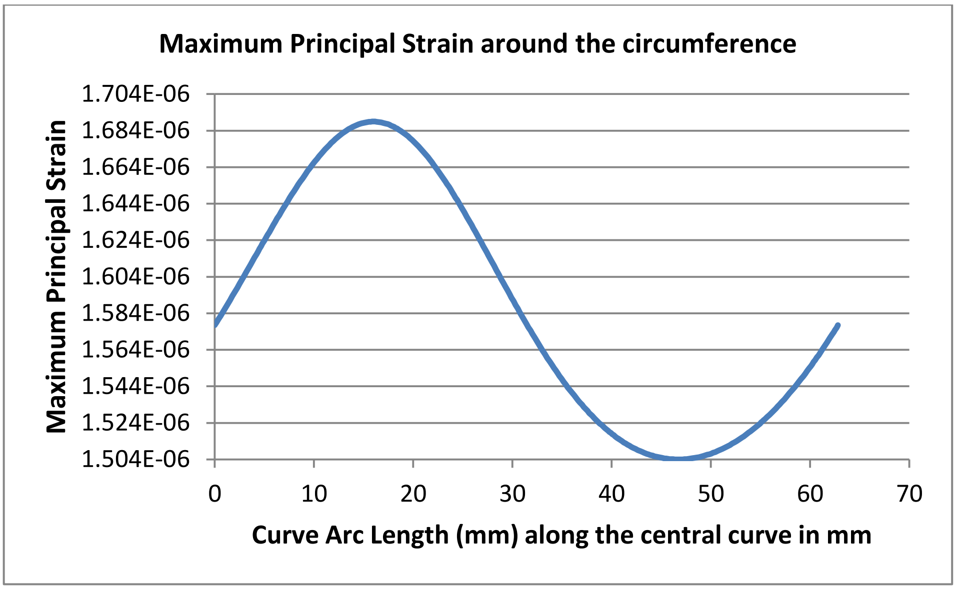


Figure 32 Maximum principal strain value for a shaft along the centre- FEM

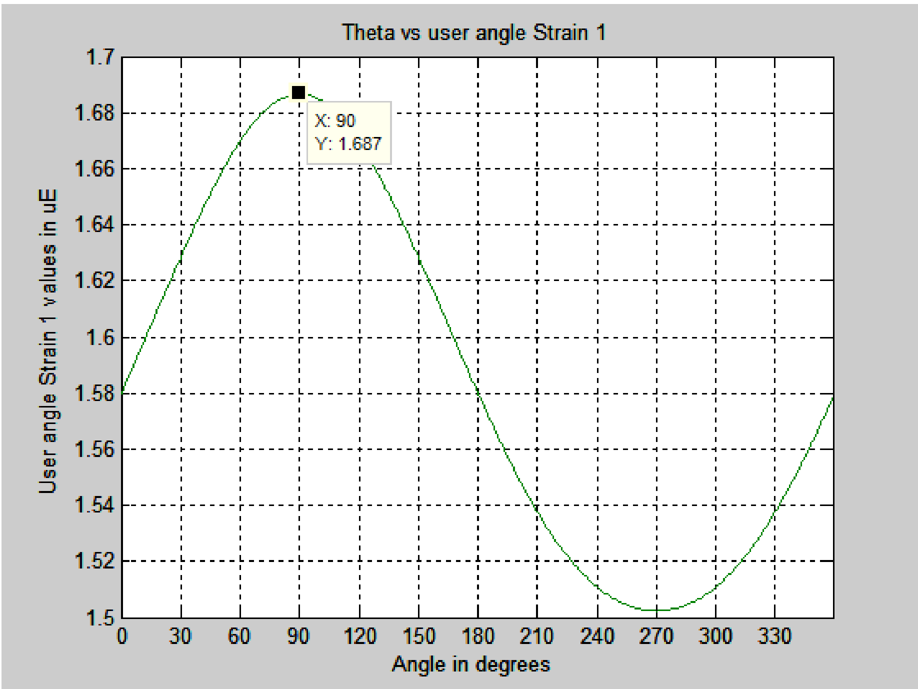


Figure 33 Maximum principal strain value for a shaft - Analytical method

Figure 32 shows the profile of the maximum principal strain along the center of a shaft using ProE Mechanica. It follows a sinusoidal curve and the profile is exactly the same as that of the plot obtained by analytical method. The values are close to each other within 1% range, which is a fairly tantamount to the FEA plot.

To make an accurate comparison between the two plots, the values from Mechanica are imported as an excel sheet. Then, the curve arc length is transformed to degrees around the shaft. After, that using interpolation, values are obtained and are plotted.

The following case is for an aluminium shaft with outer diameter 12 mm and inner hole 10 mm. The eccentricity is set as 0.1571 mm with a length 30 mm. The profiles of the two plots are compared and the error between them at all the angle values between 0 and 359 is obtained.

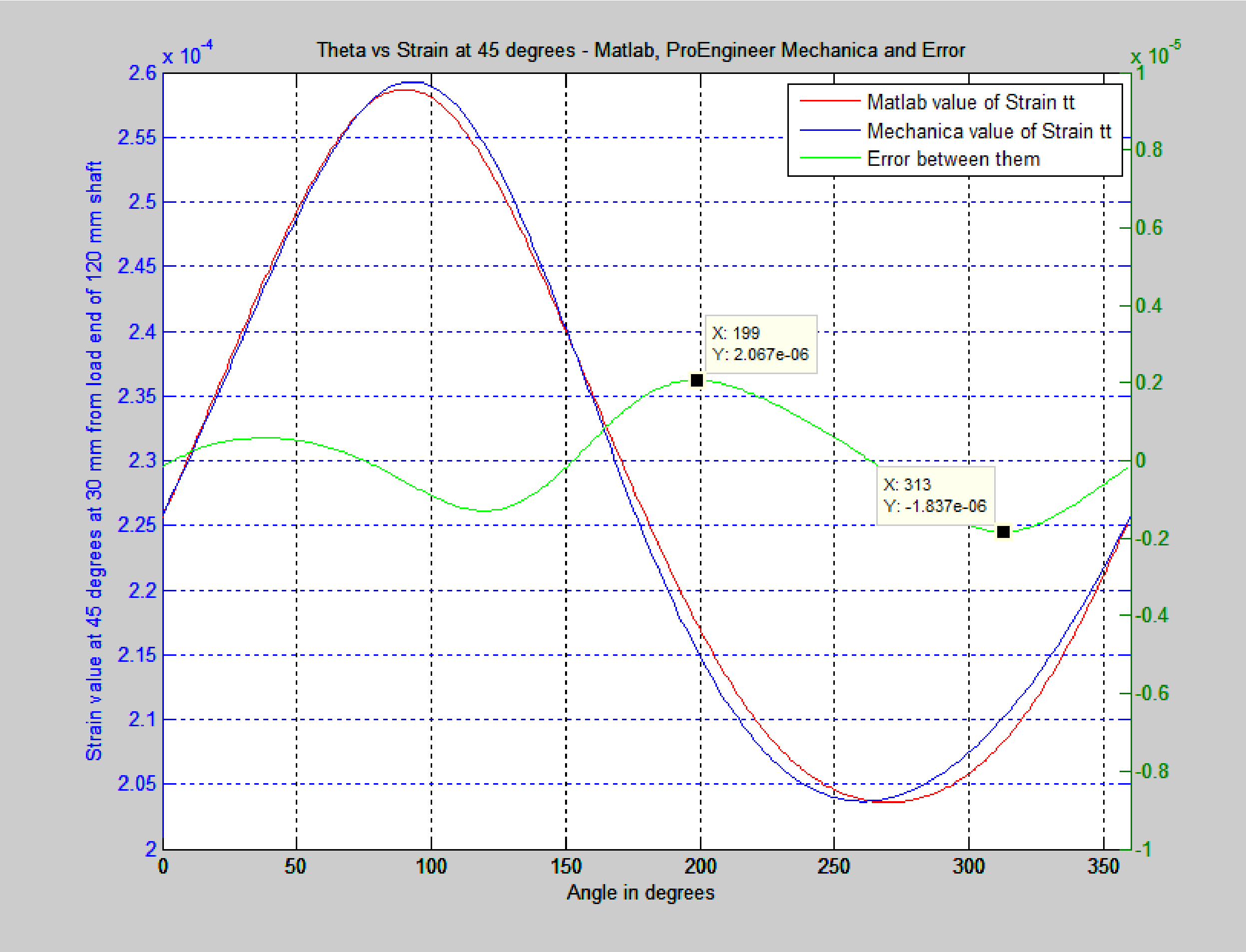


Figure 34 Comparison of strain between Matlab and Mechanica

It can be seen from the above figure that, the error is between 2 με to -1.8 με for the strain values. A maximum of 1 % error is observed between the analytical model and the FEA model which is attributable to the least possible boundary condition definition of 1% in Mechanica.

The maximum strain occurs at 90 degrees, which is understandable owing to the fact that the eccentricity is the least at 90 degrees as per the data entered. In a brief conclusion, the strain will be maximum along the lengths where the distance between the outer and inner diameter radially is the least (thinnest wall).

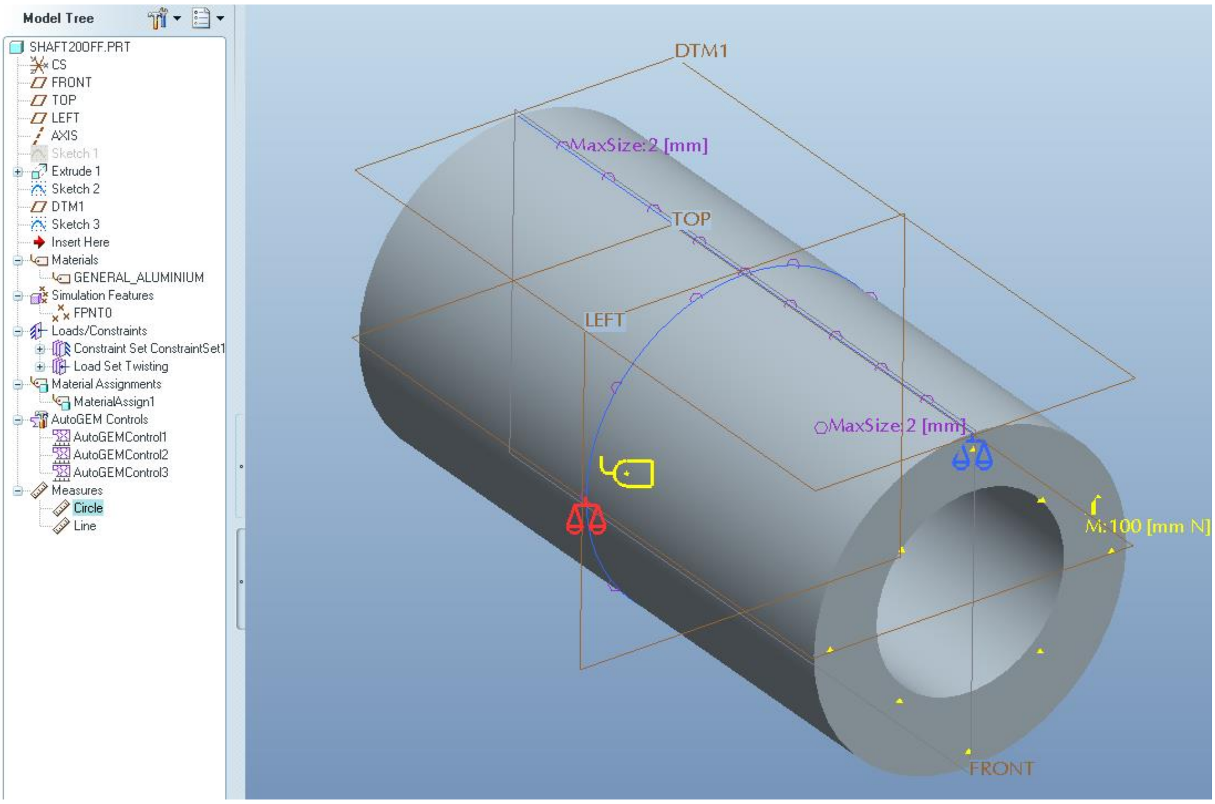


Figure 35 ProE Mechanica Shaft Illustration

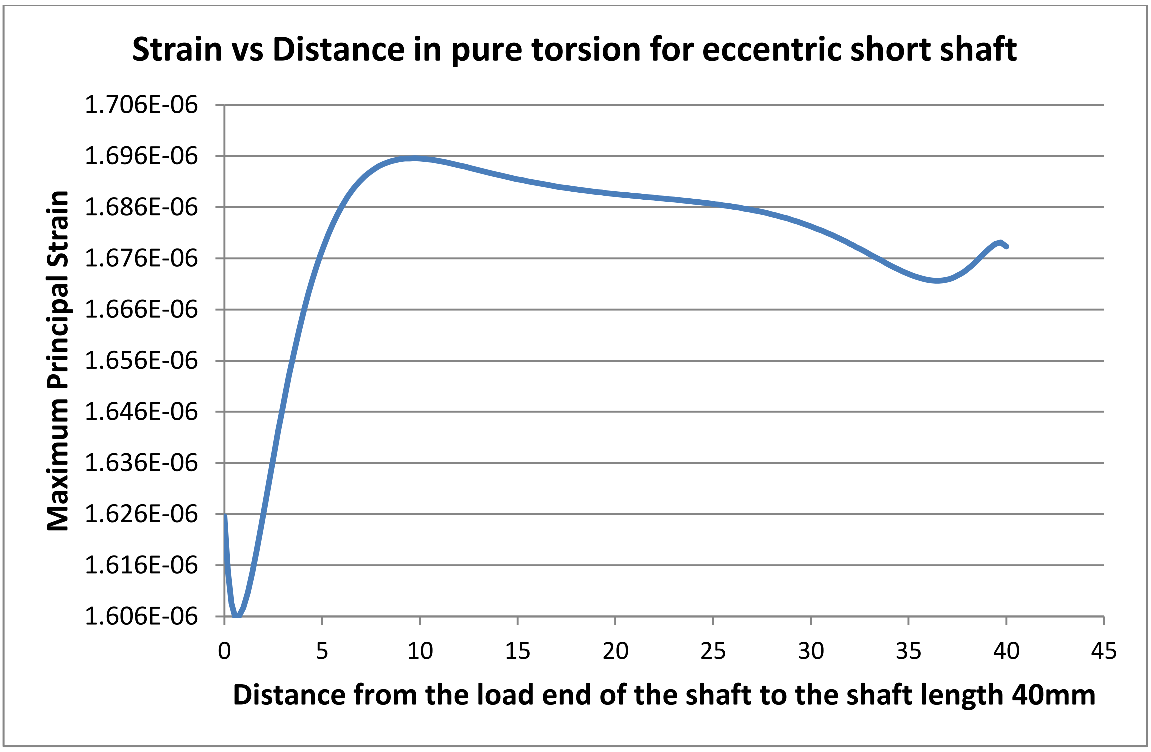


Figure 36 Maximum principal strain along a reference line for a 40mm long shaft

The maximum principal strain variation at one particular angle along the length of the shaft is an interesting observation. Figure 36 shows the variation of strain along the length of the shaft where the eccentricity is the least. In other words, the plot shows the strain distribution along the horizontal line (on the top of the shaft) drawn along the shaft as seen in Figure 35.

It can be observed the maximum principal strain varies along the length. However, it shouldn't happen owing to the fact that a twisting moment should cause the same deformation along the lengths of the shaft. The possible explanation for this is attributable to the effect of constraints at the other end of the shaft when loaded in Mechanica. The length of the shaft was increased from 40 to 120 with the other parameters remaining the same.

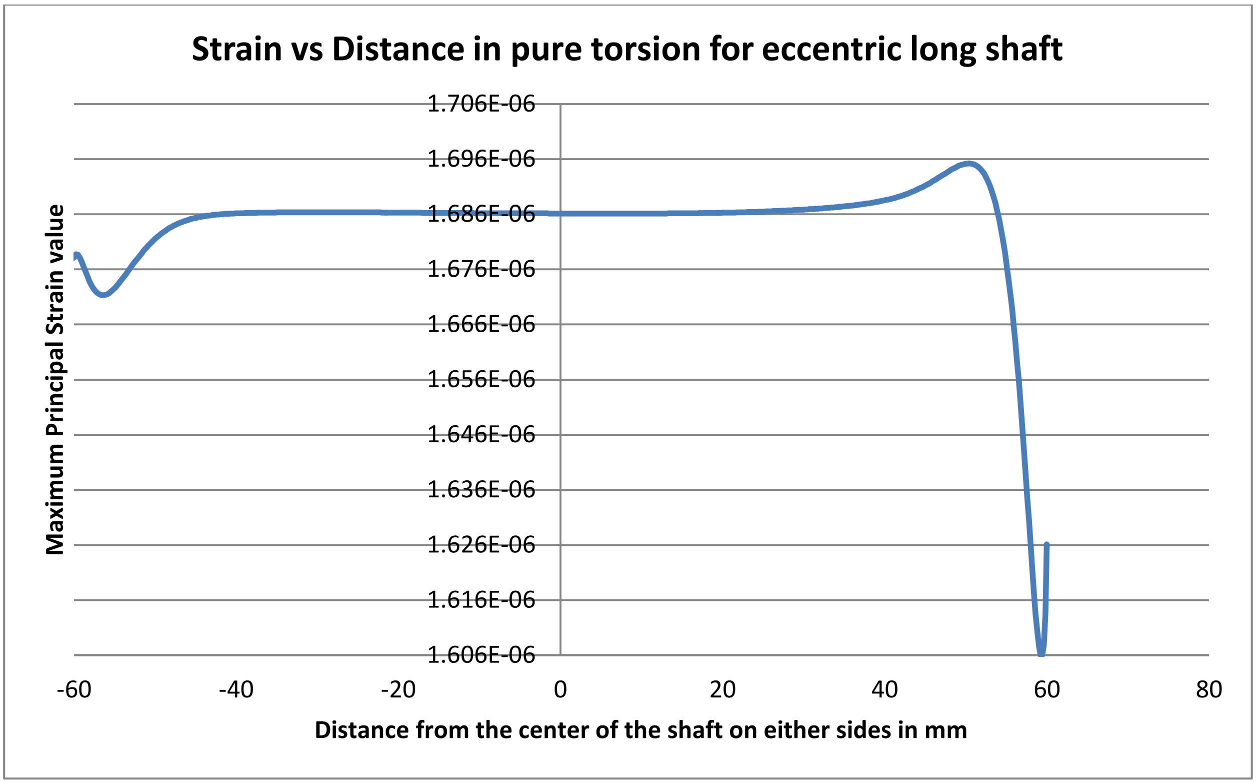


Figure 37 Maximum principal strain along a reference line for a 120mm long shaft

Figure 37 shows the strain distribution of a shaft subjected to torsion for a 120 mm shaft but with the same parameters as in the previous case. It is observed that the strain value remains the same for almost 80mm of the shaft. Apparently, this value is around 1.686 με which is the same as the one computed using Analytical method.

The values were also checked for many shaft lengths are tabulated below. The corresponding Analytical values are also computed.

The same method was applied for a steel shaft of length 40mm, outer diameter of 25 mm and inner diameter of 15 mm with 0.7 mm eccentricity. The analytical solution at a particular point yielded a result of 0.763 με. In FEA, the value was within 0.758 με to 0.765 με and the normalized value being 0,762 με which suggests an accuracy of 0.1%. However, it is to be noted that 0.3mm and 0.7 mm eccentricity were used just for illustration purposes and to magnify the effects of tolerances and it is an un-realistic value in manufacturing industry for tolerances.

4.3 Verification of ε_{zz} and ε_{tt} with FEA

The most important parameters to be verified as far as bending is concerned are ε_{zz} and ε_{tt}. These are the strain caused by bending load. ε_{zz} is due to bending and axial effect whereas ε_{tt} is because of Transverse sensitivity (by Poisson's ratio).

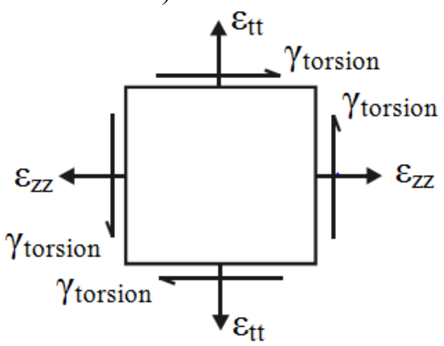


Figure 38 Stress plane

Following are the different scenarios and the profiles of the strain along a particular length are mentioned. The Ten different cases in bending are as follows:

i. Solid Shaft subjected to bending
ii. Concentric hollow shaft subjected to bending
iii. Eccentric hollow shaft 1 subjected to bending
iv. Eccentric hollow shaft 2 subjected to bending
v. Solid Shaft subjected to bending and torsion
vi. Concentric hollow shaft subjected to bending and torsion
vii. Eccentric hollow shaft 1 subjected to bending and torsion
viii. Eccentric hollow shaft 2 subjected to bending and torsion
ix. Solid Shaft subjected to only Axial load
x. Solid shaft subjected to bending, torsion and axial load

For each case, the plot from FEA is compared with the plot from Matlab. While the shear stress (and subsequently strain) due to torsion is already verified, the strains ε_{zz} and ε_{tt} are to be verified.

ε_{zz} and ε_{tt} calculated from ProE Mechanica have their x axis as the curve arc length. In Mechanica, the assignment of theta from 0 to 360 as x coordinate couldn't have been made. As a result, the horizontal coordinate as curve arc length was the default one that could be set as horizontal x coordinate.

In Mechanica, the following specifications are employed during analysis. Polynomial order of minimum 3 and maximum 9 is ensured in Mechanica with the percentage convergence being set to 1. The average mesh size of 2 mm is employed for the component, with 1 mm average mesh size for the interested curves or lines.

4.3.1 Solid shaft – only bending

Table 11 Different input values for a solid shaft subjected to only bending

Parameters	Value
D (in mm)	25
d (in mm)	
L (in mm)	40
ex (in mm)	
ey (in mm)	
Torque (in Nmm)	
Bending Load (in N)	10
Considered point for verification from load end (in mm)	20

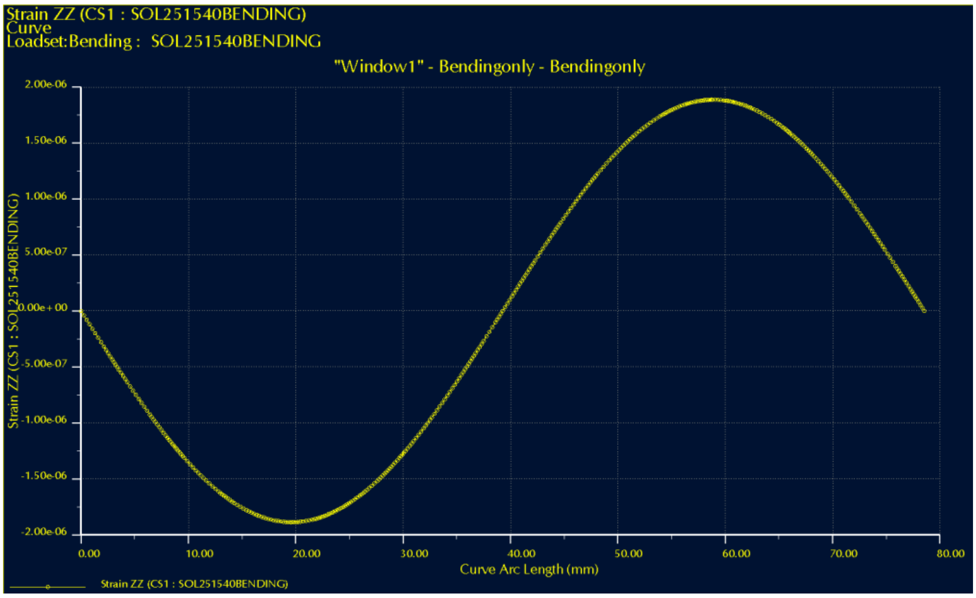


Figure 39 theta vs ε_{zz} in ProEngineer Mechanica

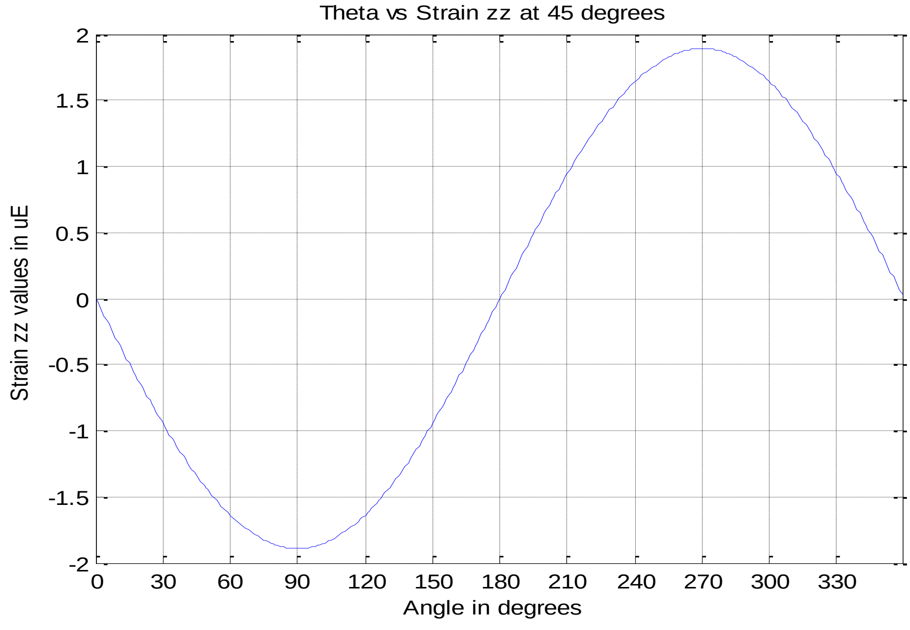


Figure 40 theta vs ε_{zz} theta in Matlab

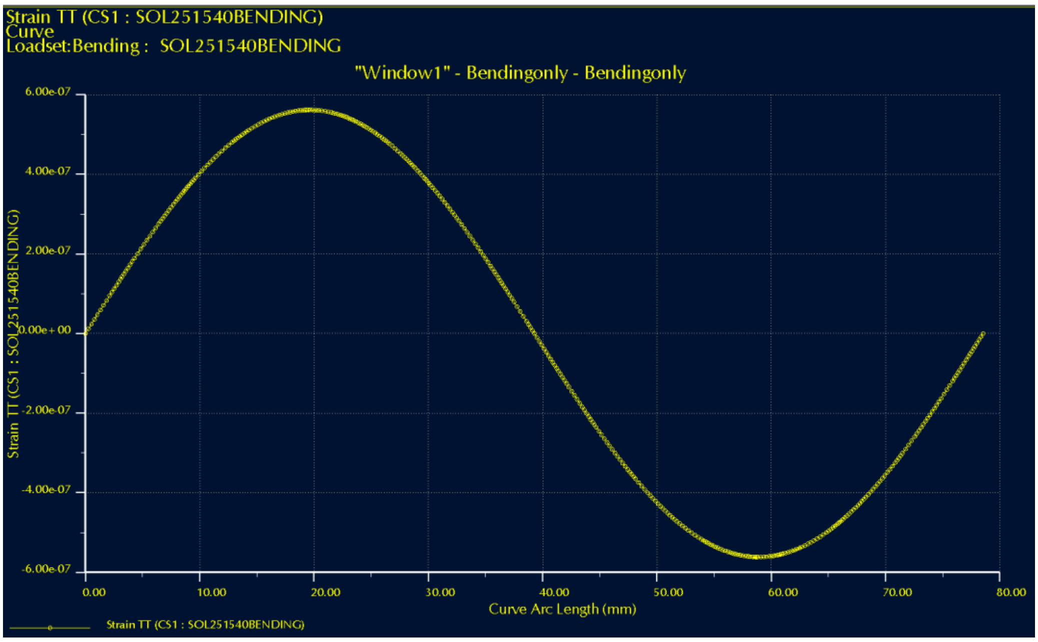


Figure 41 theta vs ε_{tt} in ProEngineer Mechanica

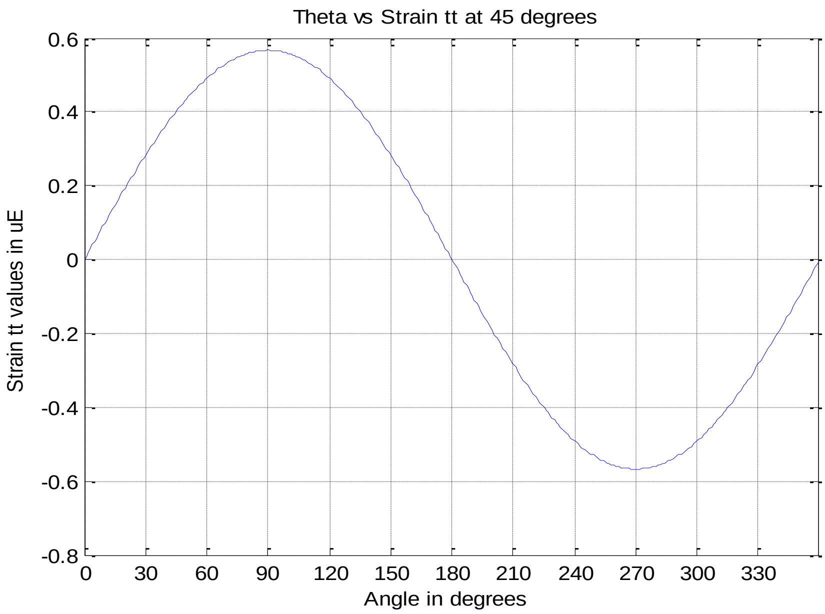


Figure 42 theta vs ε_{tt} in Matlab

Figure 39 to 42 represents the plots obtained for either ε_{tt} or ε_{zz} from Matlab and Mechanica. To obtain an accurate comparison between the two, it is necessary to plot them on a common x axis. The Mechanica plots are plotted with respect to the curve arc length (or the circumference) originating from the conventional horizontal axis.

The Mechanica plots are exported into an excel sheet and the curvature values are converted into corresponding angles. Then, the values are interpolated so that, the x axis ranges between 0 and 359 degrees. The function ‘interp1()’ is used for this purpose.

After interpolation, they are plotted in the same figure and the maximum and minimum errors are indicated henceforth. Matlab plots are in red, Mechanica in blue and Error in green. Please note that the vertical axis of error is towards the right, while the axis of Matlab and Mechanica values are towards left. The resolution of error is increased just for visually showcasing their magnitude.

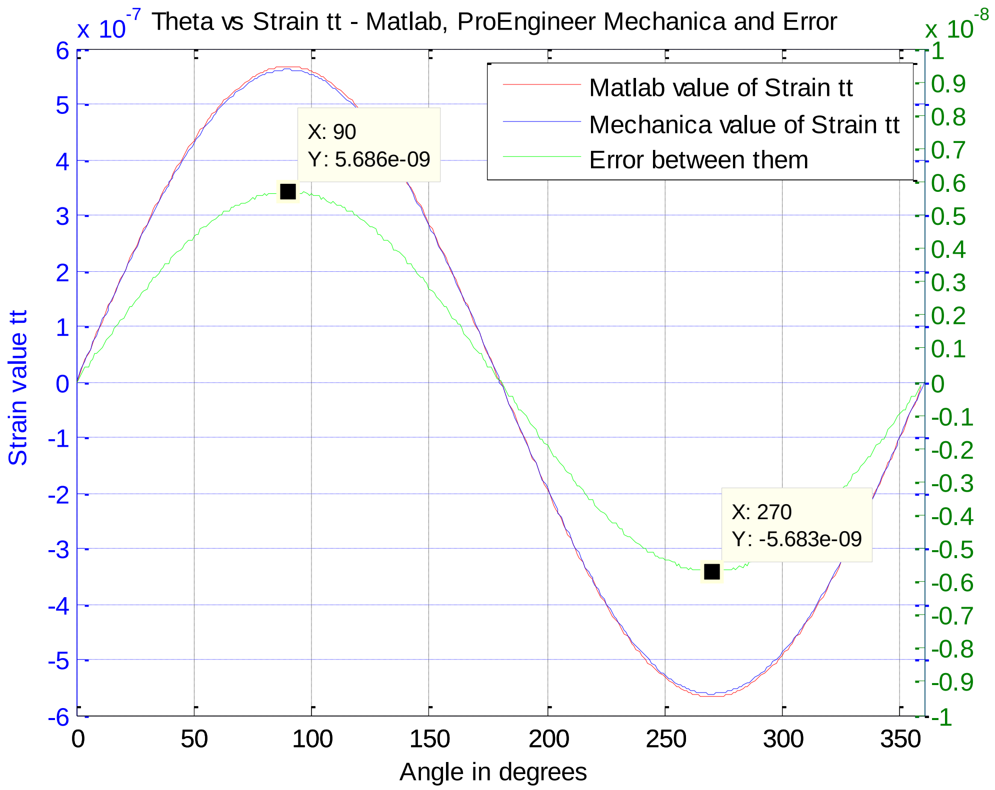


Figure 43 Theta vs ε_{tt} – comparison between Matlab and ProEngineer Mechanica

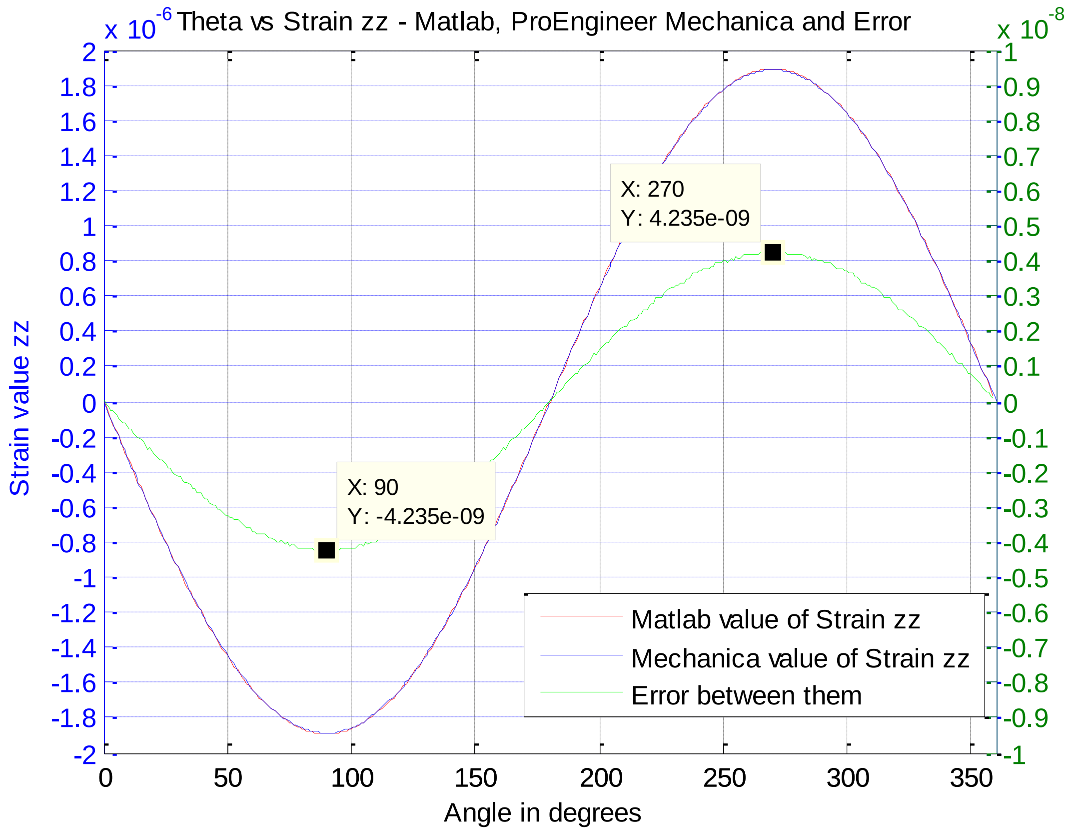


Figure 44 Theta vs ε_{zz} – comparison between Matlab and ProEngineer Mechanica

4.3.2 Concentric hollow shaft –only bending

Here, a concentric shaft is subjected to bending load alone and the plots are compared. The specifications are as follows:

Table 12 Different input values for a concentric shaft subjected to only bending

Parameters	Value
D (in mm)	25
d (in mm)	15
L (in mm)	40
ex (in mm)	
ey (in mm)	
Torque (in Nmm)	
Bending Load (in N)	1
Considered point for verification from load end (in mm)	20

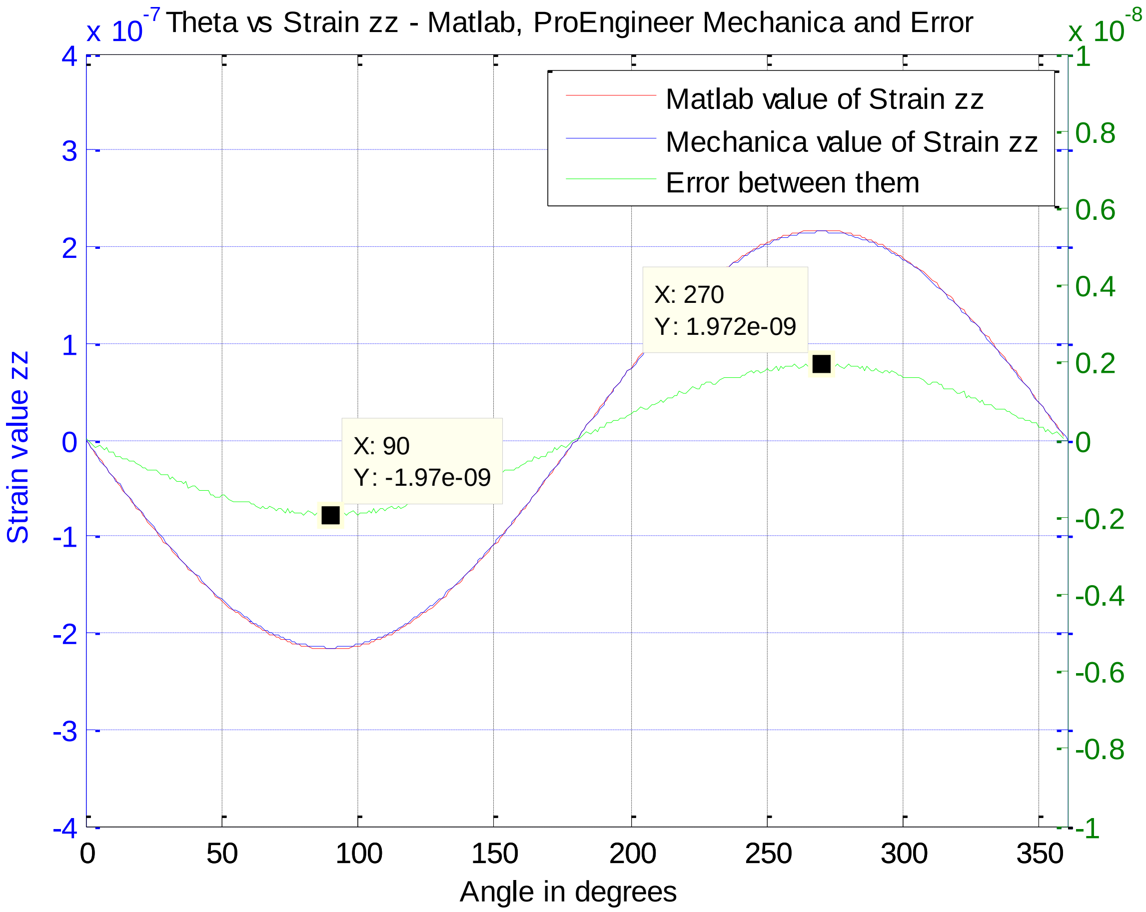


Figure 45 Theta vs ε_{zz} – comparison between Matlab and ProEngineer Mechanica

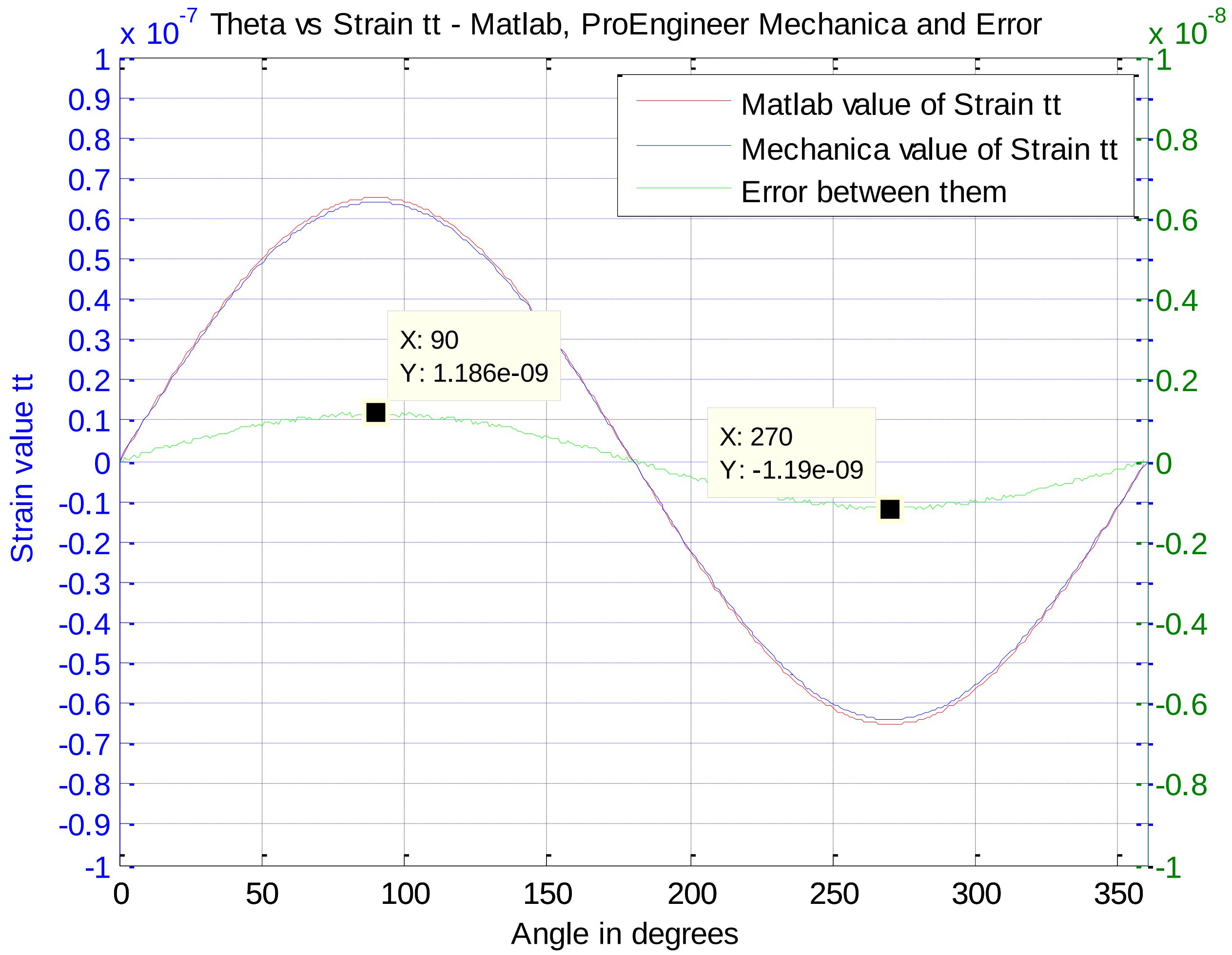


Figure 46 Theta vs ε_{tt} – comparison between Matlab and ProEngineer Mechanica

4.3.3 Eccentric shaft 1– only bending

Here, an eccentric shaft is subjected to bending load alone and the plots are compared. The specifications are as follows:

Table 13 Different input values for an eccentric shaft 1 subjected to only bending

Parameters	Value
D (in mm)	20
d (in mm)	14
L (in mm)	40
ex (in mm)	
ey (in mm)	0.3
Torque (in Nmm)	
Bending Load (in N)	1
Considered point for verification from load end (in mm)	20

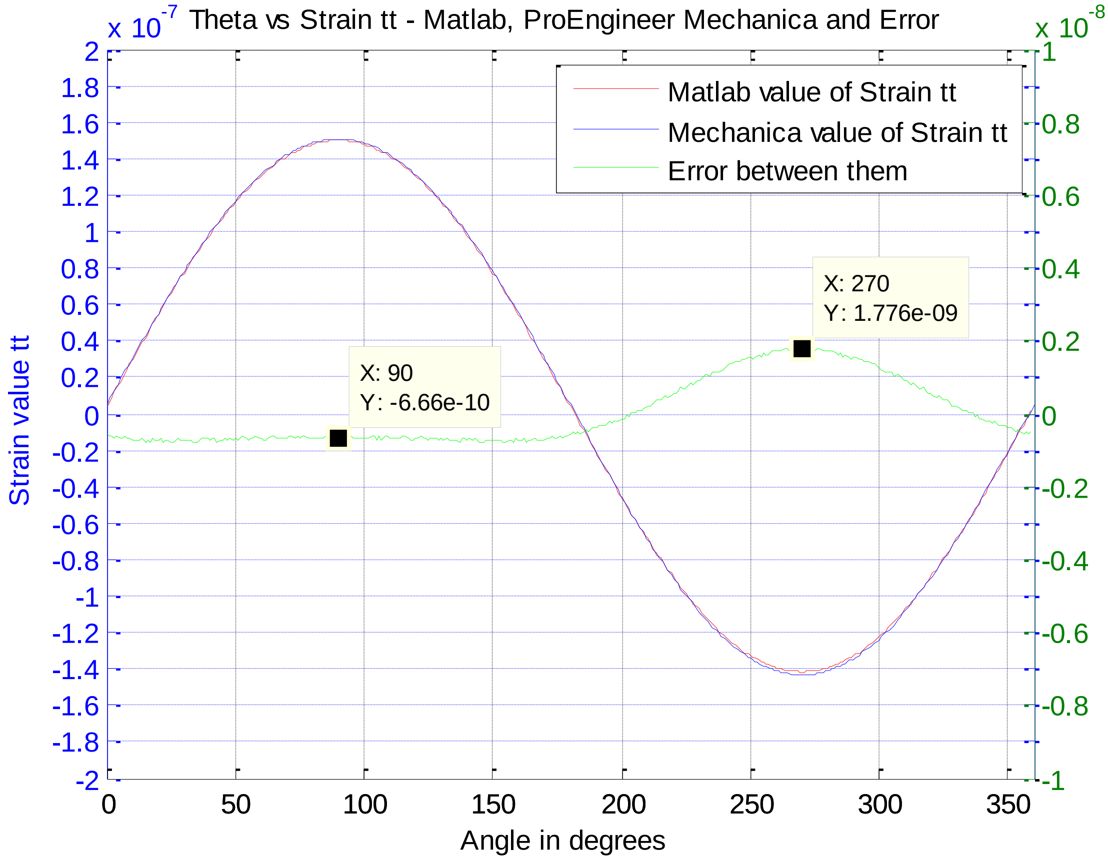


Figure 47 Theta vs ε_{tt} – comparison between Matlab and ProEngineer Mechanica

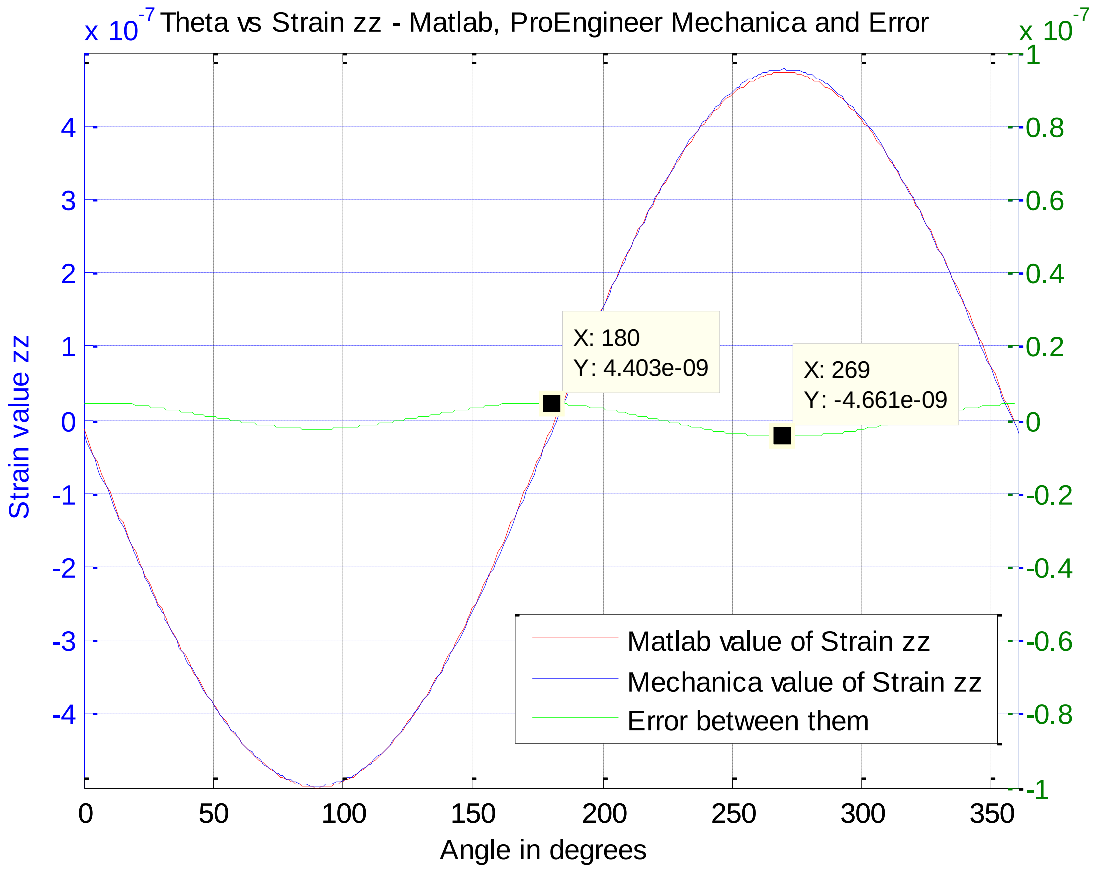


Figure 48 Theta vs ε_{zz} – comparison between Matlab and ProEngineer Mechanica

4.3.4 *Eccentric shaft 2– only bending*

Here, an eccentric shaft is subjected to bending load alone and the plots are compared. The value of eccentricity is increased from the previous case. The specifications are as follows:

Table 14 Different input values for an eccentric shaft 2 subjected to only bending

Parameters	Value
D (in mm)	25
d (in mm)	15
L (in mm)	40
ex (in mm)	0.2
ey (in mm)	0.2
Torque (in Nmm)	
Bending Load (in N)	1
Considered point for verification from load end (in mm)	20

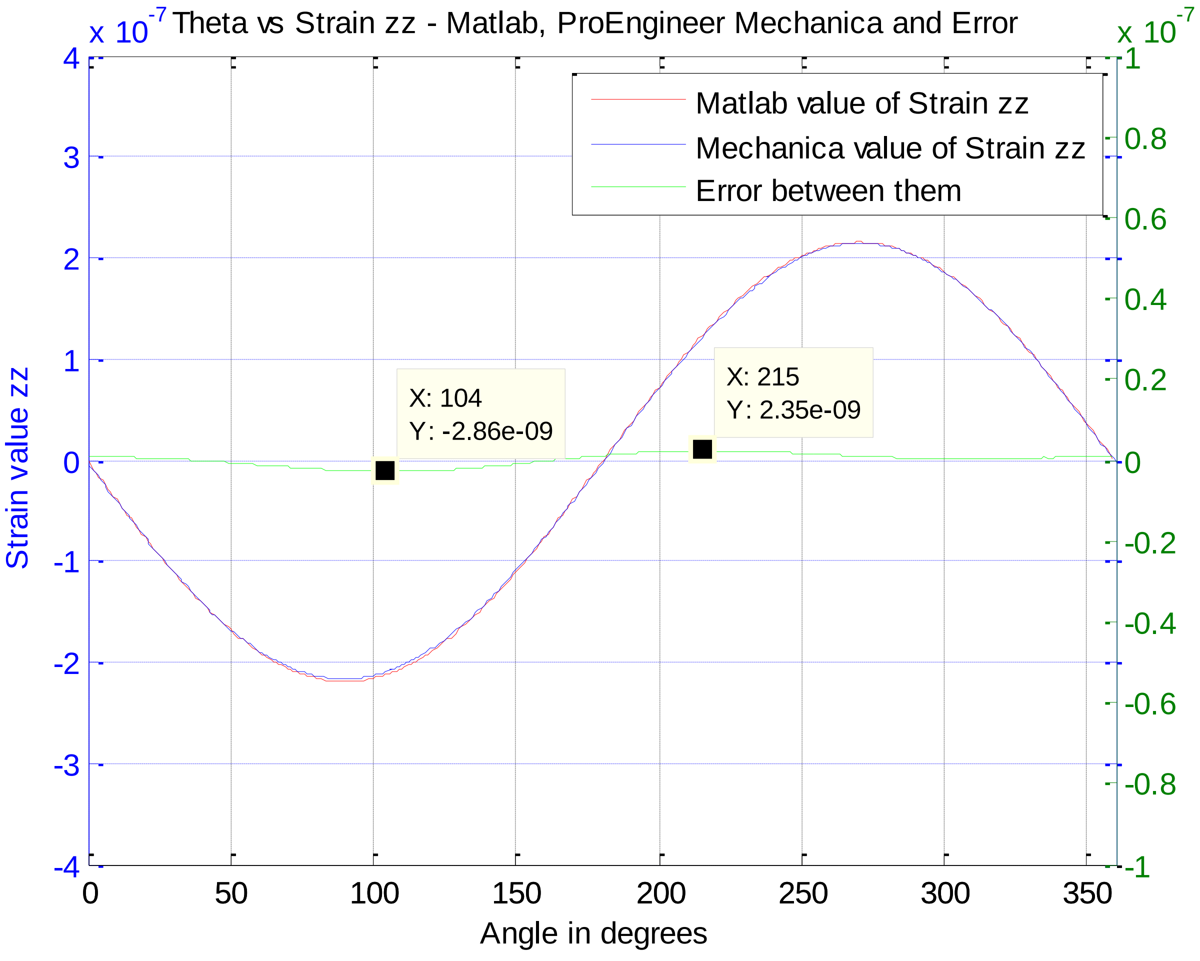


Figure 49 Theta vs ε_{zz} – comparison between Matlab and ProEngineer Mechanica

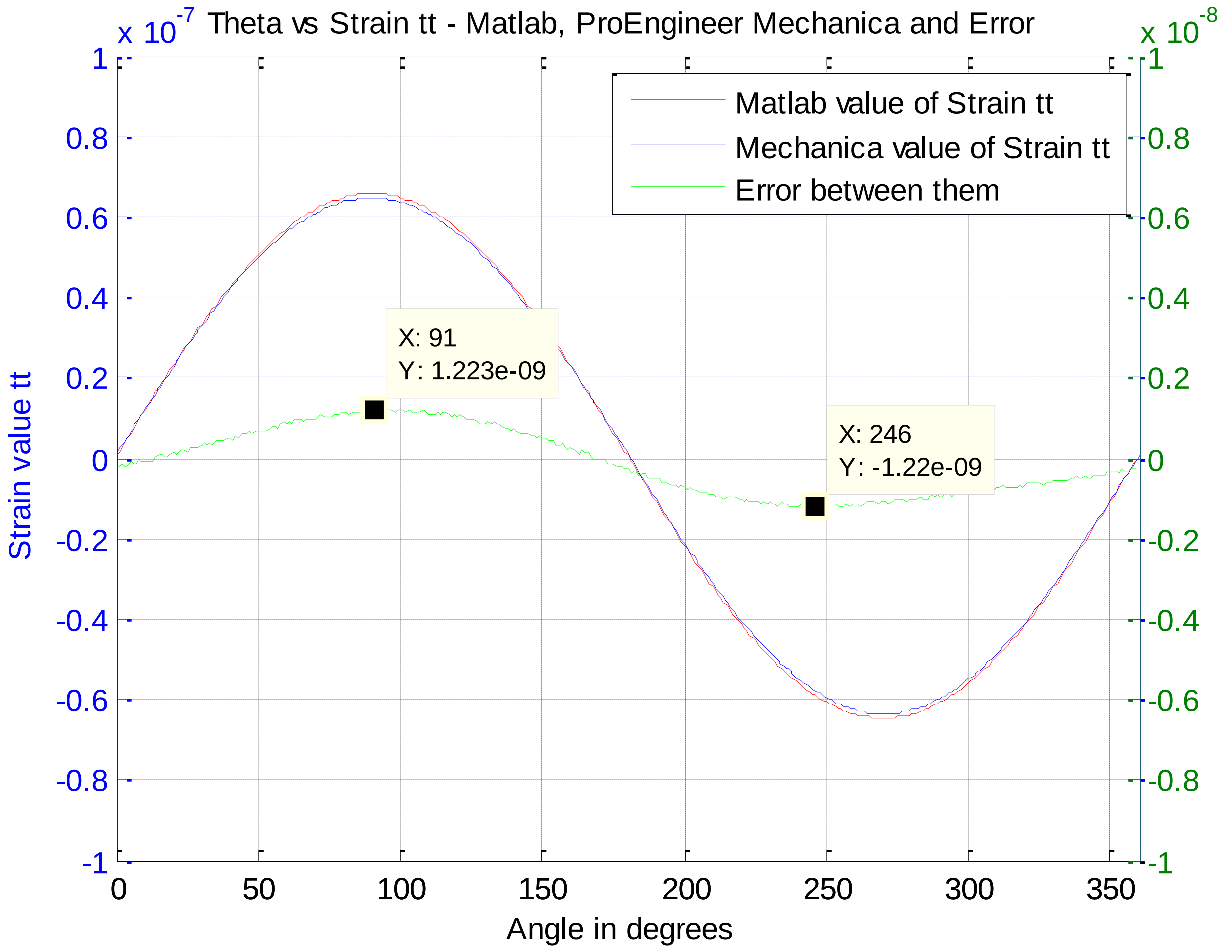


Figure 50 Theta vs ε_{tt} – comparison between Matlab and ProEngineer Mechanica

4.3.5 Solid shaft – torsion and bending

Here, a solid shaft is subjected to torsion and bending loads and the plots are compared. The specifications are as follows:

Table 15 Different input values for a solid shaft subjected to torsion and bending

Parameters	Value
D (in mm)	25
d (in mm)	
L (in mm)	40
ex (in mm)	
ey (in mm)	
Torque (in Nmm)	100
Bending Load (in N)	10
Considered point for verification from load end (in mm)	20

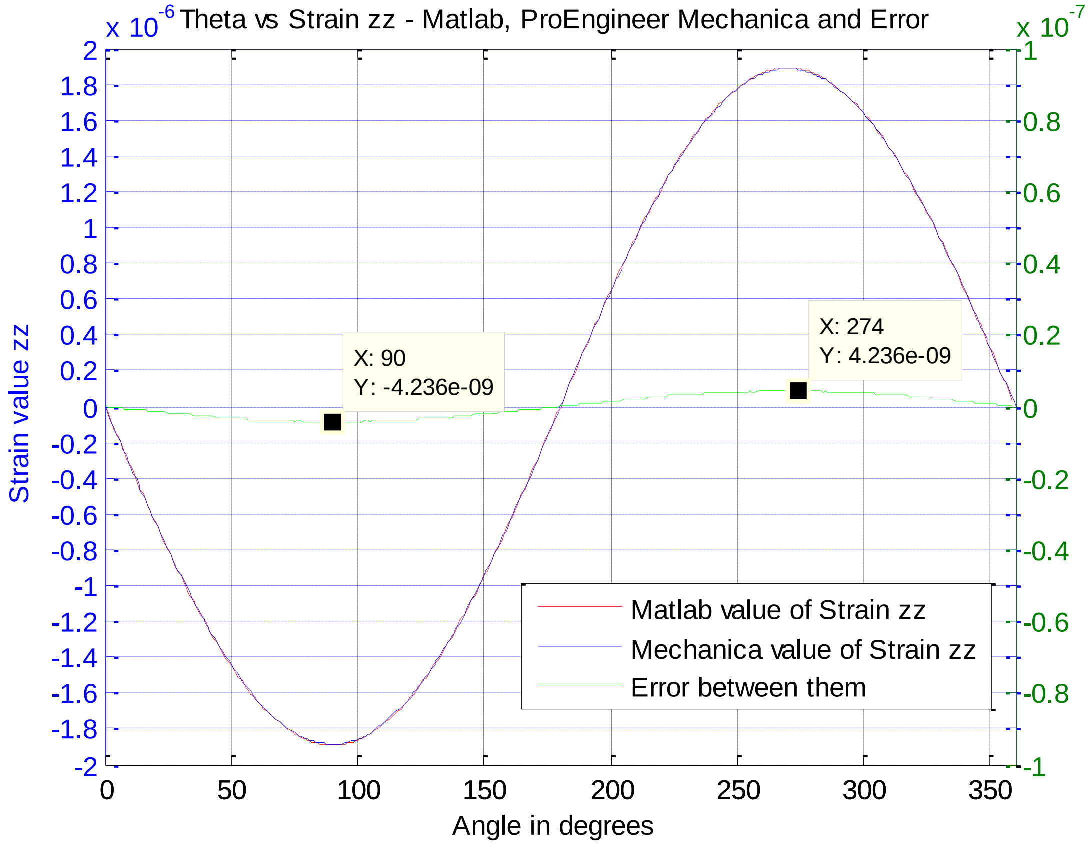


Figure 51 Theta vs ε_{zz} – comparison between Matlab and ProEngineer Mechanica

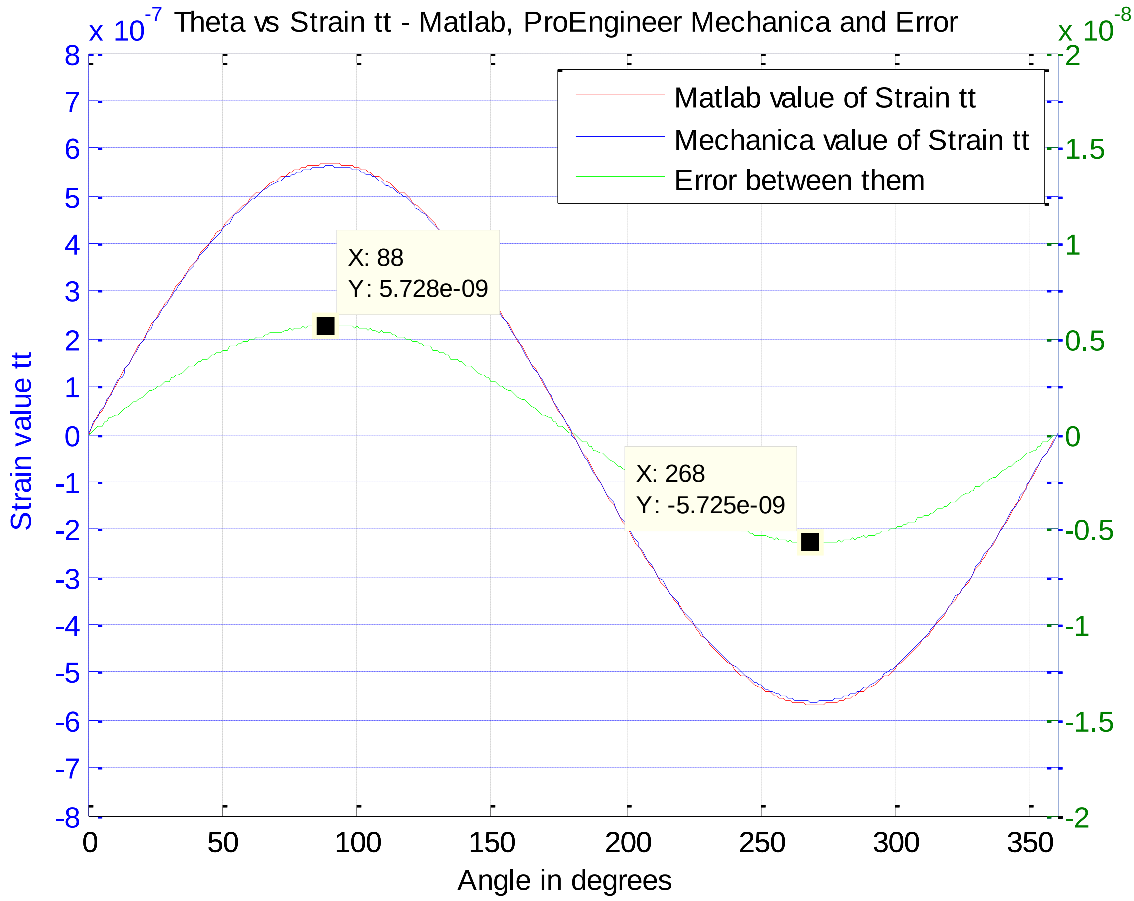


Figure 52 Theta vs ε_{tt} – comparison between Matlab and ProEngineer Mechanica

4.3.6 *Concentric hollow shaft – torsion and bending*

Here, a concentric shaft is subjected to torsion and bending loads and the plots are compared. The specifications are as follows:

Table 16 Different input values for a concentric shaft subjected to torsion and bending

Parameters	Value
D (in mm)	25
d (in mm)	15
L (in mm)	40
ex (in mm)	
ey (in mm)	
Torque (in Nmm)	100
Bending Load (in N)	1
Considered point for verification from load end (in mm)	20

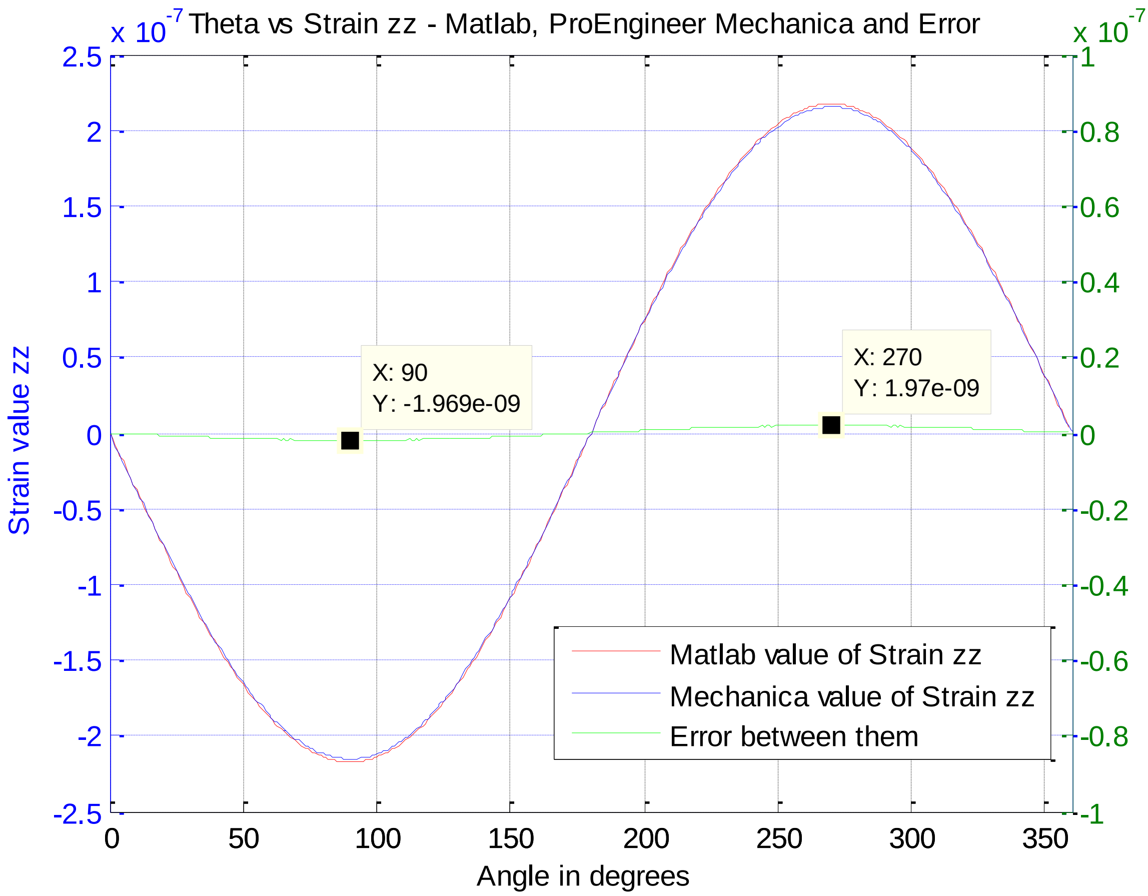


Figure 53 Theta vs ε_{zz} – comparison between Matlab and ProEngineer Mechanica

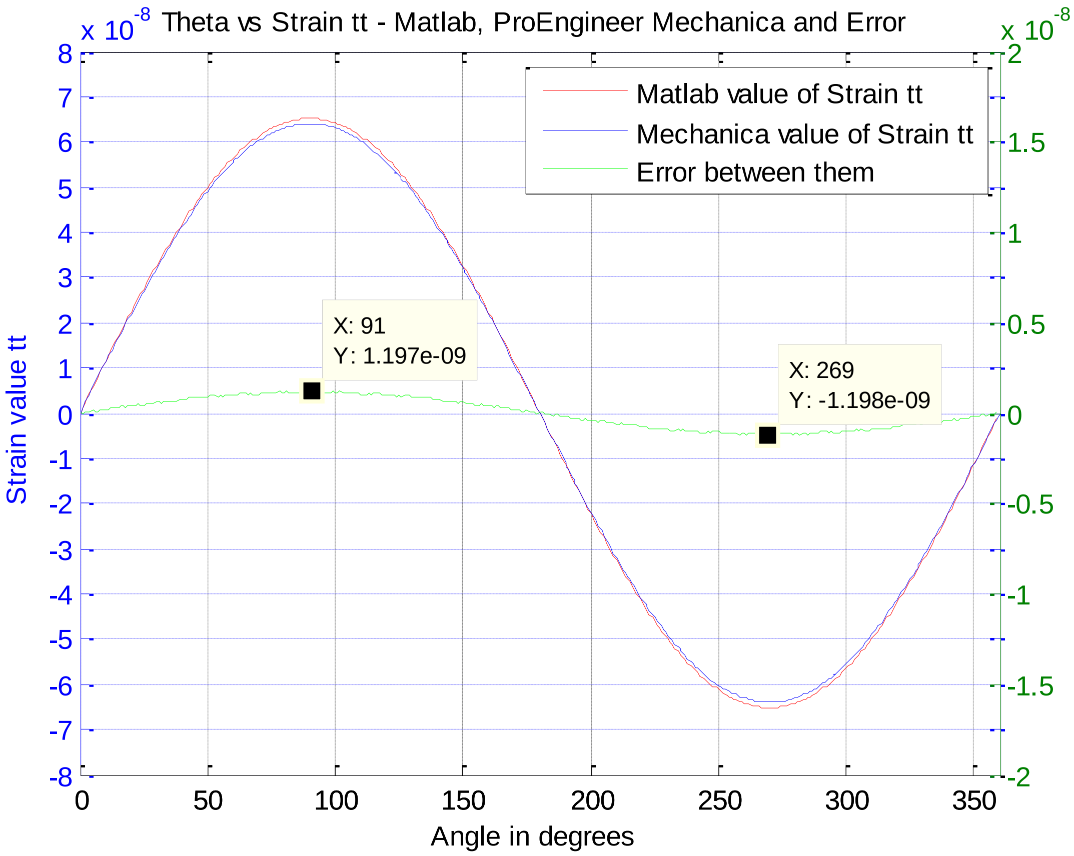


Figure 54 Theta vs ε_{tt} – comparison between Matlab and ProEngineer Mechanica

4.3.7 Eccentric shaft 1– torsion and bending

An eccentric shaft is subjected to torsion and bending loads and the plots are compared. The specifications are as follows:

Table 17 Different input values for an eccentric shaft 1 subjected to torsion and bending

Parameters	**Value**
D (in mm)	20
d (in mm)	14
L (in mm)	40
ex (in mm)	
ey (in mm)	0.3
Torque (in Nmm)	100
Bending Load (in N)	1
Considered point for verification from load end (in mm)	20

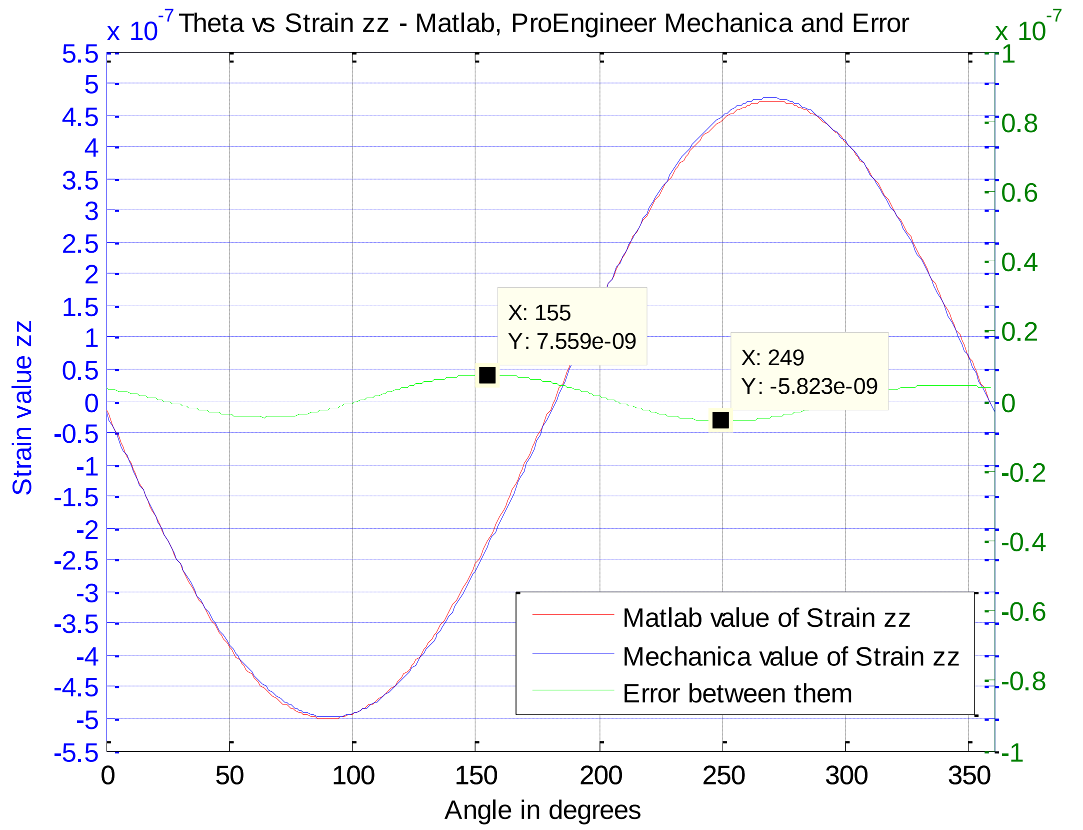


Figure 55 Theta vs ε_{zz} – comparison between Matlab and ProEngineer Mechanica

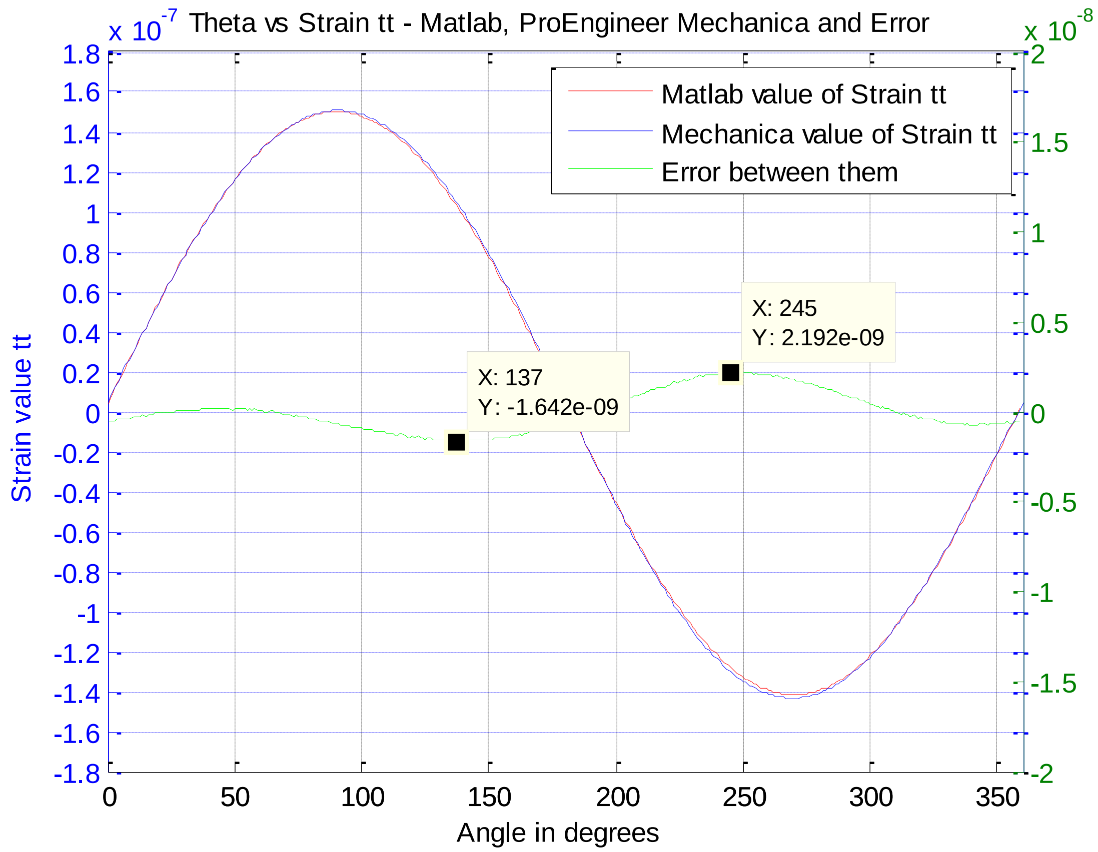


Figure 56 Theta vs ε_{tt} – comparison between Matlab and ProEngineer Mechanica

4.3.8 Eccentric shaft 2– torsion & bending

An eccentric shaft is subjected to torsion and bending loads and the plots are compared. The value of eccentricity is increased from the previous case. The specifications are as follows:

Table 18 Different input values for an eccentric shaft 2 subjected to torsion and bending

Parameters	Value
D (in mm)	25
d (in mm)	15
L (in mm)	40
ex (in mm)	0.2
ey (in mm)	0.2
Torque (in Nmm)	100
Bending Load (in N)	1
Considered point for verification from load end (in mm)	20

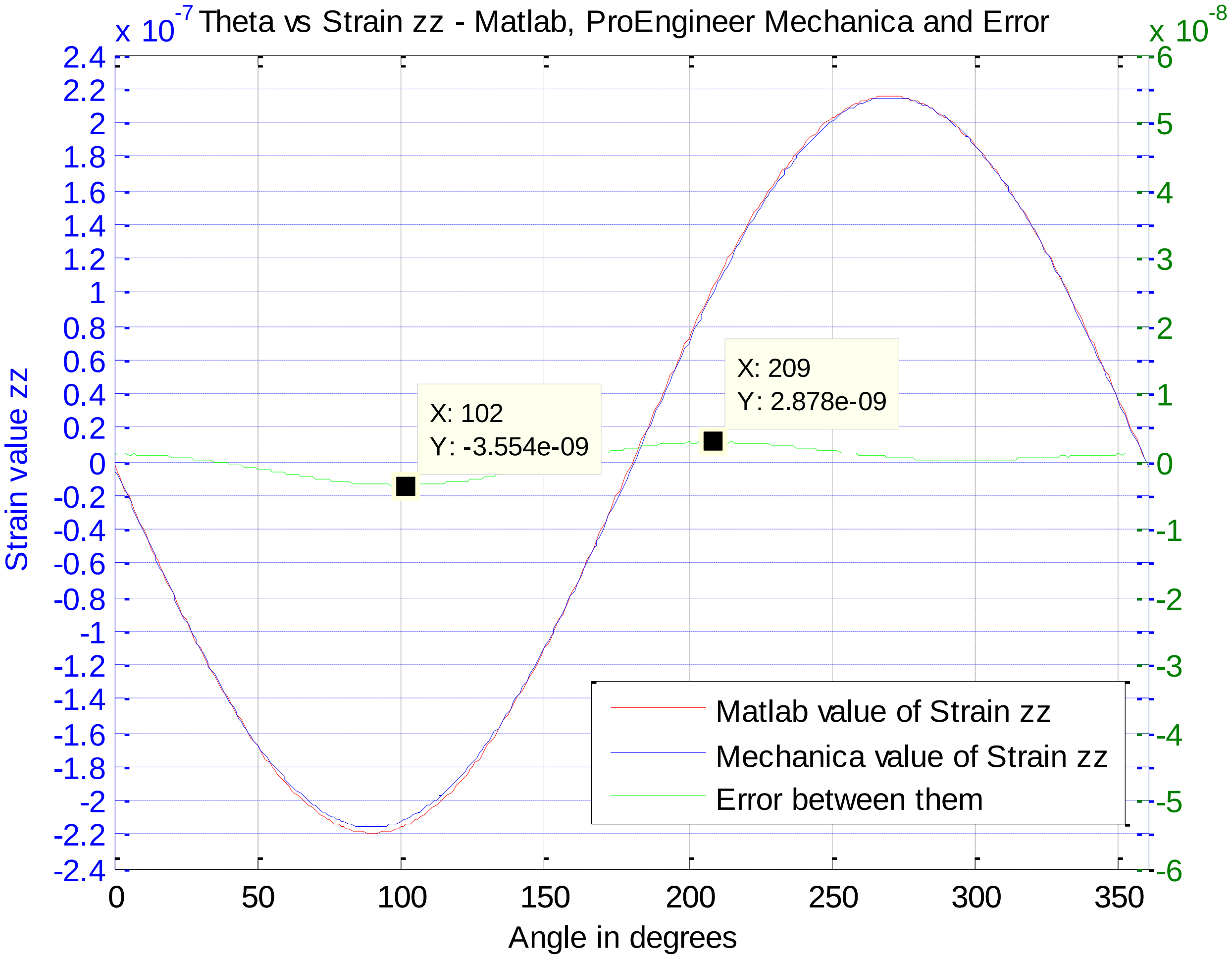


Figure 57 Theta vs ε_{zz} – comparison between Matlab and ProEngineer Mechanica

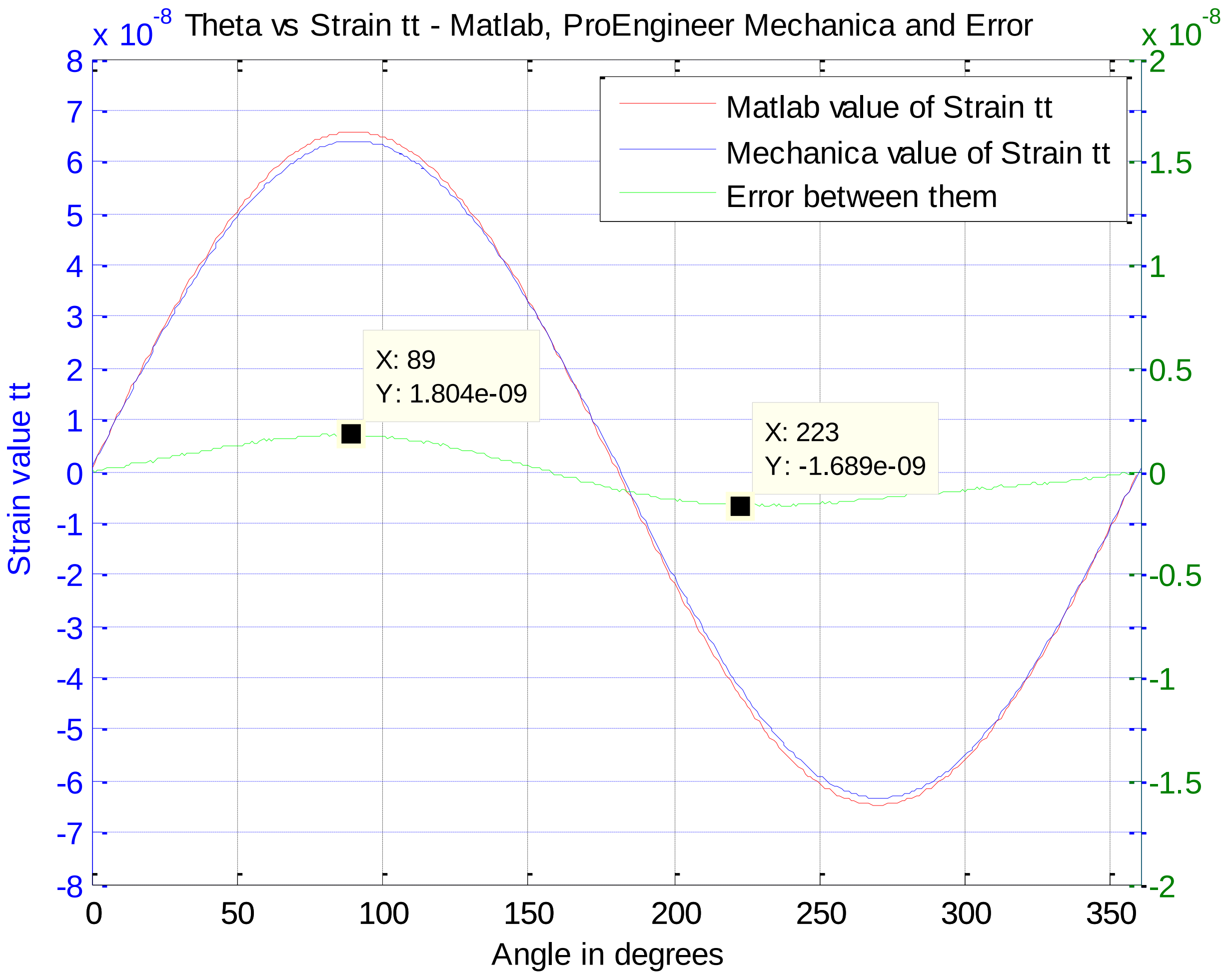


Figure 58 Theta vs ε_{tt} – comparison between Matlab and ProEngineer Mechanica

4.3.9 Solid shaft – only axial load

A solid shaft is subjected to axial load alone and the plots are compared. The specifications are as follows:

Table 19 Different input values for an solid shaft subjected to axial load

Parameters	Value
D (in mm)	20
d (in mm)	
L (in mm)	40
ex (in mm)	
ey (in mm)	
Torque (in Nmm)	
Bending Load (in N)	
Axial load (in N)	5
Considered point for verification from load end (in mm)	20

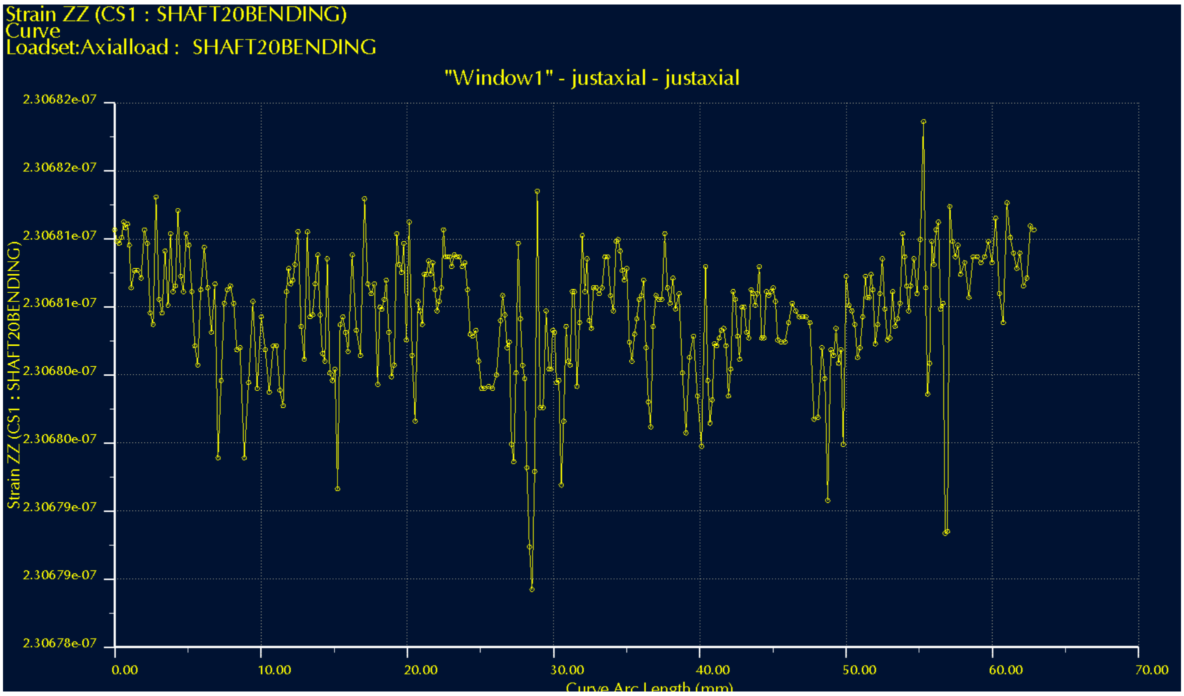


Figure 59 Theta vs ε_{zz} in ProEngineer Mechanica

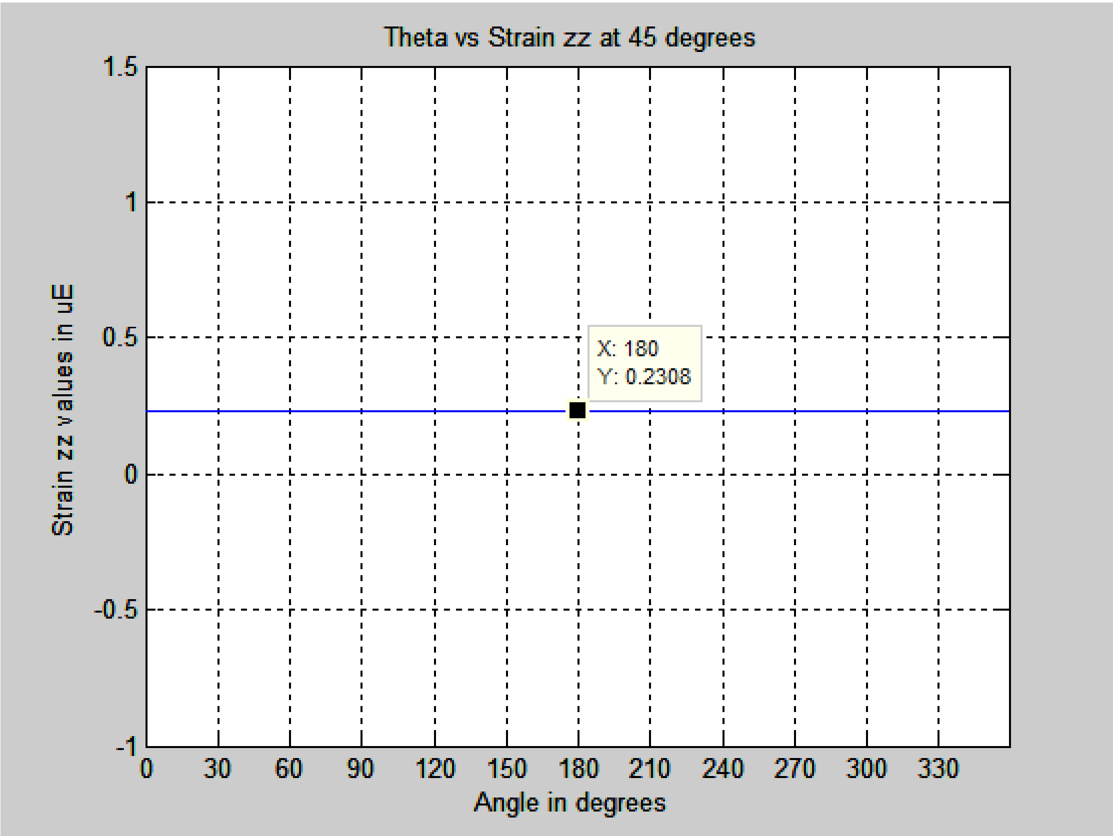


Figure 60 Theta vs ε_{zz} in Matlab

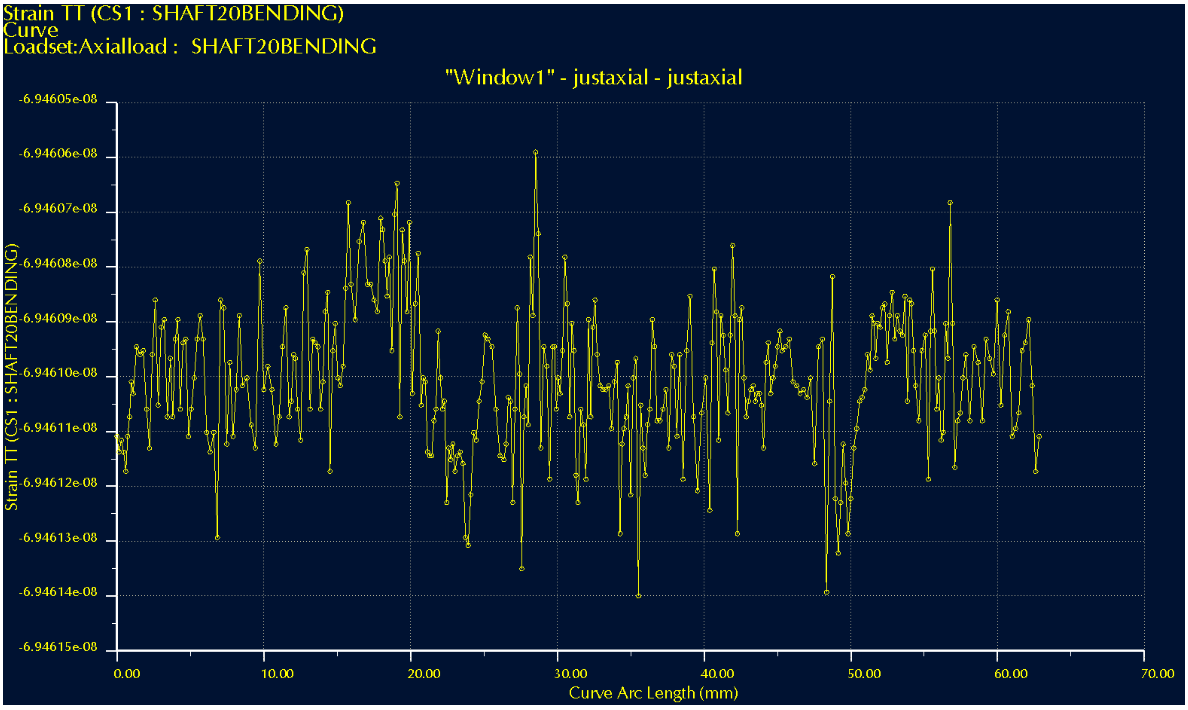


Figure 61 Theta vs ε_{tt} in ProEngineer Mechanica

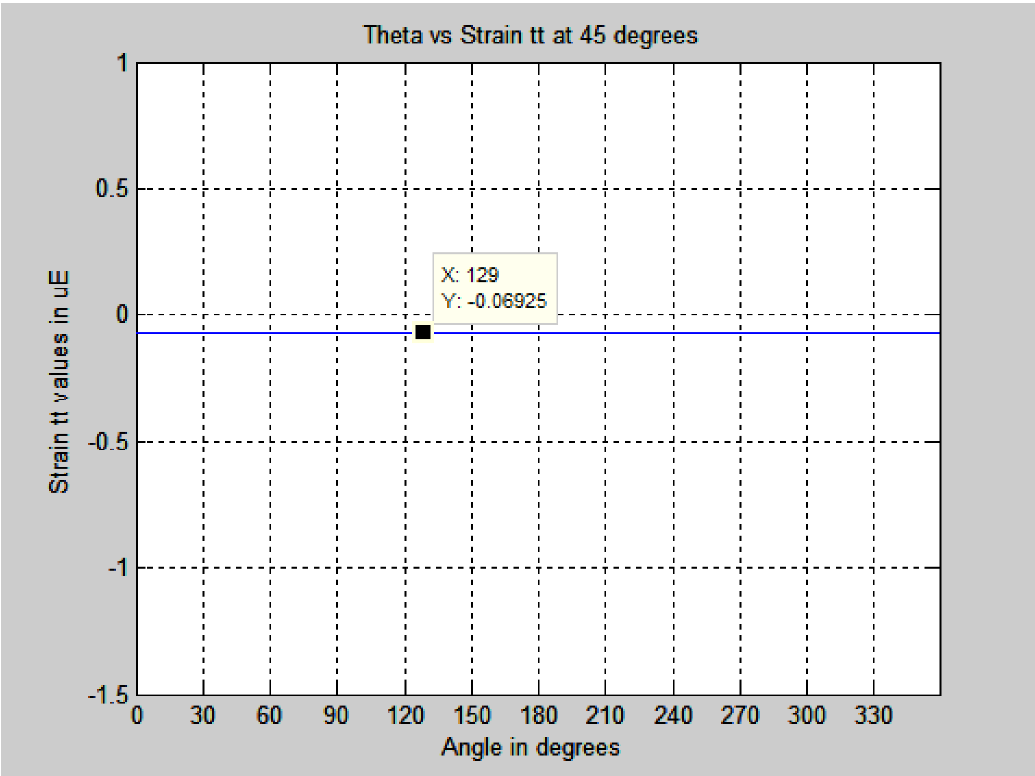


Figure 62 Theta vs ε_{tt} in Matlab

Here, it can be seen that for just an axial load, the strain zz is not constant. However, theory suggests that it should be the same (the stress being the load divided by area). These some irregularities that occur in Mechanica can be attributed to the effect of constraints (numerical elements and element size).

The irregularities are around 1% of the theoretical optimal value and can be regarded as normal errors by the FEA approach mode. The percentage of convergence was set at 1% and that is a possible explanation for the difference.

4.3.10 Solid shaft – torsion, bending and axial load

A solid shaft is subjected to bending, axial and torsional loads and the plots are compared. The specifications are as follows:

Table 20 Different input values for a solid shaft subjected to bending, torsion and axial load

Parameters	Value
D (in mm)	12
d (in mm)	
L (in mm)	40
ex (in mm)	
ey (in mm)	
Torque (in Nmm)	2100
Bending Load (in N)	2.5
Axial load (in N)	5
Considered point for verification from load end (in mm)	20

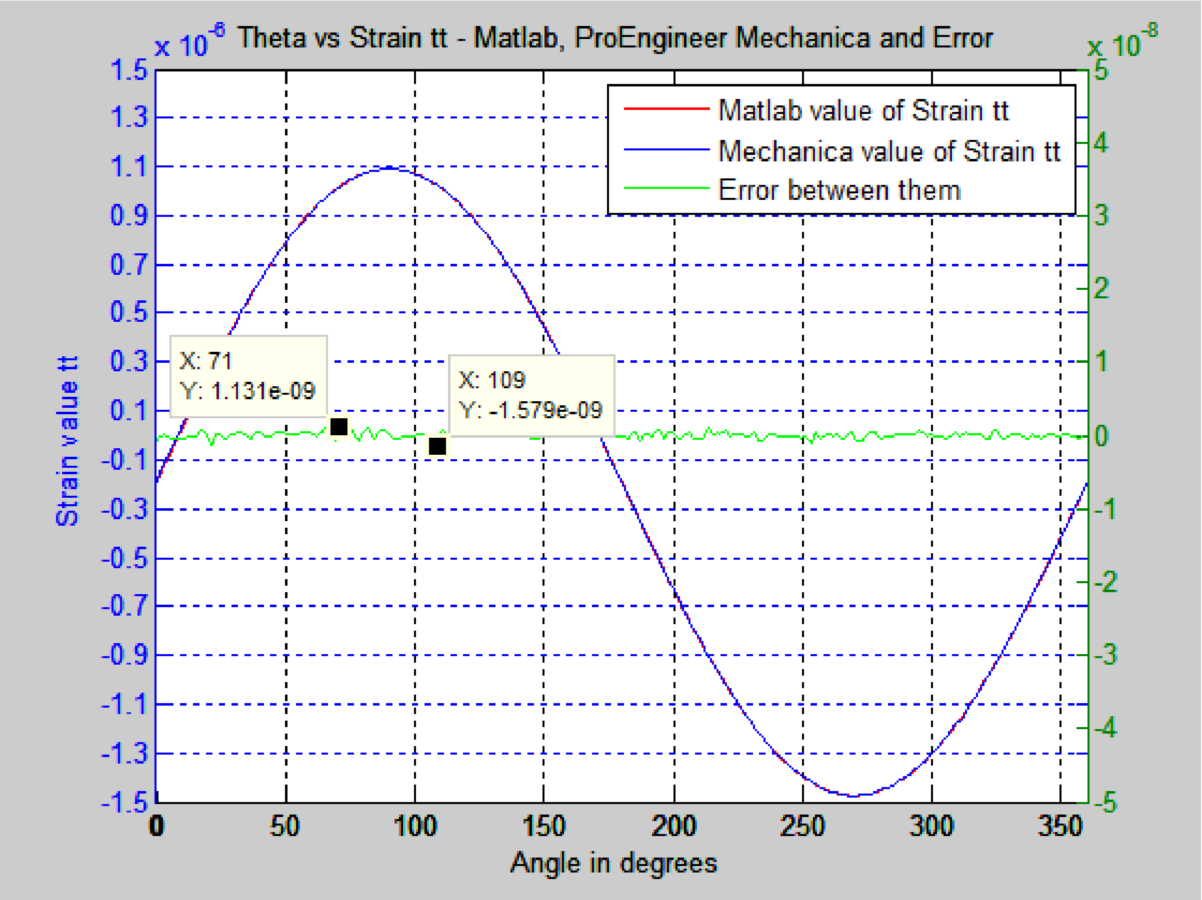


Figure 63 Theta vs ε_{tt} – comparison between Matlab and ProEngineer Mechanica

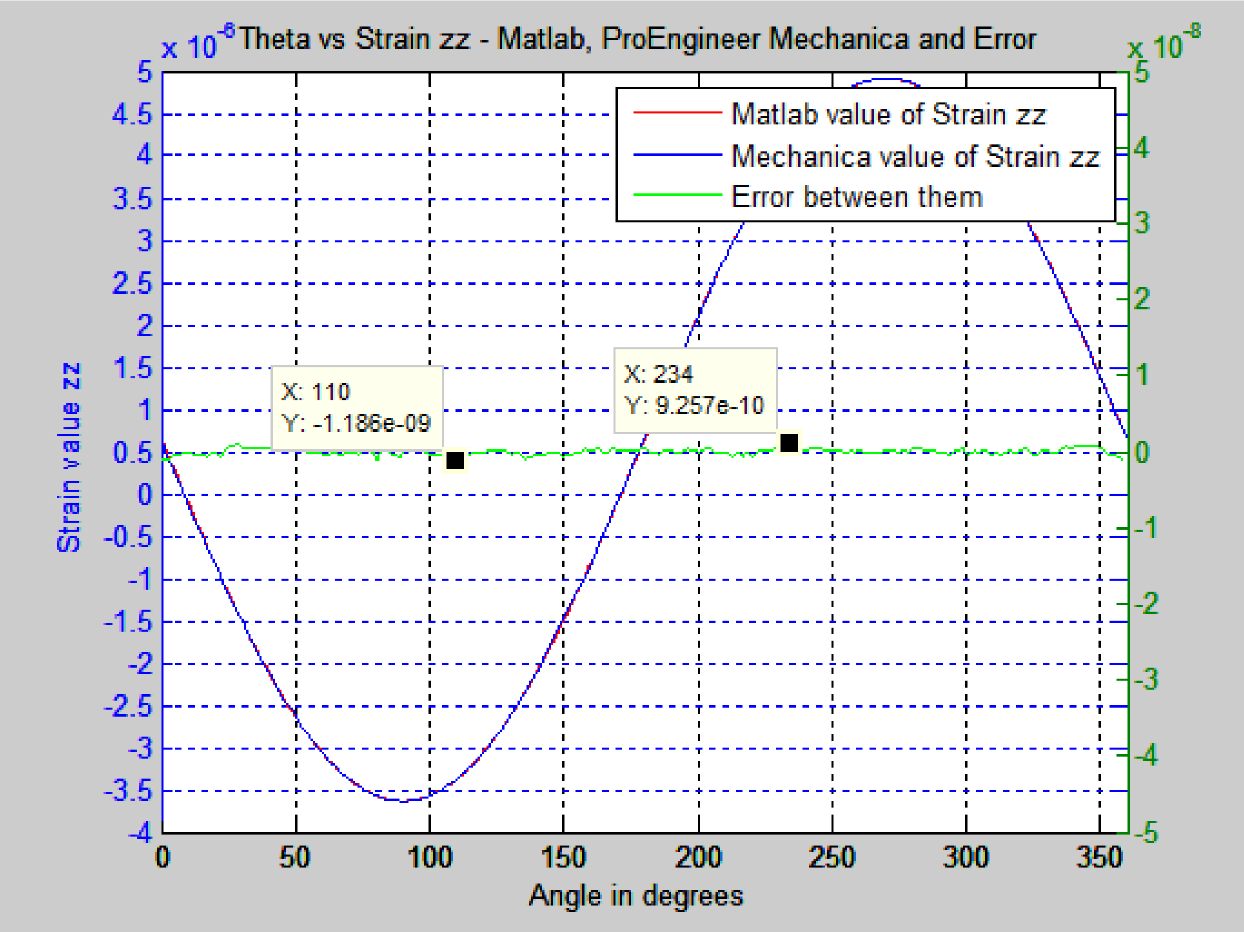


Figure 64 Theta vs ε_{zz} – comparison between Matlab and ProEngineer Mechanica

4.3.11 Stress plane summary

So far, the verification of the following parameters are carried out for different cases and the values are found to be the same in FEA and Matlab.

- Torsional Shear stress
- ε_{tt}
- ε_{zz}

In all the case, the ε_{zz} and ε_{tt} are similar, with a small difference in the range of 1%.

After a long discussion and relevant literature study, it was found out the strain gauges will be insensitive to shear stress (subsequently strain) caused by bending, since the strain gauges which will be placed at 45 degrees (along the curved surfaces of the shaft) are in a different plane to that of the plane where the shear stress because of bending acts. However, that is not the case for shear stresses caused by torsion. At 45 degrees, the shear stress due to Torsion causes compressive and tensile stresses (and subsequently strain) and that is the main reason for keeping them aligned at 45 degrees. In short, when a strain gauge is aligned at 45 degrees, it can sense the maximum and minimum principal strain attributable to torsion. If a bending load is present, then along with the torsional principal strain, it records the normal bending strain.

The normal bending strains (in z direction as well as the transverse strain) are verified using FEM and the values are the ones that have been concluded in the previous section. The torsional shear stress is also verified using FEM. Since, there are only these three parameters, it is safe to conclude that up and until this point, the verification is carried out and the values are the same.

The next step is to compute the strain that will be sensed by the strain gauge. It is to be noted that strain at 45 degrees was not able to be graphically interpreted in ProE Mechanica and that is the reason for working on the stress transformation.

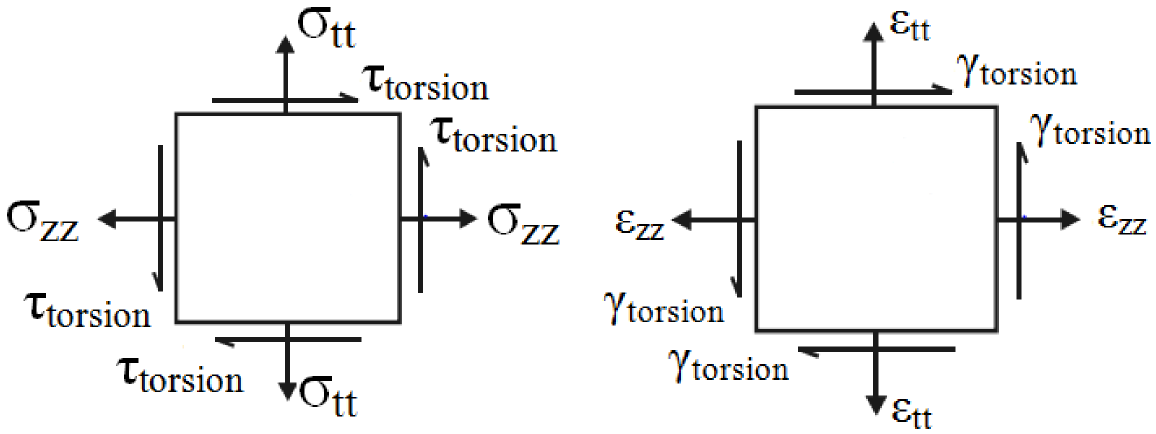


Figure 65 Stress plane system

$\tau_{torsion}$ is verified for different cases. $\gamma_{torsion}$ is the ratio of $\tau_{torsion}$ to Shear Modulus. ε_{zz} and ε_{tt} were also verified. So, having verified ε_{zz}, ε_{tt} and $\gamma_{torsion}$, the next step is to calculate the strain at 45 degrees and also at other angles. Using strain transformation, strain at any angle can be found out using the formula,

$$\varepsilon_{angle} = \left(\frac{\varepsilon_{zz}+\varepsilon_{tt}}{2}\right) + \left(\frac{\varepsilon_{zz}-\varepsilon_{tt}}{2}\right)\cos 2\theta + (\gamma_{xy}/2)\sin 2\theta \qquad (73)$$

Using the above expression, strain at any angle orientation (the strain that a strain gauge will indicate when oriented at that particular angle) can be computed.

In the results chapter the results that are obtained with the methods described in the method chapter are compiled, and analyzed and compared with the existing knowledge and theory presented in the frame of reference chapter.

5.1. Strain plot

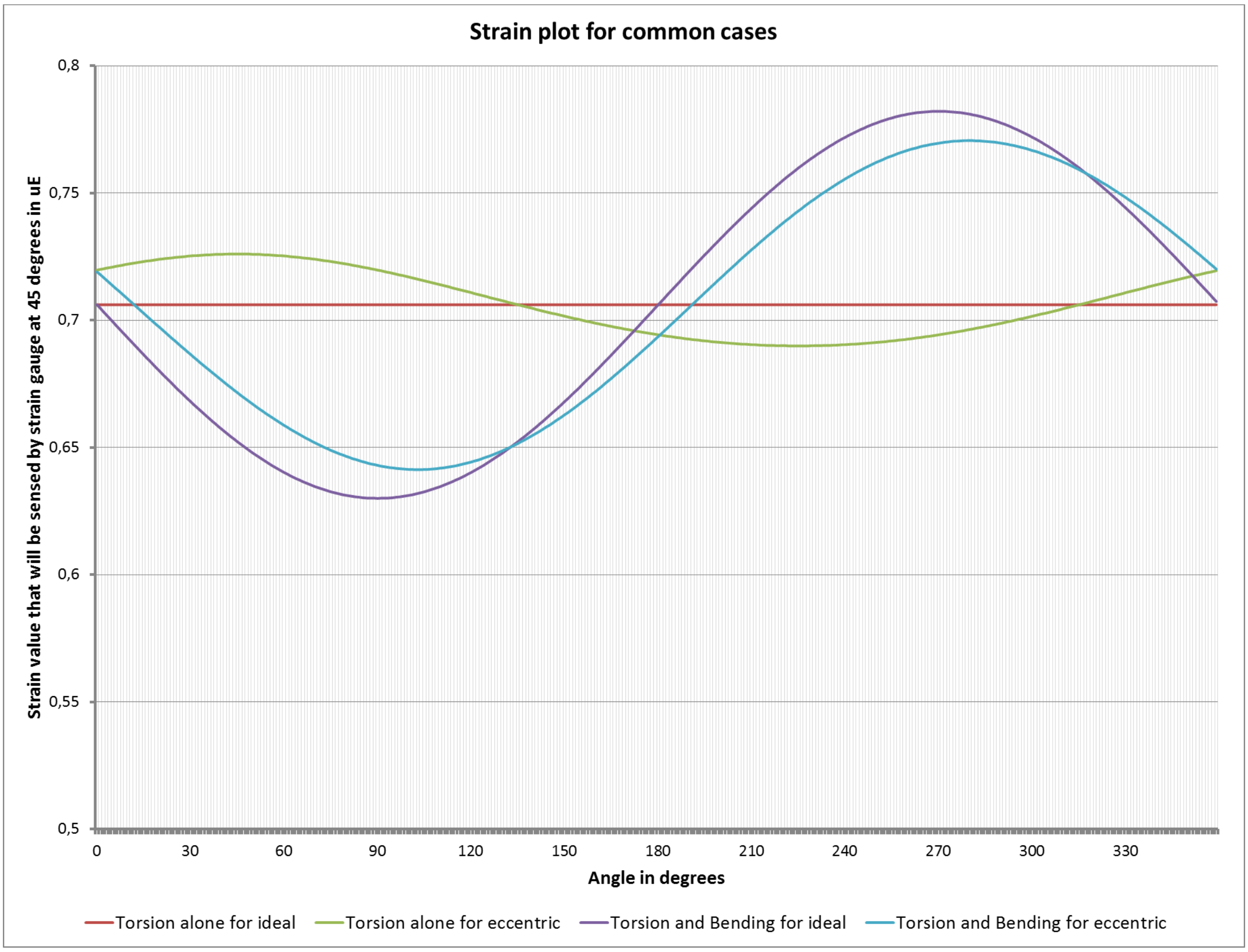


Figure 66 Strain plot for common cases

The above figure shows the resultant strain along 45 degrees for different cases. For an ideal shaft subjected to torsion alone, then the strain plot will be a straight line. If the shaft is slightly eccentric, then the strain plot will be sinusoidal in nature. However, if the shaft is subjected to torsion and bending load, then the strain plots will also be sinusoidal, with the magnitude depending upon the difference in the torsion and bending loads. The above figure is just for illustration purposes, primarily to show the appearance of the profile.

5.2. Strain gauge orientation angle

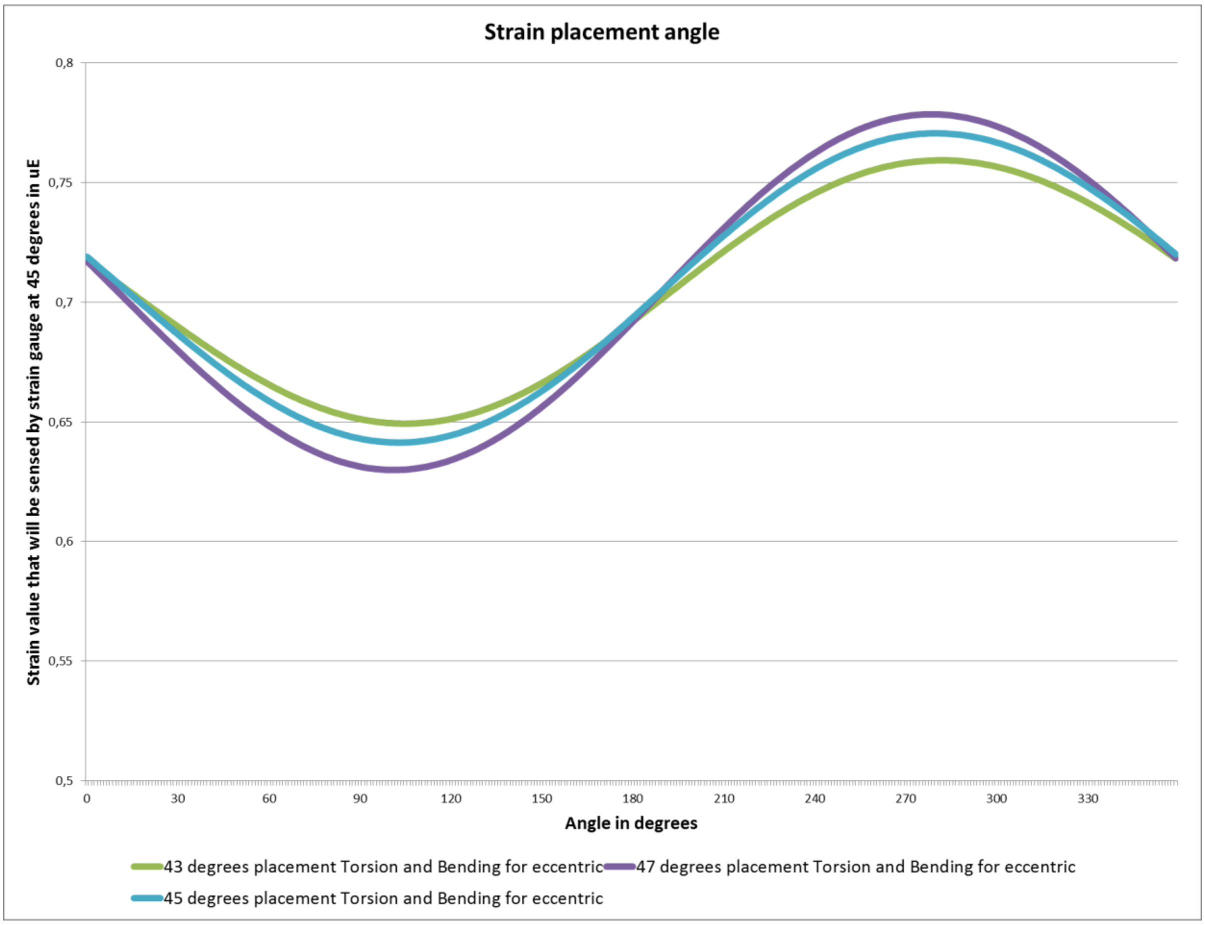


Figure 67 Influence of strain gauge placement

Strain gauges are intended to be placed at 45 degrees. However, because of some misalignments, chances are that they might be placed at some angle other than 45 degrees. Figure 67 shows the cases, when the strain gauge is aligned at 43, 45 and 47 degrees.

5.3. Strain plots for an existing transducer

The specifications of the Aluminium transducer that is used in Atlas Copco are as follows:

Table 21 Specifications of aluminium transducer

Parameters	Value
D (in mm)	12
d (in mm)	11.2
L (in mm)	30
ex (in mm)	0
ey (in mm)	0.02
Torque (in Nmm)	2100
Bending Load (in N)	2.5
Axial load (in N)	5

Following are the different plots that the Matlab code gives as output

5.3.1 *Shear stress due to torsion alone*

Here, the shear stress caused by torsion alone is plotted against the different angles.

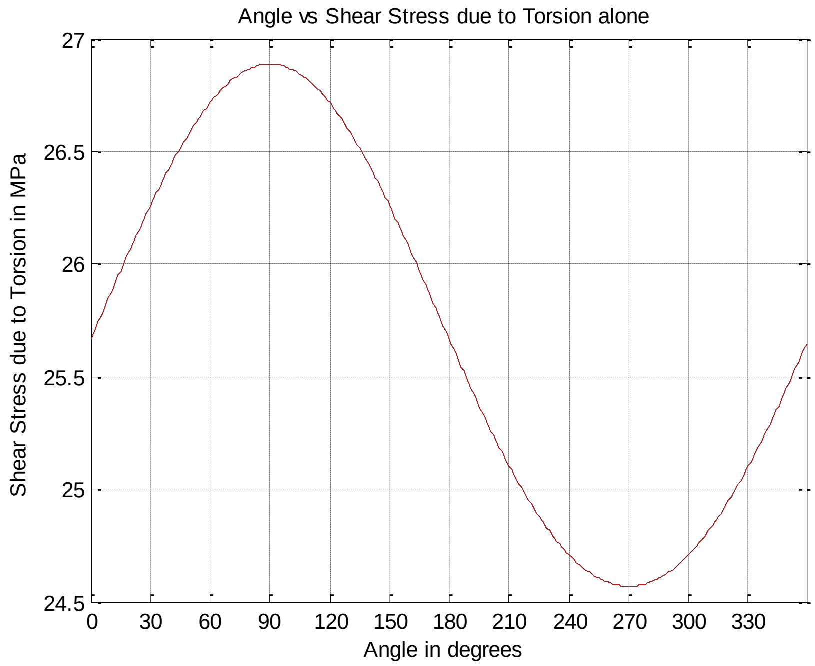


Figure 68 Shear stress due to torsion alone

5.3.2 *ε_{zz}*

Here, the strain at z plane towards z direction is plotted against the different angles for different lengths. The entire array is plotted against theta values. Each line corresponds to a particular length from the load end. The resolution was set at 1 mm length and 1 degree in angles.

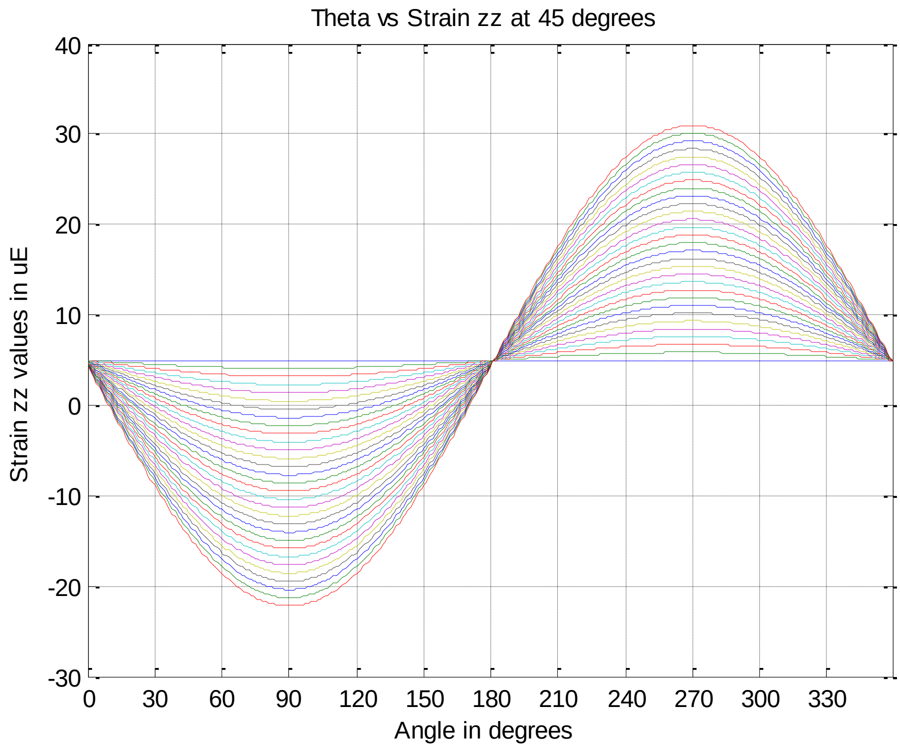


Figure 69 theta vs ε_{zz} at 45 degrees

5.3.3 ε_{tt}

Here, the strain at phi plane towards phi direction is plotted against the different angles for different lengths. The entire array is plotted against theta values.

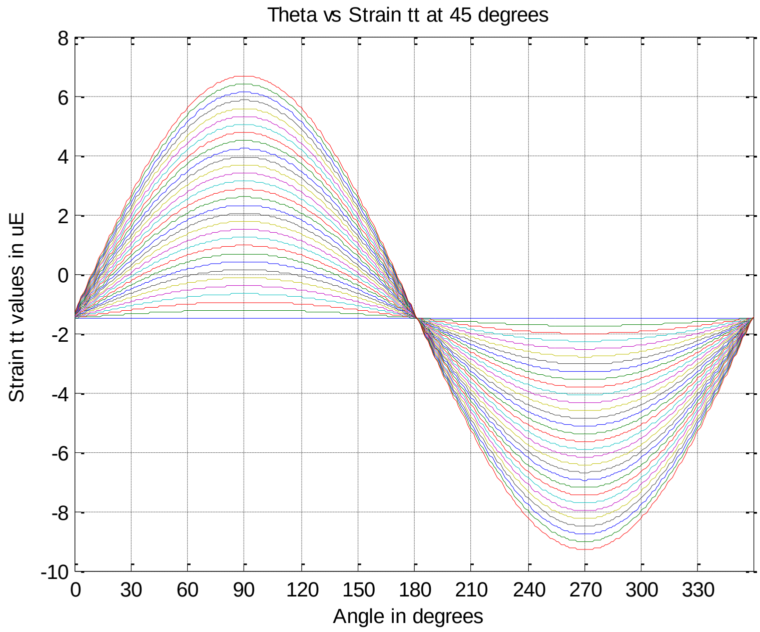


Figure 70 theta vs ε_{tt} at 45 degrees

5.3.4 Strain at user angle 1 (45 degrees)

Here, the strain at the angle which the user specifies is plotted against the different angles for different lengths. The entire array is plotted against theta values. It is generally common to choose at 45 degrees.

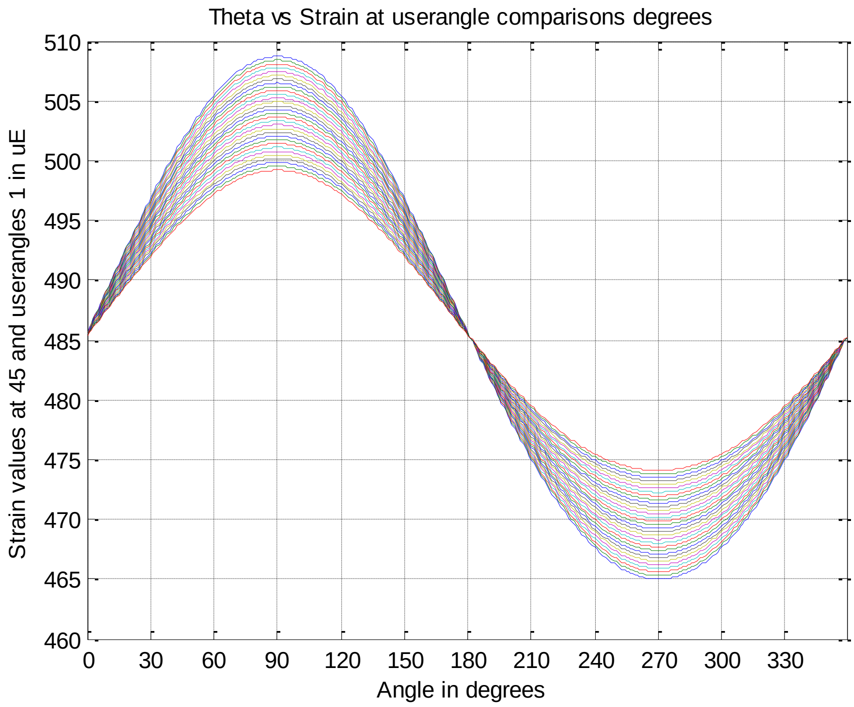


Figure 71 theta vs strain at user angle 1

5.3.5 *Strain at user angle 2 (-45 degrees)*

Here, the strain at the angle which the user specifies is plotted against the different angles for different lengths. The entire array is plotted against theta values. It is generally common to choose at -45 degrees.

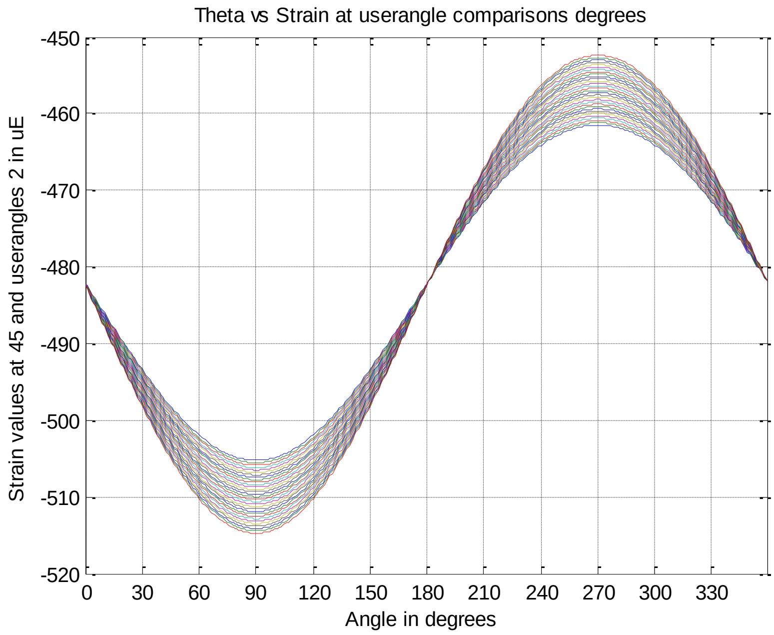


Figure 72 theta vs strain at user angle 2

5.4. Sensitivity for different cases

Strain gauge transducers usually employ four strain gauge elements electrically connected to form a Wheatstone bridge circuit. The sensor can occupy 1/2/3/4 arms of the bridge, depending on the application. The change in resistance is a parameter which is directly a resultant from strain. Thus, sensitivity can also be written as the following (Hoffman, 1989),

$$Sensitivity = \frac{Gauge\ Factor}{4 \times 1000} * (\varepsilon 1 - \varepsilon 2 + \varepsilon 3 - \varepsilon 4) \tag{74}$$

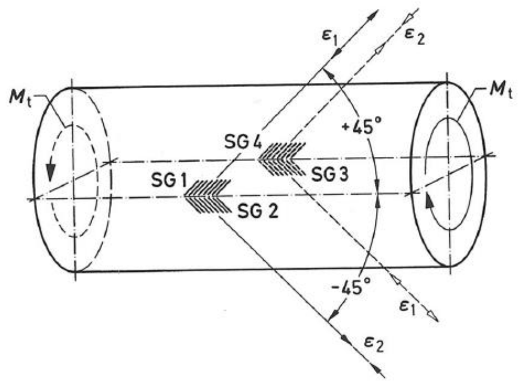


Figure 73 Strain gauge sensitivity

Here, ε_1 corresponds to a strain gauge, the primary objective being to measure the elongation along that particular direction (which is 45 degrees from the horizontal axis). ε_2 corresponds to the compressive strain along the negative 45 degrees sensed by strain gauge 2. Similarly, ε_3 and ε_4 are strain sensed by strain gauges 3 and 4, which are compressive and tensile strain

respectively. Since ε_2 and ε_4 measure strain compressive strain (along the negative direction), a negative sign is used in the expression. In general, the above expression gives an idea about the sensitivity for a strain gauge, which is a relative term used to show how sensitive a strain gauge is to the measurements.
Now, four regions are considered and their coordinates are mentioned below in table 23. Then sensitivity for Case 1-9 are found out in order to see how the eccentricity affect sensitivity.

Table 22 Coordinates of the four strain regions

Region	Strain Gauge 1	Strain Gauge 2	Strain Gauge 3	Strain Gauge 4
Length Coordinate of Point A (mm)	26.7	26.7	26.7	26.7
Theta Coordinate of Point A (degrees)	171.4	81.4	351.4	261.4
Length Coordinate of Point B (mm)	23.3	23.3	23.3	23.3
Theta Coordinate of Point B (degrees)	171.4	81.4	351.4	261.4
Length Coordinate of Point C (mm)	23.3	23.3	23.3	23.3
Theta Coordinate of Point C (degrees)	188.6	98.6	8.6	278.6
Length Coordinate of Point D (mm)	26.7	26.7	26.7	26.7
Theta Coordinate of Point D (degrees)	188.6	98.6	8.6	278.6

Table 23 Input parameters for some sample cases

Parameters and Cases	Case 1	Case 2	Case 3	Case 4	Case 5	Case 6	Case 7	Case 8	Case 9
Outer Diameter of the shaft (mm)	12	12	12	12	12	12	12	12	12
Inner Diameter of the shaft (mm)	11.2	11.2	11.2	11.2	11.2	11.2	11.2	11.2	11.2
Total Length of the shaft (mm)	30	30	30	30	30	30	30	30	30
Offset in x direction (mm)	0	0	0.02	0	0	0	0	0.02	0.02
Offset in y direction (mm)	0	0.02	0.02	0	0	0.02	0.02	0.02	0.02
Torque (Nmm)	2100	2100	2100	2100	2100	2100	2100	2100	2100
Bending Load at the Load End (N)	0	0	0	2.5	5	2.5	5	2.5	5
Young's Modulus - Aluminium	68947.6	68947.6	68947.6	68947.6	68947.6	68947.6	68947.6	68947.6	68947.6
Poisson's Ratio	0,3	0,3	0,3	0,3	0,3	0,3	0,3	0,3	0,3

Table 24 Strain in the four regions and the Sensitivity for the sample cases

Parameters and Cases	Case 1	Case 2	Case 3	Case 4	Case 5	Case 6	Case 7	Case 8	Case 9
ε1 Tensile (in με)	483.899	483.912	463.316	483.899	483.899	483.737	483.562	463.141	462.967
ε1 Compressive (in με)	-483.899	-483.912	-463.316	-483.899	-483.899	-484.086	-484.261	-463.491	-463.665
ε2 Tensile (in με)	483.899	506.882	506.866	476.172	468.446	498.981	491.079	498.964	491.063
ε2 Compressive (in με)	-483.899	-506.882	-506.866	-491.625	-499.352	-514.783	-522.684	-514.767	-522.668
ε3 Tensile (in με)	483.899	483.912	506.866	483.899	483.899	483.737	483.562	506.691	506.516
ε3 Compressive (in με)	-483.899	-483.912	-506.866	-483.899	-483.899	-484.086	-484.261	-507.040	-507.215
ε4 Tensile (in με)	483.899	463.323	463.316	491.6256	499.352	470.875	478.426	470.868	478.420
ε4 Compressive (in με)	-483.899	-463.323	-463.316	-476.172	-468.446	-455.771	-448.219	-455.764	-448.213
Sensitivity	0.9678	0.9690	0.9702	0.9678	0.9678	0.9690	0.9690	0.9702	0.9702

It can be seen that as the absolute eccentricity increases, the value of sensitivity increases. It is also to be noted that although the values in the third, eighth and ninth case are the same, they actually differ in their fourth decimal digit.

5.5. Location of strain gauges on an existing transducer

There are two extreme ways of arranging strain gauges. If we know the eccentricity, then strain gauges can be placed in either of the two arrangements

The table below shows different cases for the considered torque transducer and the measurement of strain at different regions. The least distance is located at different zones which are indicated in the first row of Table 26.

5.5.1 The least distance is midway between the two strain gauge

Here the least distance between the outer and inner circle is located at the mid angle between two successive strain gauges. The other strain gauges will be placed accordingly each of them at 90 degrees apart.

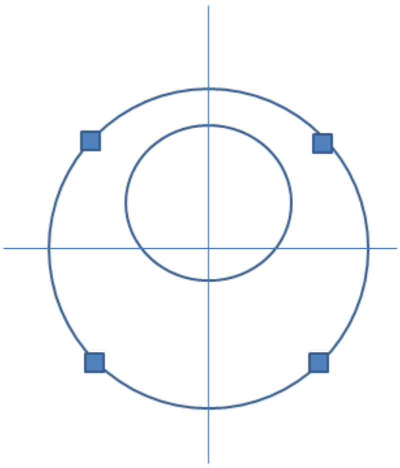


Figure 74 Least distance is midway between the two strain gauges

Table 25 Strain in four regions and the sensitivity for Case - least distance is midway between the two strain gauges

Top	Bottom	Left	Right	Top Left	Bottom Right	Top Right	Bottom Left
Region 1	Region 1	Region 1	Region 1	Region 1	Region 1	Region 1	Region 1
494.2379	463.8338	494.4058	463.6523	499.7517	469.2453	468.9982	499.9988
-505.5127	-474.4097	-505.3448	-474.5912	-499.9988	-468.9982	-469.2453	-499.7517
Region 2	Region 2	Region 2	Region 2	Region 2	Region 2	Region 2	Region 2
494.2379	463.8338	463.6523	494.4058	492.0122	461.5313	492.0412	461.5065
-505.5127	-474.4097	-474.5912	-505.3448	-507.7107	-476.7357	-507.7397	-476.7108
Region 3	Region 3	Region 3	Region 3	Region 3	Region 3	Region 3	Region 3
474.4097	505.5127	474.5912	505.3448	469.0045	499.9840	499.7529	469.2379
-463.8338	-494.2379	-463.6523	-494.4058	-469.2590	-499.7443	-500.0074	-468.9983
Region 4	Region 4	Region 4	Region 4	Region 4	Region 4	Region 4	Region 4
474.4097	505.5127	505.3448	474.5912	476.7236	507.7242	476.7236	507.7242
-463.8338	-494.2379	-494.4058	-463.6523	-461.5199	-492.0264	-461.5199	-492.0264
0.9689970	**0,9689970**	**0,9689970**	**0,9689970**	**0,9689934**	**0,9689956**	**0,9690053**	**0,9689870**
Approximately 0.9689							

Table 26 Rotation and bending load correlation – Case A

If hole is fixed at top (ey = 0.02mm), then bending load can be assumed to be at an angle	Sensitivity	If bending load is always acting at top, then least distance will be at
Bending load at Top	0.968997	Top
Bending load at Top Right	0.968993	Top Left
Bending load at Right	0.968997	Left
Bending load at Bottom Right	0.968987	Bottom Left
Bending load at Bottom	0.968997	Bottom
Bending load at Bottom Left	0.968995	Bottom Right
Bending load at Left	0.968997	Right
Bending load at Top Left	0.969005	Top Right
Average Value	0.968996	Average Value
Ideal Value	0.967790	Ideal Value

The above Table shows the various sensitivity values when the least distance is at different locations. Here each value corresponds to a case, wherein the bending load is rotated by a distance. For example, in Matlab, it is always considered that bending load acts form the top to the uppermost point of the shaft. If the bending load is at top and if the least distance is at top right, it also can be thought as a case with the least distance at top and the bending load acting at an angle along top left. This table is just for illustration purpose.

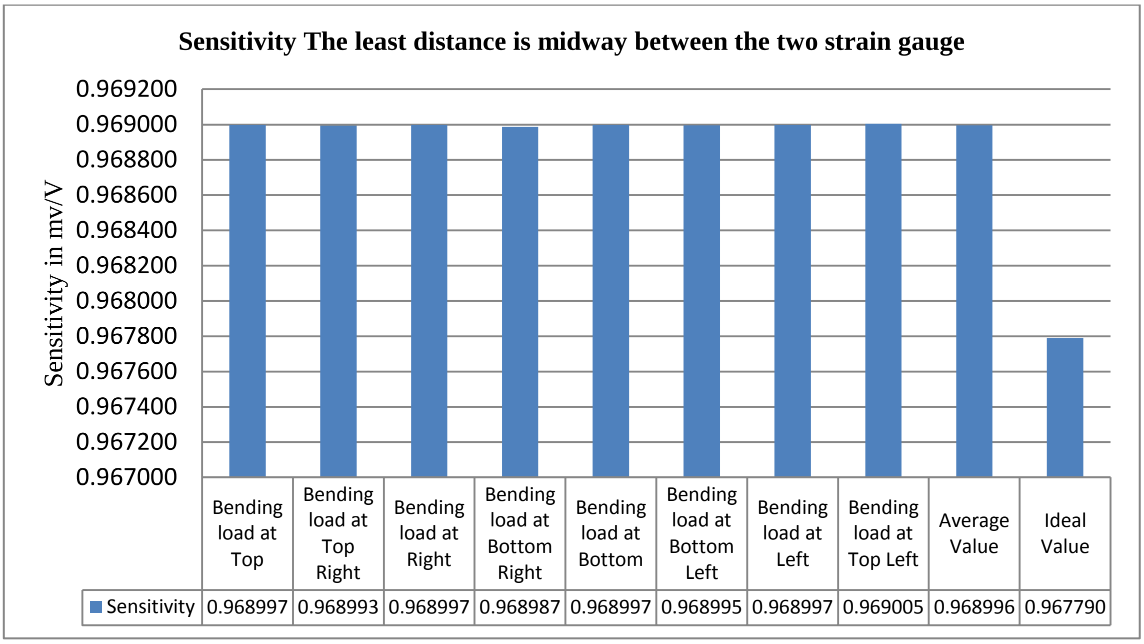


Figure 75 Sensitivity comparison for the case where least distance is in midway between the two strain gauges

5.5.2 One strain gauge is placed at the least distance (thin wall)

One strain gauge will be placed at the point where the distance between outer and inner circle is the least followed by the other three strain gauges at 90, 180 and 270 degrees apart respectively.

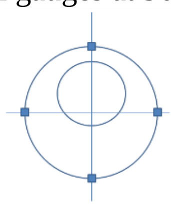


Figure 76 One strain gauge is placed at the least distance

Table 27 Strain in four regions and the sensitivity for Case - One strain gauge is placed at the least distance

Top	Bottom	Left	Right	Top Left	Bottom Right	Top Right	Bottom Left
Region 1	Region 1	Region 1	Region 1	Region 1	Region 1	Region 1	Region 1
483.7376	484.0870	506.8787	463.3261	501.2925	457.9870	478.3261	478.5732
-484.0870	-483.7376	-506.8787	-463.3261	-512.4649	-468.6653	-489.4985	-489.2514
Region 2	Region 2	Region 2	Region 2	Region 2	Region 2	Region 2	Region 2
498.9794	455.7742	476.1579	476.1958	478.3261	478.5732	501.2925	457.9870
-514.7802	-470.8762	-491.6286	-491.6665	-489.4985	-489.2514	-512.4649	-468.6653
Region 3	Region 3	Region 3	Region 3	Region 3	Region 3	Region 3	Region 3
483.7448	484.0734	463.3264	506.8700	468.6653	512.4649	489.2514	489.4985
-484.1017	-483.7314	-463.3338	-506.8774	-457.9870	-501.2925	-478.5732	-478.3261
Region 4	Region 4	Region 4	Region 4	Region 4	Region 4	Region 4	Region 4
470.8767	514.7788	491.6473	491.6473	489.2514	489.4985	468.6653	512.4649
-455.7754	-498.9786	-476.1773	-476.1773	-478.5732	-478.3261	-457.9870	-501.2925
0.9690190	**0.9690076**	**0.9690055**	**0.9690199**	**0.9690147**	**0.9690147**	**0.9690147**	**0.9690147**
Approximately 0.9690							

Table 28 Rotation and bending load correlation – Case B

If hole is fixed at top (ey = 0.02mm), then bending load can be assumed to be at an angle	**Sensitivity**	**If bending load is always acting at top, then least distance will be at**
Bending load at Top	0.969019	Top
Bending load at Top Right	0.969014	Top Left
Bending load at Right	0.969005	Left
Bending load at Bottom Right	0.969014	Bottom Left
Bending load at Bottom	0.969007	Bottom
Bending load at Bottom Left	0.969014	Bottom Right
Bending load at Left	0.969020	Right
Bending load at Top Left	0.969014	Top Right
Average Value	0.969013	Average Value
Ideal Value	0.967790	Ideal Value

The above Table shows the various sensitivity values when the least distance is at different locations. Here each value corresponds to a case, wherein the bending load is rotated by a distance. For example, in Matlab, it is always considered that bending load acts form the top to the uppermost point of the shaft. If the bending load is at top and if the least distance is at top right, it also can be thought as a case with the least distance at top and the bending load acting at an angle along top left. This table is just for illustration purposes as to show that bending load acting at an angle at top-right when the least distance is at the top is similar to bending load acting at the top when the least distance is at the top left.

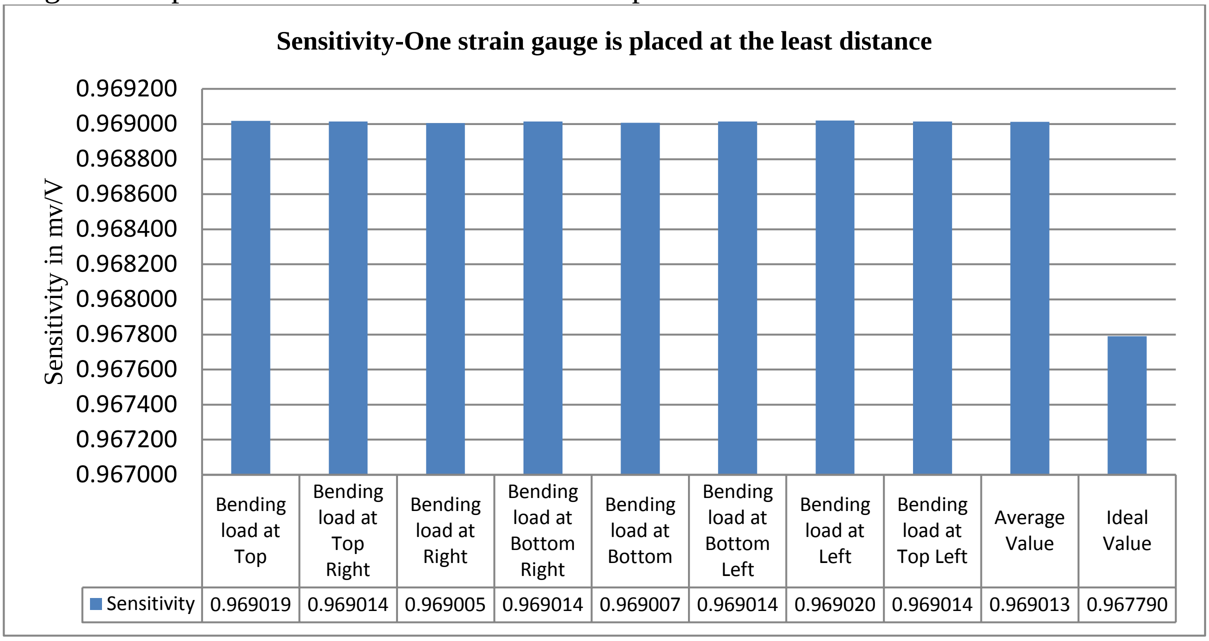


Figure 77 Sensitivity comparison for the case where one strain gauge is placed at the thin-wall

From the two locations, it can be seen that the sensitivity value is closer to the ideal value 0,967790 in the case when the least distance is midway between the two strain gauges. Therefore, it is always desirable to position the strain gauges in such a way that the least distance is always between the two strain gauges.

5.6. Sensitivity due to axis inclination

Axis inclination is again another geometrical tolerance that can be an influencing parameter as far as strain gauge sensitivity is concerned. Axis inclination refers to the alignment between the inner circle axis and outer circle axis. For the given transducer, the reasonable value of inclination is assumed as 0.3 degrees and the sensitivity value was computed. 0.3 degrees refers to the alignment between the outer circle axis and the inner circle axis at the midpoint. From the lateral view, the shaft will like the figures below

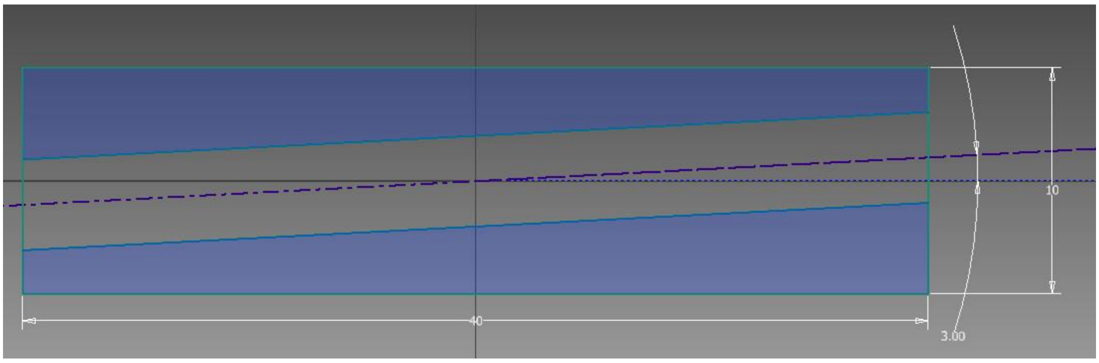


Figure 78 Axis inclination of an exaggerated shaft

Figure 78 is an exaggerated representation. It is a 10 mm shaft with 8 mm inner hole, which is aligned at the centre by 3 degrees. However, since Mechanica gives rises to errors for a short shaft, the length of the shaft was increased to 120 mm with the angle of inclination being a reasonable value of 0.3 degrees. The diameter of the outer shaft is 12 mm with the inner hole being 10 mm.

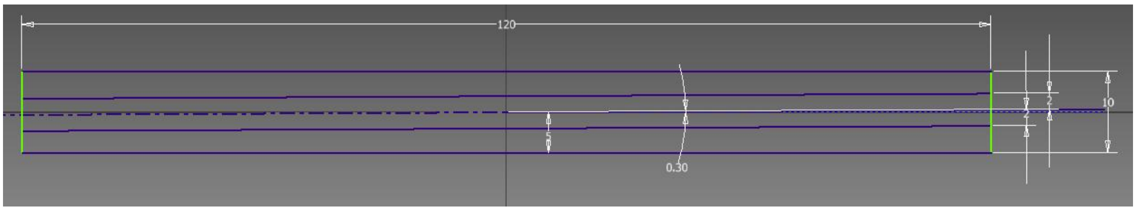


Figure 79 Axis inclination for a long shaft

5.6.1 Max principal strain along a particular angle throughout the shaft length

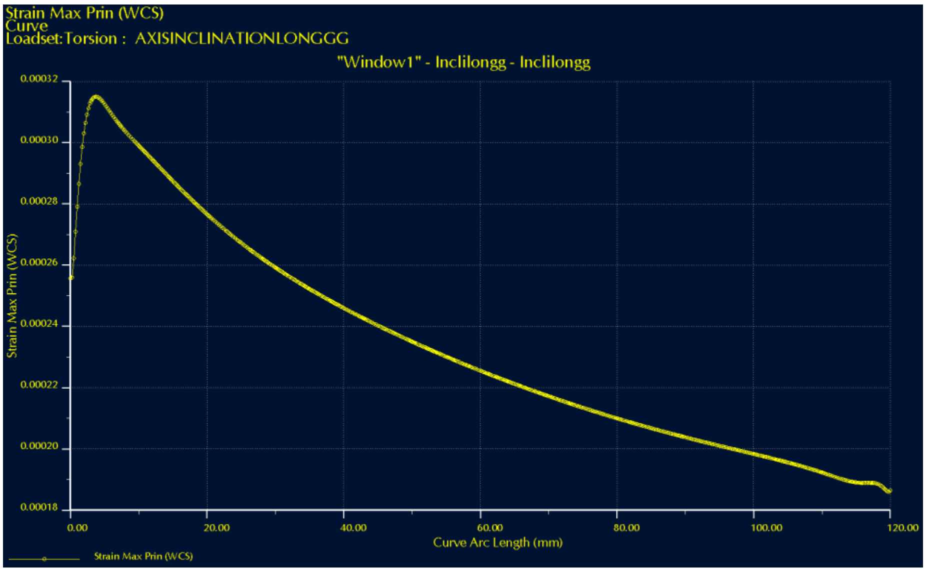


Figure 80 Maximum principal strain along 90 degree throughout the shaft length in FEA

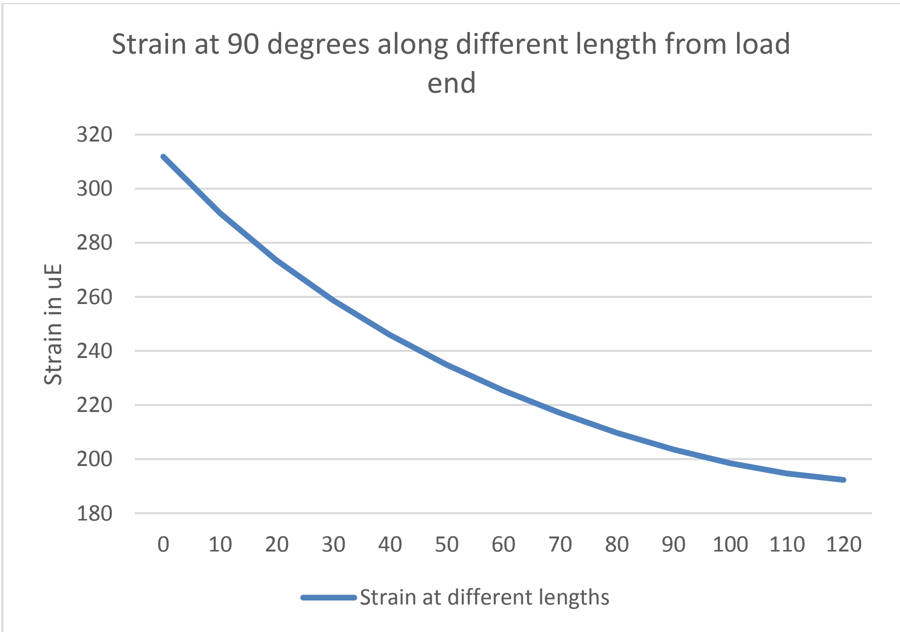


Figure 81 Maximum principal strain along 90 degree throughout the shaft length in Matlab

Maximum principal strain along a straight line is to be computed for verification. The plot from Mechanica shows a linear relationship along the line. From a hind sight, it can be observed that at the load end, the offset in y direction is the maximum and it continues reducing.

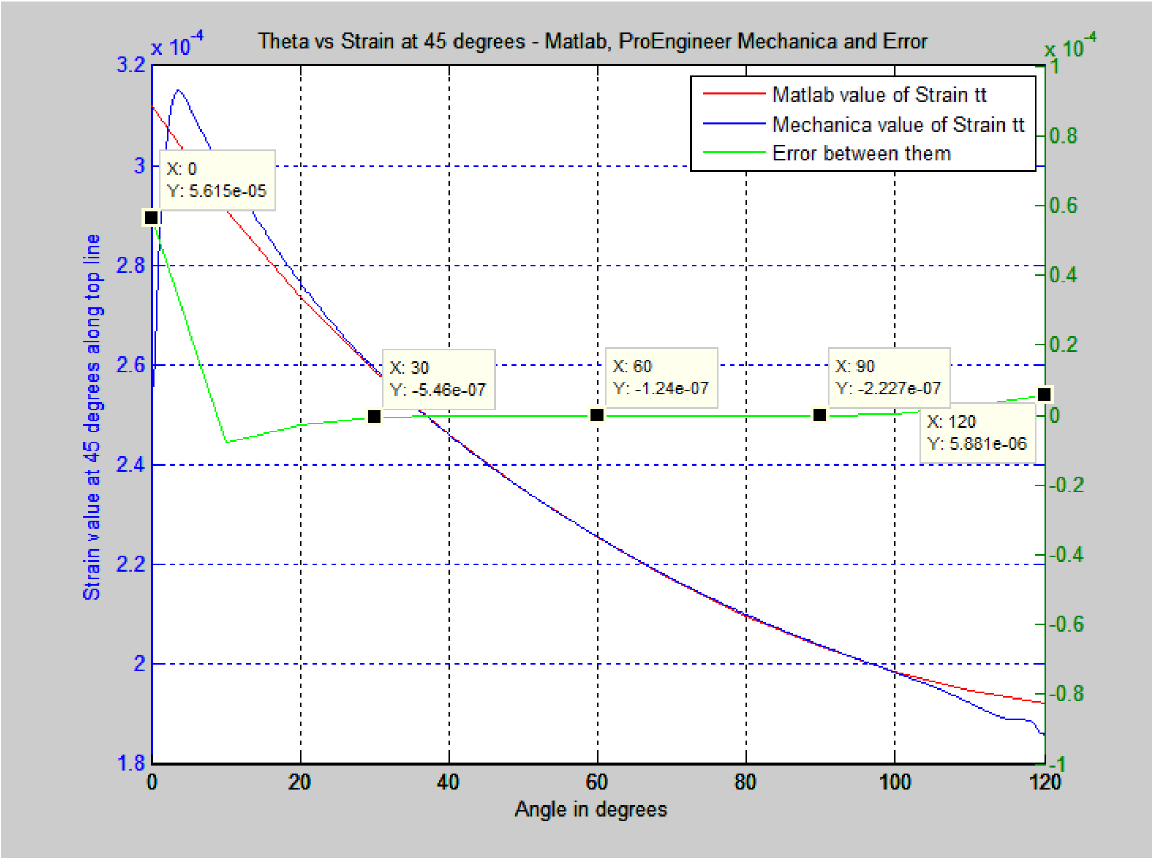


Figure 82 Maximum principal strain along 90 degree throughout the shaft length – FEA vs Matlab

It can be from the Mechanica plot that the value of strain continues dripping as with the offset. An exactly the same profile is obtained when the offset continues dripping for a shaft using Matlab. Thus barring the constraint effects, it can be concluded that the strain along a particular angle throughout the shaft is in accordance with the FEA.

5.6.2 Max principal strain along center (length 60 mm from load end)

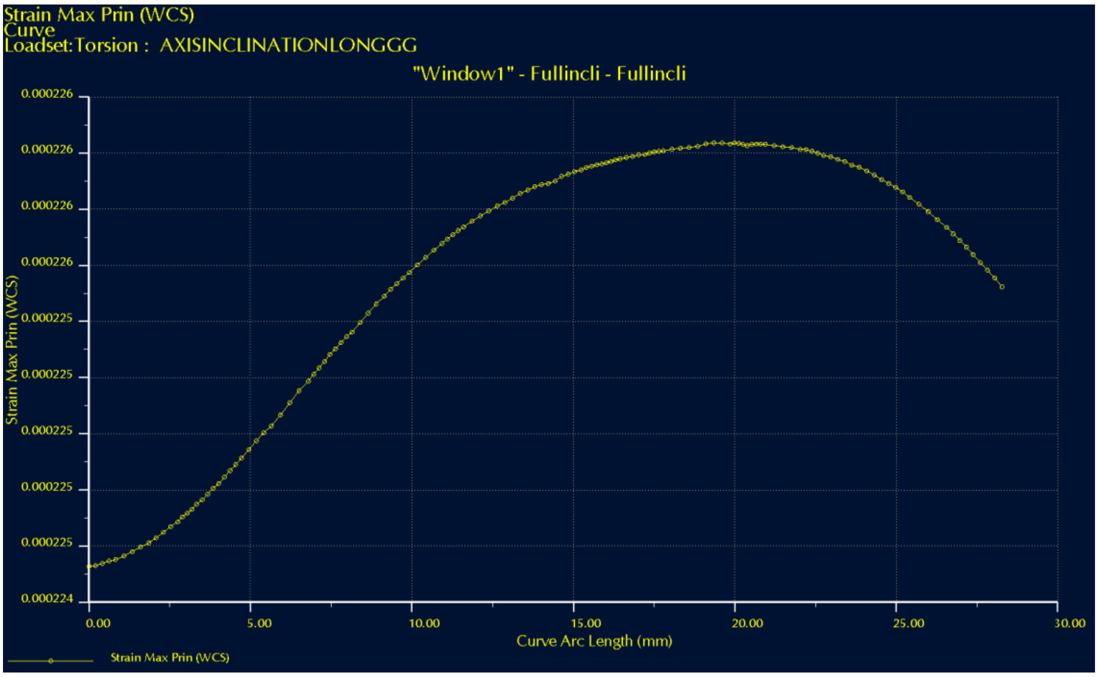


Figure 83 Maximum principal strain along the central slice of the shaft in FEA

As per the case at the center, the offset is 0 in y direction. Thus it can be considered to be an ideal concentric slice at the center. As a result, the value at the center shall be 225.4 uE. However, the value in Mechanica is not precisely 225.4 uE as it continues to fluctuate between 226 and 225 uE. Although the difference is very negligible, the value is not in accordance with the FEA. It should have been the same constant value of 225.4 uE in Mechanica, but because of the constraint effects, it was not meant to be.

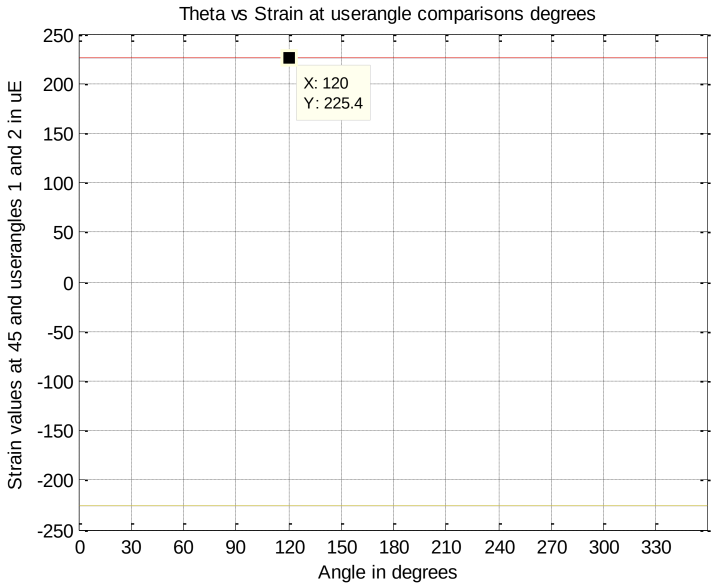


Figure 84 Maximum principal strain along the central slice of the shaft in Matlab

5.6.3 Max principal strain along length 30 mm (of a 120 mm shaft)

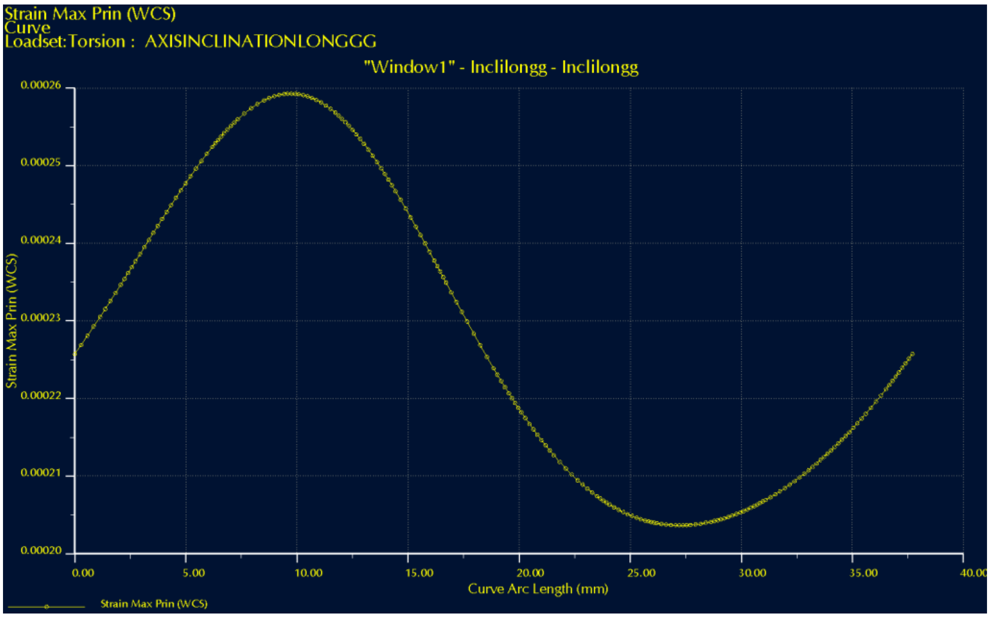


Figure 85 Maximum principal strain along the slice at length 30mm from load end of the shaft in FEA

This was a case, wherein the principal strain is checked at a length of 30 mm from the center. At the center using trigonometry, the value of yoffset was found to be tan(0.3)x30 which is 0,1571mm. This value is considered as the yoffset value and is entered in Matlab. The following profile is obtained.

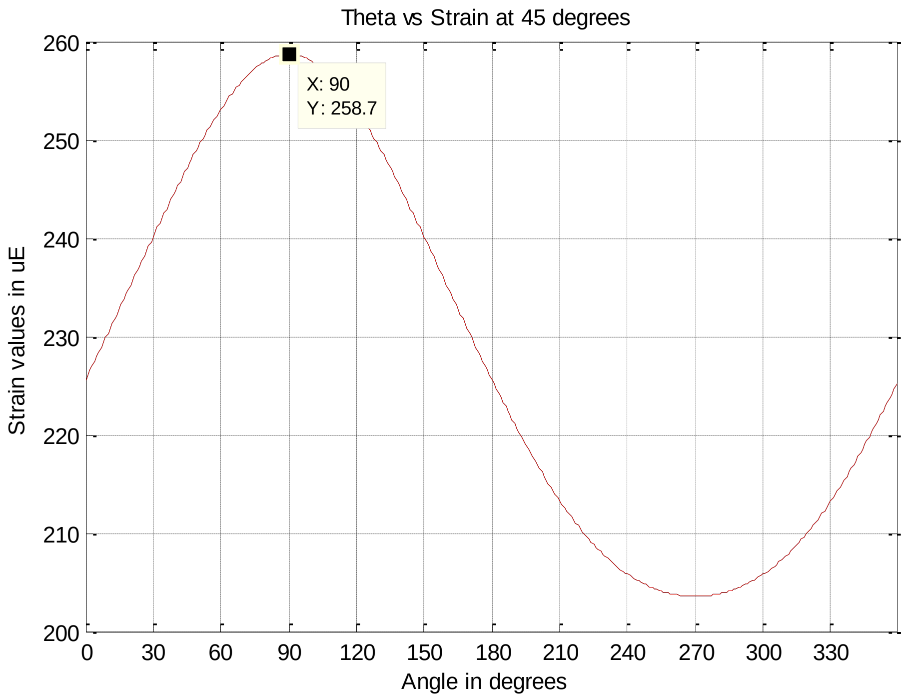


Figure 86 Maximum principal strain along the slice at length 30mm from load end of the shaft in Matlab

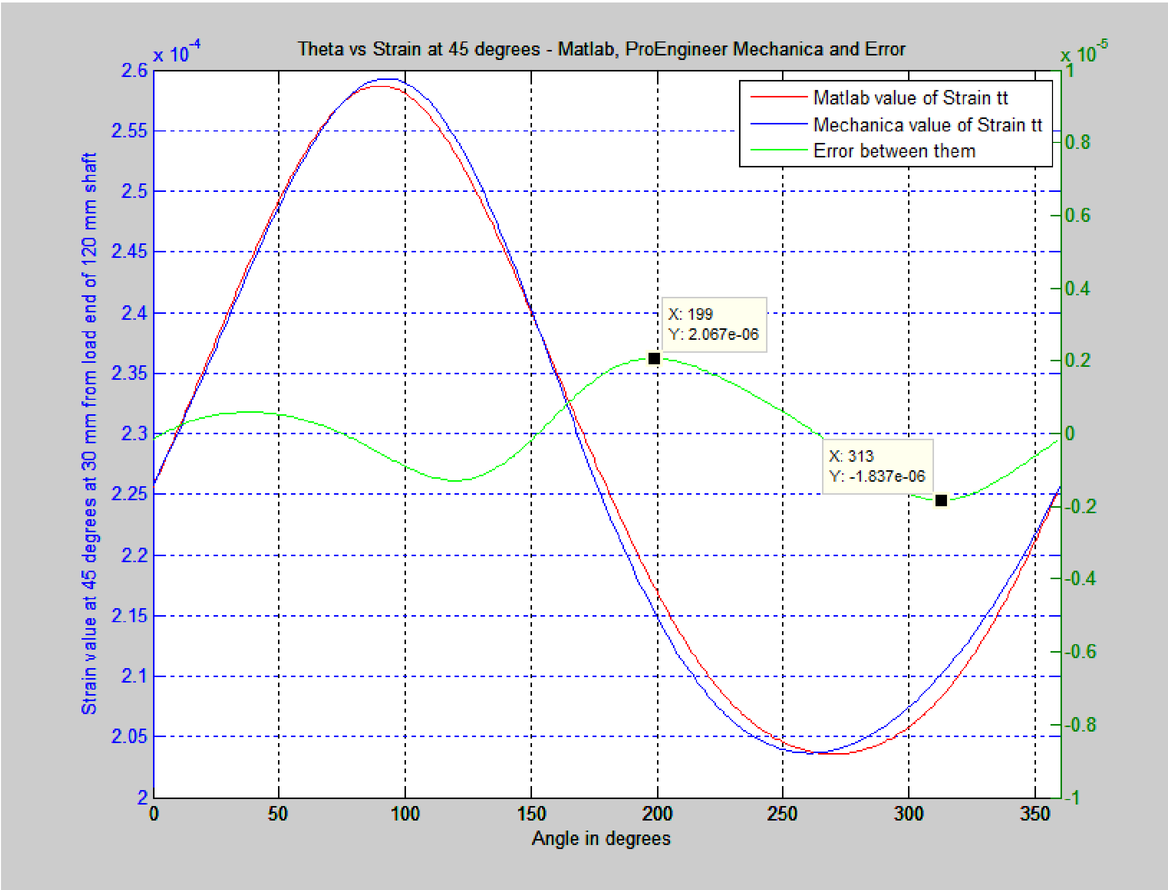


Figure 87 Matlab and ProEngineer Mechanica comparison

It can be seen that the profile of the two curves at length 30mm is similar in Mechanica as well as in Matlab with an error of 1%. Thus it can be concluded that axis inclination with respect to torsion alone is feasible and is verified.

5.7. DOE

DOE was carried out in the three assembly imperfections to see the magnitude of effects.

- Strain gauge angle orientation imperfection
    - Two levels of 44 and 46 degrees have been identified. Thus if the inclination is 46 degree in the upper set of strain gauge, then it will be -44 in the lower set of the strain gauge with respect to the horizontal axis
- Strain gauge angle inclination imperfection
    - The angle between two successive strain gauges (in a 4 strain gauge system) is not 90 degrees. Two levels have been identified for this setup with 89 degree and 91 degree as the angle between two strain gauges
- Strain gauge length displacement imperfection
    - The strain gauges are intended to be kept at 4 locations usually, symmetrically along the same length. Two levels are identified with distance of 0.2 mm and -0.2 mm length displacement

Now that the three common errors were identified with their commonly noticeable levels, the next step is to visually represent their magnitude which these errors would affect. As a result, full factorial design with two levels with three factors was carried out and the corresponding sensitivity was found out. All these were performed for a transducer with D=12, d=11.2, l=30, ey=0.02, subjected to bending and torsion. Then the sensitivity is compared with an ideal case (with no offset and no errors) and an eccentric case with no errors.

Table 29 Design of Experiments

When ex = 0, ey = 0,02					With eccentricity, No errors	Ideal Case No Eccentricity, No Errors
Plus Arrangment						
Case	45+-1	90+-1	dely+-0,2	Sensitivity Value		
	A	B	C			
1	-	-	-	0.9684151	0.9690190	0.9677900
2	+	-	-	0.9683737	0.9690190	0.9677900
3	-	+	-	0.9684156	0.9690190	0.9677900
4	+	+	-	0.9683742	0.9690190	0.9677900
5	-	-	+	0.9684824	0.9690190	0.9677900
6	+	-	+	0.9684328	0.9690190	0.9677900
7	-	+	+	0.9684829	0.9690190	0.9677900
8	+	+	+	0.9684333	0.9690190	0.9677900

Table 30 Positive and negative effects in DOE

Effects	EYBar+	EYBar-	Eeffects	Positive & Negative Percentages	Overall Contributors
SG Angle Orientation (AO)	0.9684035	-0.96844900	-0.0000455	91.73387097	40.15887026
SG Angle Inclination (AI)	0.9684265	-0.96842600	0.0000005	0.784929356	0.441306267
SG Length Displacement (LD)	0.96845785	-0.96839465	0.0000632	99.21507064	55.78111209
AO and AI	0.96842625	-0.96842625	0.0000000	0	0
AO and LD	0.9684242	-0.96842830	-0.0000041	8.266129032	3.618711386
AO and LD	0.96842625	-0.96842625	0.0000000	0	0
ΣPositive	0.0000637	ΣNegative	0.0000496	ΣOverall	0.0001133

The different values of the sensitivity obtained are shown below. The first column in each case shows the sensitivity value obtained for each run using the design of experiments. The second column in each case shows the sensitivity value that is obtained for the same case if there were no errors. The third column in each case shows the sensitivity value that shall be obtained if the shaft has no eccentricity and no errors. Ideally this sensitivity value is the one that the other cases should show. The second column differs from the third in the fact that it has an eccentricity value associated with it.

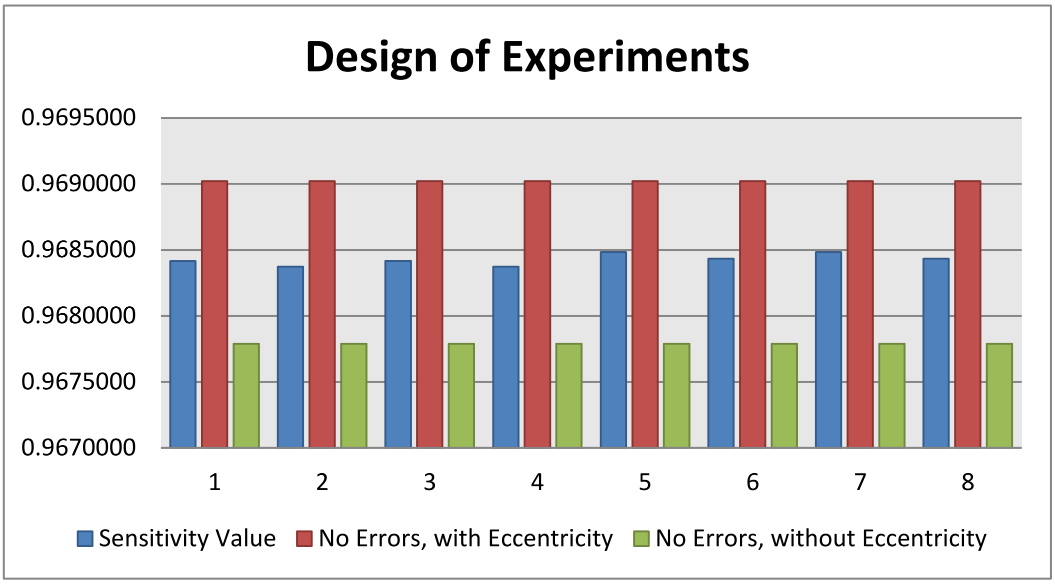


Figure 88 Design of experiments

The figure below shows the magnitude of effects of the different effects and their interactions. The magnitudes of the interaction effects are negligible. It can be seen that the biggest positive effect is that of the strain gauge length displacement, whereas the biggest negative effect is that of the strain gauge angle orientation.

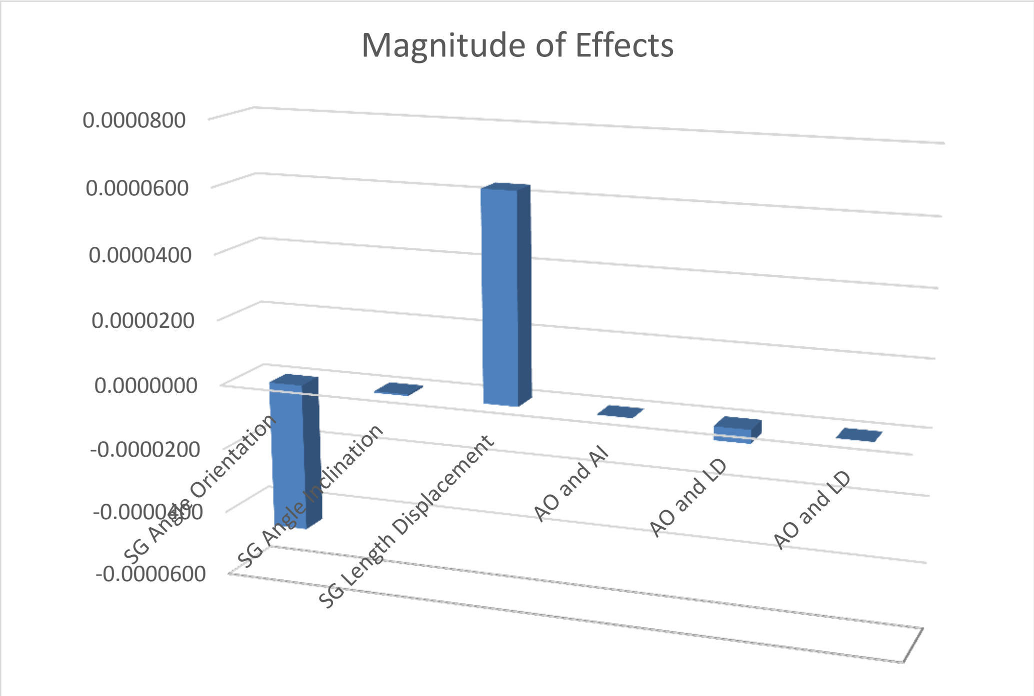


Figure 89 Magnitude of the effects in DOE

The pie chart (figure 90) shows the split up of different effects. It can be seen that the strain gauge length displacement contributes the most effect followed by strain gauge angle orientation.

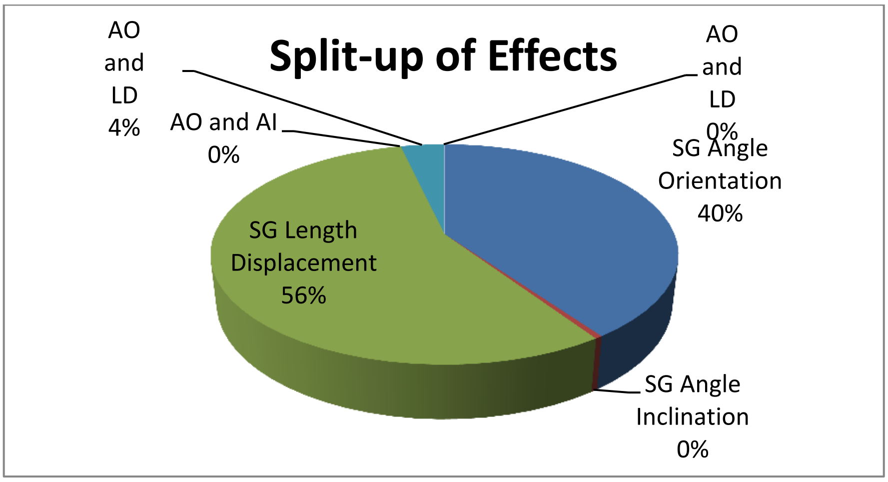


Figure 90 Split-up of the effects

Now that the magnitudes of the effects have been visualized, the next interesting result was that of the profile of these errors. Thus for the case, different values of errors have been identified and their values as a result of percentage of the ideal case with offset and without errors are computed.

Table 31 Error and their resultant sensitivity in percentages of ideal

	Values	Sensitivity Value mV/V	Percentage of Ideal	%
AO	43	0.9654393	0.244158281	0.2442
	43.5	0.9664704	0.137210617	0.1372
	44	0.9672071	0.060938345	0.0609
	44.5	0.9676492	0.015222459	0.0152
	45	0.9677965	0	0.0000
	45.5	0.9676490	0.015243131	0.0152
	46	0.9672068	0.060969381	0.0610
	46.5	0.9664700	0.137252062	0.1373
	47	0.9654387	0.244220581	0.2442
	47.5	0.9641134	0.382019377	0.3820
	48	0.9623494	0.566016851	0.5660
AI	-2	0.9677940	0.000258319	0.0003
	-1.5	0.9677951	0.000144659	0.0001
	-1	0.9677958	7.23293E-05	0.0001
	-0.5	0.9677963	2.06655E-05	0.0000
	0	0.9677965	0	0.0000
	0.5	0.9677964	1.03328E-05	0.0000
	1	0.9677960	5.16638E-05	0.0001
	1.5	0.9677953	0.000123993	0.0001
	2	0.9677943	0.000227321	0.0002
	2.5	0.9677930	0.000361648	0.0004
	3	0.9677914	0.000526973	0.0005
LD	-0.6	0.9677038	0.009579377	0.0096
	-0.4	0.9677347	0.006386048	0.0064
	-0.2	0.9677656	0.003192922	0.0032
	0	0.9677965	0	0.0000
	0.2	0.9678274	-0.003192718	-0.0032
	0.4	0.9678583	-0.006385232	-0.0064
	0.6	0.9678892	-0.009577543	-0.0096
	0.8	0.9679201	-0.012769649	-0.0128
	1	0.9679510	-0.015961552	-0.0160
	1.2	0.9679819	-0.019153251	-0.0192
	1.4	0.9680128	-0.022344746	-0.0223

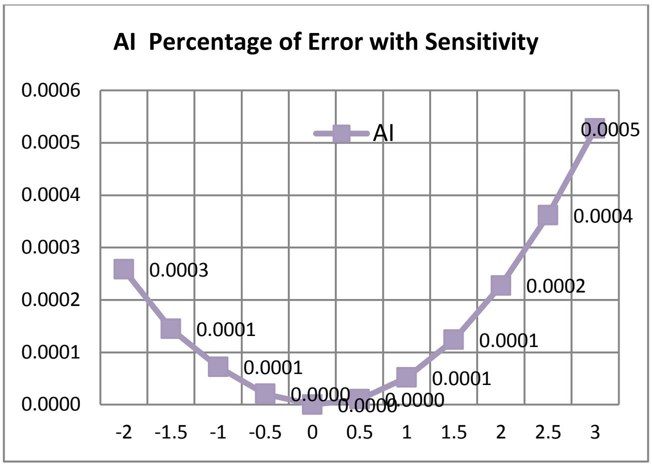


Figure 91 Angle inclination indicated as percentage of error of ideal sensitivity

The above plot shows the Angle inclination for different cases and their value as percentage of the value that they should have shown if there were no errors. It followed a symmetrical curve

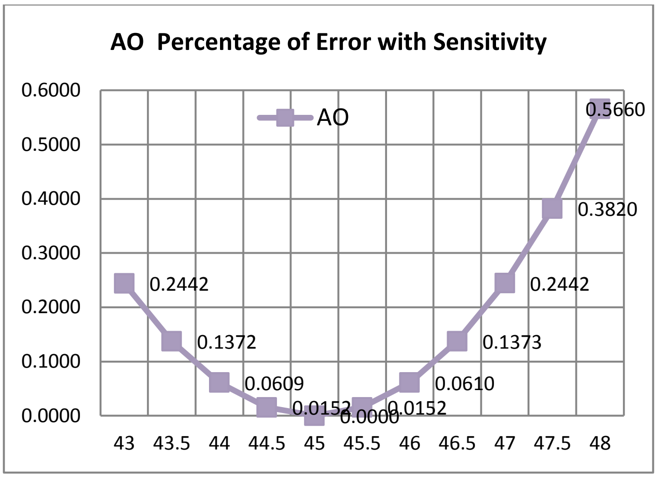


Figure 92 Angle orientation indicated as percentage of error of ideal sensitivity

Figure 92 shows the angle orientation for different cases and their value as percentage of the value that they should have shown if there were no errors. It followed a symmetrical curve as well; however the values as percentages were much more than that of the axis inclination error.

Figure 93 shows the profile of the length displacement error. Unlike the previous two cases, it followed a linear relationship.

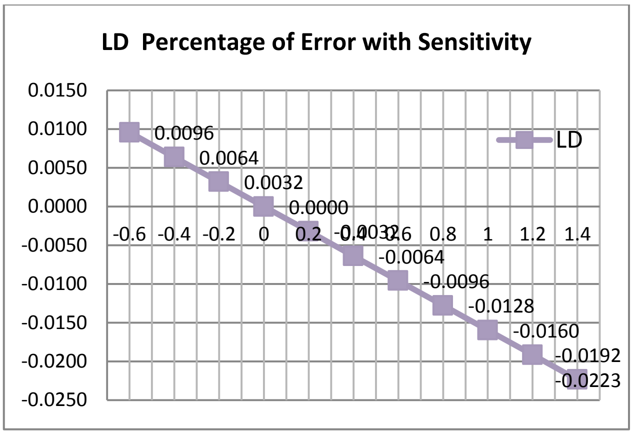


Figure 93 Length displacement indicated as percentage of error of ideal sensitivity

6 DISCUSSION AND CONCLUSIONS

6.1 Discussion

The calculation of the sensitivity for a shaft subjected to bending, axial or torsional load with and without geometrical imperfections and errors was the ultimate goal of the thesis. For computing that, the strain along different regions in user interested directions was to be calculated. Accordingly, the strains along different regions were calculated.

It was found that strain recorded by the strain gauge will be influenced by normal stress caused by axial load, normal stress caused by bending load and shear stress caused by torsion. All these values were individually found and verified with FEA for proceeding further. It was observed that the profile of the curves were sinusoidal except that of the axial stress and the shear stress caused by torsion in ideal case.

During verification, in some cases, the values obtained in ProE Mechanica have not been a constant value (with small irregular fluctuations). However theory suggests that it should be constant and that has been employed in Matlab. The minor fluctuations have been attributed to the effect of constraints in Mechanica.

Also, literature review from Hoffman and Khurmi suggested that the Shear stress caused by torsion should be constant at a particular angle throughout the length of the shaft. Keeping that in mind, the calculations were carried out in Matlab. However, in Mechanica, the values were varying. One possible reason shall be the effect of the constraints and some numerical computation errors. If the length of the shaft were drastically increased, then for much parts of length, the value of the shear stress along a particular angle seemed to be constant.

Based on discussions with people at Atlas Copco, there were three major errors that occur during the mounting of strain gauge layers on the shaft. These three errors were the errors occurring because of strain gauge angle orientation, angle inclination and strain gauge displacement error. These three parameters were considered in a factorial design in order to see the magnitude of effects. It was found out that the strain gauge length displacement contributed to the most error followed by strain gauge angle orientation error. A logical explanation can be attributed to the fact that strain varies greatly across the lengths than the angle orientation, which in turn is much more than the strain difference between successive degrees. The profiles of the three errors were plotted and it was observed that length displacement was linear whereas the angle orientation and angle misalignment were in the form of a symmetric parabola.

The next interesting thing was the measurement of strain in case of axis inclination between the inner and outer circles. It can be considered to be infinitely divided into different slices of shaft with varying eccentricity. Thus, at each slice, the value of the strain caused by torsion was verified with Matlab and they were the same barring the constraint effects. Thus, the initial thought that an inclined axis shaft will be the same as a shaft with infinite number of slices with increasing/decreasing offset value turned out to be right.

For all the comparisons between Matlab and ProEngineer Mechanica, the maximum difference in error in the values was not more than 1%, which is the least allowable boundary condition that can be stated in Mechanica.

It is to be noted that axis inclination has been employed only for a shaft subjected to torsion. It was not able to be computed for a shaft subjected to bending load, since it shall result in unsymmetrical loading and that shall arise to many complexities.

Since the scope of the thesis was only limited to the mechanical aspects of the strain gauge analysis, a good future work would be to work on the sensitivity to obtain the torque. For this, more knowledge on controls and signals would be needed.

6.2 Advantages and Conclusions

- It can be concluded that keeping the theory in mind, the values of Matlab are more accurate than FEA.
- The Matlab code is much faster and easier to use. All the inputs shall be entered in Excel and the entire time taken shall be less than a minute. In FEA, it takes more time to develop the model and then to analyze it.
- The Matlab code has the unique advantage of plotting the strain at an angle that the user is interested in. It is not possible to obtain the same in Mechanica, as the user has to manually deploy stress transformation for finding out the strain at an angle. Thereby, Matlab code is much more user friendly as long as the user has a basic understanding in solid mechanics.
- For the cases tested, it was found out if the eccentricity value increases, then the sensitivity of the strain gauge also increased.
- The general pattern in the strain readings for majority of the cases is a sinusoidal curve.
- If there is an offset in the inner circle axis, then it is always desirable to position the strain gauges in such a way that the least distance (thin-wall) is always between two strain gauges.
- The various misalignment errors were analysed and their magnitude of effects have been found. It is desirable to carefully position the strain gauge along the lateral length as it can have a significant impact in strain measurement. Also proper care must be taken to ensure that the strain gauge is aligned at 45 degrees.

7 RECOMMENDATIONS AND FUTURE WORK

- In the bending as well as in the axial loading, the load is always assumed to be acting along the neutral axis. However, in real life, it will be acting at a fixed point on the shaft. Though the difference between the neutral axis and the fixed point is negligibly small, it shall be interesting to know the effect if the load is always acting at the fixed point. The major disadvantage with that is that it shall lead to unsymmetrical loading which is very complex to solve in analytical way.
- Efforts can be put on FEA in order to ensure that the finite element methods are more similar to analytical methods using Matlab.
- Another good future work would be to extend the Matlab script such that the Matlab script can give the allowable tolerance. The current script can give only the sensitivity of a tolerance level, whereas the proposed one shall work on a number of tolerances and shall display the allowable tolerance level instead of manually checking for each tolerance limit.
- Axis inclination has been employed only for a shaft subjected to torsion. It was not able to be computed for a shaft subjected to bending load, since it shall result in unsymmetrical loading and that shall arise to many complexities. Therefore, another potential future work shall be to see and understand the irregularities when the axis is inclined and when a number of loads act.
- Combination of multiple bending loads will be an interesting scenario to watch. Though there wasn't any need for it in the thesis requirement, it would be good and interesting to know more about the consequences of multiple bending loads acting at different direction.
- Since the scope of the thesis was only limited to only the mechanical aspects of the strain gauge analysis, a good future work would be to work further on the sensitivity to obtain the torque. Thereby, the user can compare the torque obtained because of imperfections with the actual torque that it should display if it is free of imperfections. For this, more knowledge on controls and signals would be needed.

REFERENCES

Hilal Muftah and S.M.Haris, "*A Strain Gauge Based Torque Transducer for Measuring Dynamical Load in Rotation"*, Development, Energy, Environment, Economics, World Scientific and Engineering Academy and Society, 2010

R. Schicher and G. Wegener, "*Measuring Torque Correctly*", Hottinger, Baldwin Messtechnik Gmbh, Germany, 2002.

Hilal Muftah and S.M.Haris, *"A Strain Gauge Based System for Measuring Dynamic Loading on a Rotating Shaft"*, International Journal of Mechanics, Issue 1, Volume 5, 2011

Karl Hoffman, "*An Introduction to Stress Analysis and Transducer Design using Strain Gauges*", HBM, 1989

Harish Kumar, Anil Kumar, Sanjiv Gupta, "*Design studies and testing of a torque transducer*", Indian Journal of Pure & Applied Physics, Volume 49, October 2011, pp 653-656.

Warren C Young, "*Roark's Formulas for Stress & Strain*", McGraw Hill International Edition, Sixth Edition, 1989

Madhukar Vable, "*Mechanics of Materials"*, Second Edition, Michigan Technological University, 2009

R.S.Khurmi, J.K.Gupta, "*A Textbook of Machine Design*", Eurasia publishing house (pvt.) Ltd., 2008

Timoshenko.S, *"Strength of Materials"*, D.Van Nostrand Company, 1940

T.A.Wilson, *"The Eccentric Circular Tube"*, Aircraft Engineering, 1942

Karl Hoffmann, "*An Introduction to Measurement using Strain Gauges"*, Hottinger Baldwin Messtechnik GmbH, Darmstadt, 1989

National Instruments, "*Strain Gauge Measurement – A Tutorial*", Application Number 078, year 1998

Vishay Precision Group, "*Errors Due to Misalignment of Strain Gages*", Strain gauges and instruments, Technical Note TN-511, year 2010

Vishay Precision Group, "*Errors Due to Transverse Sensitivity in Strain Gages*", Strain gauges and instruments, Technical Note TN-509, year 2011

Vishay Precision Group, "*Errors Due to Wheatstone Bridge Nonlinearity*", Strain gauges and instruments, Technical Note TN-509, year 2010

George E.P. Box, Stuart Hunter J, William G Hunter, "*Statistics for Experimenters- Design, Innovation and Discovery*", Wiley Publications, Second Edition, 2005

Softwares used:

- *Matlab R2013a*
- *ProEngineer Mechanica Wildfire 5*
- *Autodesk Inventor Profession 2014 (Student Version)*

***NOTE** - The code 'sensshort.m' (Appendix 1) is recommended for strain gauge related applications, which is the purpose of this thesis, whereas the code 'senslong.m' (Appendix 2) is to be used for generic strength of materials related applications. All the computations in Matlab in this thesis are obtained using the code – 'sensshort.m'. An excel sheet 'Datainput' is used for entering the input parameters and the region coordinates.*
Let the below script be copied in Matlab and executed. The text gets automatically aligned when it is copied in Matlab and thereby more readability is obtained in Matlab than reading from this appendix.

APPENDIX 1- Matlab script - sensshort.m

```
clc
close all
clear all
format compact

%% Matlab Code sensshort
%% Section 1 - Parameter Input

filename = 'Datainput.xlsx';
sheet = 1;
xlRange = 'E5';
subsetA = xlsread(filename,sheet,xlRange);
D = subsetA;
filename = 'Datainput.xlsx';
sheet = 1;
xlRange = 'E6';
subsetA = xlsread(filename,sheet,xlRange);
d = subsetA;
filename = 'Datainput.xlsx';
sheet = 1;
xlRange = 'E7';
subsetA = xlsread(filename,sheet,xlRange);
L = subsetA;
filename = 'Datainput.xlsx';
sheet = 1;
xlRange = 'E8';
subsetA = xlsread(filename,sheet,xlRange);
xoffset = subsetA;
filename = 'Datainput.xlsx';
sheet = 1;
xlRange = 'E9';
subsetA = xlsread(filename,sheet,xlRange);
yoffset = subsetA;
filename = 'Datainput.xlsx';
sheet = 1;
xlRange = 'E10';
subsetA = xlsread(filename,sheet,xlRange);
inclin = subsetA;
filename = 'Datainput.xlsx';
sheet = 1;
xlRange = 'E12';
subsetA = xlsread(filename,sheet,xlRange);
T = subsetA;
filename = 'Datainput.xlsx';
sheet = 1;
xlRange = 'E13';
subsetA = xlsread(filename,sheet,xlRange);
W = subsetA;
filename = 'Datainput.xlsx';
sheet = 1;
xlRange = 'E14';
```

```
subsetA = xlsread(filename,sheet,xlRange);
P = subsetA;
filename = 'Datainput.xlsx';
sheet = 1;
xlRange = 'E16';
subsetA = xlsread(filename,sheet,xlRange);
E = subsetA;
filename = 'Datainput.xlsx';
sheet = 1;
xlRange = 'E17';
subsetA = xlsread(filename,sheet,xlRange);
v = subsetA;
filename = 'Datainput.xlsx';
sheet = 1;
xlRange = 'E18';
subsetA = xlsread(filename,sheet,xlRange);
ldiff = subsetA;
filename = 'Datainput.xlsx';
sheet = 1;
xlRange = 'E19';
subsetA = xlsread(filename,sheet,xlRange);
thetasplit_t = subsetA;
filename = 'Datainput.xlsx';
sheet = 1;
xlRange = 'E25';
subsetA = xlsread(filename,sheet,xlRange);
userang1 = subsetA;
filename = 'Datainput.xlsx';
sheet = 1;
xlRange = 'E26';
subsetA = xlsread(filename,sheet,xlRange);
userang2 = subsetA;

%% Section 2 - Torsion Computation

R =D/2;
r = d/2;
lambda = 1;

thetafirstinp_tor=0; %Starting theta point
thetalastinp_tor=359; %Ending theta point
lfirst=0; %Starting length point
llast = L; %Ending length point
lvals = lfirst:ldiff:llast;    %Range of length values

if (xoffset == 0 && yoffset == 0)
   thetaj = thetafirstinp_tor:thetasplit_t:thetalastinp_tor; %Range of theta values
   for ldx = 1:length(lvals);
      l = lvals(ldx);
      for thetadx = 1:length(thetaj)
         theta = thetaj(thetadx);
         tautorsion(thetadx,ldx) = (16*T*D)/(pi()*((D*D*D*D)-(d*d*d*d))); %Ideal case, tau torsion using formula
      end
   end
else
   exab = abs(xoffset);    %absolute value of xoffset
   eyab = abs(yoffset);    %absolute value of yoffset
   etor = sqrt(exab^2+eyab^2);
   ator = d/2;      %refer report            % inner circle radius
   btor = D/2;      %refer report             % outer circle radius
   mtor = d/D;
   ltor = etor/D;
   ptor = etor/ator;
   % Below are the conventions used for computing the limits of theta values according to the thetoritical framework
   if(xoffset>=0 && yoffset>=0)
      theta_eccentricity = atand(eyab/exab);
      thetadiff_tor = 180-theta_eccentricity;
      thetafirst_tor=thetadiff_tor-thetafirstinp_tor;
      thetalast_tor=thetadiff_tor-thetalastinp_tor;
   else
```

```
        if(xoffset<0 && yoffset>=0)
            theta_eccentricity = 180-(atand(eyab/exab));
            thetadiff_tor = 180-theta_eccentricity;
            thetafirst_tor=thetadiff_tor-thetafirstinp_tor;
            thetalast_tor=thetadiff_tor-thetalastinp_tor;
        else
            if(xoffset<=0 && yoffset<0)
                theta_eccentricity = 180+(atand(eyab/exab));
                thetadiff_tor = 180-theta_eccentricity;
                thetafirst_tor=thetadiff_tor-thetafirstinp_tor;
                thetalast_tor=thetadiff_tor-thetalastinp_tor;
            else
                if(xoffset>0 && yoffset<0)
                    theta_eccentricity = 360-(atand(eyab/exab));
                    thetadiff_tor = 180-theta_eccentricity;
                    thetafirst_tor=thetadiff_tor-thetafirstinp_tor;
                    thetalast_tor=thetadiff_tor-thetalastinp_tor;
                else
                    disp('Error ')
                end
            end
        end
    end

    if(thetafirst_tor<=0)
        thetafirstnew=thetafirst_tor;
    else
        thetafirstnew=thetafirst_tor;
    end

    if(thetalast_tor<=0)
        thetalastnew=720+thetalast_tor;
    else
        thetalastnew=thetalast_tor;
    end

    thetaj = thetafirstnew:thetasplit_t:thetalastnew;
    disp('------------------------------------------------------------------------');

    for ldx = 1:length(lvals);
        l = lvals(ldx);
        for thetadx = 1:length(thetaj);
            theta = thetaj(thetadx);
            % Extrapolation of Wilsons work, refer report
            a11_t = (ator^3)/(btor^2-ator^2);
            a12_t = 0;
            a13_t = (2*(ator^7)*(btor^2))/(((btor^2-ator^2)^2)*(btor^4-ator^4));
            a14_t = 0;
            a22_t = -(ator^4*btor^2)/((btor^2-ator^2)*(btor^4-ator^4));
            a23_t = 0;
            a24_t = (ator^8*btor^2)*((3*ator^8)-(ator^6*btor^2)+(3*ator^4*btor^4)-(3*ator^2*btor^6)-(2*btor^8))/(((btor^2-
ator^2)^2)*((btor^4-ator^4)^2)*(btor^6-ator^6));
            a33_t = ((ator^5*btor^2)*(ator^4+btor^4))/((btor^2-ator^2)*(btor^4-ator^4)*(btor^6-ator^6));
            a34_t = 0;
            a44_t = -(btor^2*ator^6)*((btor^10)+(2*ator^6*btor^6)+(2*ator^6*btor^4)+(ator^10))/((btor^2-ator^2)*(btor^4-
ator^4)*(btor^6-ator^6)*(btor^8-ator^8));
            a1_t = (a11_t*ptor)+(a12_t*ptor*ptor)+(a13_t*ptor*ptor*ptor)+(a14_t*ptor*ptor*ptor*ptor);
            a2_t = (a22_t*ptor*ptor)+(a23_t*ptor*ptor*ptor)+(a24_t*ptor*ptor*ptor*ptor);
            a3_t = (a33_t*ptor*ptor*ptor)+(a34_t*ptor*ptor*ptor*ptor);
            a4_t = (a44_t*ptor*ptor*ptor*ptor);
            be1_t = -a1_t*(btor^2);
            be2_t = -a2_t*(btor^4);
            be3_t = -a3_t*(btor^6);
            be4_t = -a4_t*(btor^8);
            Q_t = 1+((ltor^2)*(16*(mtor)^2/((1-mtor^2)*(1-mtor^4))))+((ltor^4)*(384*mtor^4/(((1-mtor^2)^2)*((1-mtor^4)^2))));
            phio_t = 32*T/(pi()*E*(D^4-d^4));
            phi_t = Q_t*phio_t;
            f1_t = -btor;
            f2_t = (a1_t-(be1_t/(btor*btor)))*cosd(theta);
            f3_t = 2*((a2_t*btor)-(be2_t/(btor*btor*btor)))*cosd(2*theta);
```

```
            f4_t = 3*((a3_t*btor*btor)-(be3_t/(btor*btor*btor*btor)))*cosd(3*theta);
            f5_t = 4*((a4_t*btor*btor*btor)-(be4_t/(btor*btor*btor*btor*btor)))*cosd(4*theta);
            fsum_t = f1_t+f2_t+f3_t+f4_t+f5_t;
            tautorsion(thetadx,ldx) = -E*phi_t*fsum_t;
        end
    end
end

thetaindividual_tor = 0:thetasplit_t:359;

Areaentire = pi*(D^2-d^2)/4; %Area of entire shaft
G = E/(2+(2*v)); %Shear modulus

if (thetasplit_t == 1)
    tautor = tautorsion(1:360,1:ldx);
else
    if (thetasplit_t == 0.5)
        tautor = tautorsion(1:719,1:ldx);
    else
        if (thetasplit_t == 0.1)
            tautor = tautorsion(1:3591,1:ldx);
        end
    end
end
gamma_strain = tautor/G*1000000; %shear strain

%% Section 3 - Bending, Axial and Combination Computation

thetafirst = thetafirstinp_tor;              %Starting theta point
thetalast = thetalastinp_tor;                %Ending theta point
thetadiff = thetasplit_t;                    %Resolution of theta values
lfirst=0;                                    %Starting length point
llast = L;                                   %Starting length point
thetavals = thetafirst:thetadiff:thetalast;  %Range of theta values
lvals = lfirst:ldiff:llast;                  %Range of length values

for lidx = 1:length(lvals);
    l = lvals(lidx);
    for thetaidx = 1:length(thetavals);
        theta = thetavals(thetaidx);
        %  Section 3.1 Centroid, Neutral Axis and Intended Point
        x_outer_circle = 0;                          % xbar of outer circle is zero
        y_outer_circle = 0;                          % ybar of outer circle is zero
        x_inner_circle = xoffset;                    % xbar of inner circle is equal to offset in x direction
        y_inner_circle = yoffset;                    % ybar of inner circle is equal to offset in y direction
        area_outer = pi*D*D/4;
        area_inner = pi*d*d/4;
        area_shaft = area_outer-area_inner;
        Xbar = ((x_outer_circle*area_outer)-(x_inner_circle*area_inner))/area_shaft;     % xbar of hollow circle
        Ybar = ((y_outer_circle*area_outer)-(y_inner_circle*area_inner))/area_shaft;     % ybar of hollow circle
        ynaoff = yoffset-Ybar;                       % ynaoff is the distance between y centroid and y bar of inner
circle
        xnaoff = xoffset-Xbar;                       % xnaoff is the distance between x centroid and x bar of inner
circle
        e1 = sqrt((xoffset^2)+(yoffset^2));          % eccentricity resultant between centre point and centroid
of inner
        e2 = sqrt((xnaoff^2)+(ynaoff^2));            % eccentricity resultant between centroid of hollow and
centroid of inner
        if(0<=theta<=90)                             % finding x1 and y1 of the interested point for the given input
angle
            x1 = R*cosd(theta);
            y1 = R*sind(theta);
            % disp('First Quadrant ');
        else if(theta<=180)
                x1 = R*cosd(180-theta);
                y1 = R*sind(180-theta);
                % disp('Second Quadrant ');
            else if (theta<=270)
                    x1 = R*cosd(180-theta);
                    y1 = R*sind(180-theta);
```

```
                % disp('Third Quadrant ');
            else if(theta<360)
                    x1 = R*cosd(180-theta);
                    y1 = R*sind(180-theta);
                    % disp('Fourth Quadrant ');
                end
            end
        end
    end

    %% Section 3.2 - Polar Moment of Inertia w.r.t. Neutral Axis

    Ona = sqrt((Xbar^2)+(Ybar^2));
    Ina = sqrt(((Xbar-x_inner_circle)^2)+((Ybar-y_inner_circle)^2));
    Jo = pi*(D^4)/32;
    Ji = (pi*(d^4)/32)+((pi*(d^2)/4)*e1^2);
    J = Jo-Ji;
    Jon = (pi*(D^4)/32)+((pi*(D^2)/4)*Ona^2);
    Jin = (pi*(d^4)/32)+((pi*(d^2)/4)*Ina^2);
    Jn = Jon - Jin;                 %Polar moment of inertia by Parallel axis theorem

    %% Section 3.3 - Second Moment of Inertia w.r.t. Neutral Axis

    if (abs(xoffset) <= abs(yoffset))
        Ixx_o = (pi*(D^4)/64)+((pi*(D^2)/4)*(Ona)^2);
        Ixx_i = (pi*(d^4)/64)+((pi*(d^2)/4)*(Ina)^2);
        Ixx = Ixx_o - Ixx_i;
        Iyy = Jn - Ixx;
    else
        Iyy_o = (pi*(D^4)/64)+((pi*(D^2)/4)*(Ona)^2);
        Iyy_i = (pi*(d^4)/64)+((pi*(d^2)/4)*(Ina)^2);
        Iyy = Iyy_o - Iyy_i;
        Ixx = Jn - Iyy;
    end
    b = 2*(D-d);

    %% Section 3.4 - Bending load parameters
    Mz = W*l;
    if (y1>Ybar)
        yver = (y1-Ybar);             % Bending Load acts on A or above YBar, resulting in Tensile Stress, hence yver>0
    else
        if (y1<Ybar)
            yver = -(Ybar-y1);        % Bending Load acts on B or below YBar, resulting in Compressive Stress, hence yver<0
        else
            yver = 0;                 % On the Centroidal Neutral Axis, yver is zero and hence My/I is also zero, hence no normal
bending stresses
        end
    end
    Izz = Iyy;

    %% Section 3.5 - Stress and Strain Computations, refer report for formulas

    sigma_axial = P/Areaentire;

    indicesref = 1/thetasplit_t;
    indiv_angles = (indicesref*theta)+1;
    indiv_angle = fix(indiv_angles);
    tau_torsion = tautorsion(indiv_angle);

    sigma_bending_z = -Mz*yver/Izz;
    sigmaYa(thetaidx,lidx) = 0;

    Mz_total(thetaidx,lidx) = Mz;
    yvertotal(thetaidx,lidx) = yver;
    Izz_total(thetaidx,lidx)=Izz;

    sigma_phi = -0.3*(sigma_bending_z+sigma_axial);
    strain_zz(thetaidx,lidx)= (sigma_bending_z+sigma_axial)*1000000/E;
    strain_tt(thetaidx,lidx) = sigma_phi*1000000/E;
  end
```

```
end

strain_45Max = ((strain_zz + strain_tt)/2)+(gamma_strain/2);
strain_ua1 = ((strain_zz + strain_tt)/2)-(((strain_zz - strain_tt)/2)*cosd(2*userang1))+((gamma_strain/2)*sind(2*userang1));
strain_ua2 = ((strain_zz + strain_tt)/2)-(((strain_zz - strain_tt)/2)*cosd(2*userang2))+((gamma_strain/2)*sind(2*userang2));

%% Section 4 - Plots

figure(1);
plot(thetaindividual_tor,tautor);
set(gca,'XTick',0:30:359 );
xlim([0 359]);
xlabel('Angle in degrees ')
ylabel('Shear Stress due to Torsion in MPa ')
title('Angle vs Shear Stress due to Torsion alone ');
grid on;

figure(2);
plot(thetavals,strain_zz);
title('Theta vs Strain zz at 45 degrees ');
xlabel('Angle in degrees ');
ylabel('Strain zz values in uE ');
xlim([0 359]);
set(gca,'XTick',0:30:359 );
grid on

figure(3);
plot(thetavals,strain_tt);
title('Theta vs Strain tt at 45 degrees ');
xlabel('Angle in degrees ');
ylabel('Strain tt values in uE ');
xlim([0 359]);
set(gca,'XTick',0:30:359 );
grid on

figure(4);
plot(thetavals,strain_45Max);
title('Theta vs Strain at 45 degrees ');
xlabel('Angle in degrees ');
ylabel('Strain values in uE ');
xlim([0 359]);
set(gca,'XTick',0:30:359 );
grid on

figure(5);
plot(thetavals,strain_ua1,thetavals,strain_ua2);
title('Theta vs Strain at userangle comparisons degrees ');
xlabel('Angle in degrees ');
ylabel('Strain values at 45 and userangles 1 and 2 in uE ');
xlim([0 359]);
set(gca,'XTick',0:30:359 );
grid on

%% Section 5 - Region Strain

option = menu('Choose the Number of Regions for Strain Gauges',...
   '0 Region ',...
   '1 Region ',...
   '2 Regions ',...
   '3 Regions ',...
   '4 Regions ',...
   '8 Regions');

switch option
   case 1
      nooftimes = 0;

   case 2
      nooftimes = 1:1;
      sheetx = 1;
```

```
    case 3
        nooftimes = 1:2;
        sheetx = 2;

    case 4
        nooftimes = 1:3;
        sheetx = 3;

    case 5
        nooftimes = 1:4;
        sheetx = 4;

    case 6
        nooftimes = 1:8;
        sheetx = 5;
end

if nooftimes ~= 0
    for trialdx = 1:length(nooftimes);
        trial = nooftimes(trialdx);

        filename = 'Datainput.xlsx';
        sheet = trialdx;
        xlRange = 'E32';
        subsetA = xlsread(filename,sheet,xlRange);
        x1u = -subsetA;

        filename = 'Datainput.xlsx';
        sheet = trialdx;
        xlRange = 'E33';
        subsetA = xlsread(filename,sheet,xlRange);
        sheet = sheetx;
        y1u = -subsetA;

        filename = 'Datainput.xlsx';
        sheet = trialdx;
        xlRange = 'E34';
        subsetA = xlsread(filename,sheet,xlRange);
        x2u = -subsetA;

        filename = 'Datainput.xlsx';
        sheet = trialdx;
        xlRange = 'E35';
        subsetA = xlsread(filename,sheet,xlRange);
        y2u = -subsetA;

        filename = 'Datainput.xlsx';
        sheet = trialdx;
        xlRange = 'E36';
        subsetA = xlsread(filename,sheet,xlRange);
        x3u = -subsetA;

        filename = 'Datainput.xlsx';
        sheet = trialdx;
        xlRange = 'E37';
        subsetA = xlsread(filename,sheet,xlRange);
        y3u = -subsetA;

        filename = 'Datainput.xlsx';
        sheet = trialdx;
        xlRange = 'E38';
        subsetA = xlsread(filename,sheet,xlRange);
        x4u = -subsetA;

        filename = 'Datainput.xlsx';
        sheet = trialdx;
        xlRange = 'E39';
        subsetA = xlsread(filename,sheet,xlRange);
        y4u = -subsetA;
```

```
xvals = 0:-ldiff:-L;
yvals = 0:-thetasplit_t:-359;

if (y1u < y4u || y2u < y3u )
    inters = 0;
else
    inters = 1;
end

if (inters==~0)
    m1_u=(y2u-y1u)/(x2u-x1u);
    if (abs(m1_u) == Inf)
        m1u = 0;
    else
        m1u = m1_u;
    end
    c1u=y1u-(m1u*x1u);
    m2_u=(x3u-x2u)/(y3u-y2u);
    if (abs(m2_u) == Inf)
        m2u = 0;
    else
        m2u = m2_u;
    end
    c2u=x2u-(m2u*y2u);
    m3_u=(y4u-y3u)/(x4u-x3u);
    if (abs(m3_u) == Inf)
        m3u = 0;
    else
        m3u = m3_u;
    end
    c3u=y3u-(m3u*x3u);
    m4_u=(x1u-x4u)/(y1u-y4u);
    if (abs(m4_u) == Inf)
        m4u = 0;
    else
        m4u = m4_u;
    end
    c4u=x4u-(m4u*y4u);

    for xdx = 1:length(xvals);
        x = xvals(xdx);
        for ydx = 1:length(yvals);
            y = yvals(ydx);
            d1(ydx,xdx) = y-(m1u*x);
            d2(ydx,xdx) = x-(m2u*y);
            d3(ydx,xdx) = y-(m3u*x);
            d4(ydx,xdx) = x-(m4u*y);
            if(d1(ydx,xdx)<=c1u && d2(ydx,xdx)<=c2u && d3(ydx,xdx)>=c3u && d4(ydx,xdx)>=c4u);
                Apoints(ydx,xdx)=1;
            else
                Apoints(ydx,xdx)=0;
            end
        end
    end

    Region1 = strain_ua1.*Apoints;
    Region2 = strain_ua2.*Apoints;
    Region1avg(trialdx) = mean(nonzeros(Region1));
    Region2avg(trialdx) = mean(nonzeros(Region2));
    disp('Region Average in uE is ');
    disp(Region1avg(trialdx));
    disp(Region2avg(trialdx));
else
    x1_u = x4u;
    y1_u = y4u;
    x2_u = x3u;
    y2_u = y3u;
    x3_u = x2u;
    y3_u = y2u;
```

```
            x4_u = x1u;
            y4_u = y1u;
            m1_u=(y2_u-y1_u)/(x2_u-x1_u);
            if (abs(m1_u) == Inf)
                m1u = 0;
            else
                m1u = m1_u;
            end
            c1u=y1_u-(m1u*x1_u);
            m2_u=(x3_u-x2_u)/(y3_u-y2_u);
            if (abs(m2_u) == Inf)
                m2u = 0;
            else
                m2u = m2_u;
            end
            c2u=x2_u-(m2u*y2_u);
            m3_u=(y4_u-y3_u)/(x4_u-x3_u);
            if (abs(m3_u) == Inf)
                m3u = 0;
            else
                m3u = m3_u;
            end
            c3u=y3_u-(m3u*x3_u);
            m4_u=(x1_u-x4_u)/(y1_u-y4_u);
            if (abs(m4_u) == Inf)
                m4u = 0;
            else
                m4u = m4_u;
            end
            c4u=x4_u-(m4u*y4_u);

            for xdx = 1:length(xvals);
                x = xvals(xdx);
                for ydx = 1:length(yvals);
                    y = yvals(ydx);
                    d1(ydx,xdx) = y-(m1u*x);
                    d2(ydx,xdx) = x-(m2u*y);
                    d3(ydx,xdx) = y-(m3u*x);
                    d4(ydx,xdx) = x-(m4u*y);
                    if(d1(ydx,xdx)<c1u && d2(ydx,xdx)<=c2u && d3(ydx,xdx)>c3u && d4(ydx,xdx)>=c4u);
                        Apoints1(ydx,xdx)=0;
                    else
                        Apoints1(ydx,xdx)=1;
                    end
                    if(d2(ydx,xdx)<=c2u && d4(ydx,xdx)>=c4u);
                        Apoints2(ydx,xdx)=1;
                    else
                        Apoints2(ydx,xdx)=0;
                    end
                    Apoints3 = Apoints1.*Apoints2;
                end
            end
            Region1 = strain_ua1.*Apoints3;
            Region2 = strain_ua2.*Apoints3;
            Region1avg(trialdx) = mean(nonzeros(Region1));
            Region2avg(trialdx) = mean(nonzeros(Region2));
            disp('Region Average in uE is ');
            disp(Region1avg(trialdx));
            disp(Region2avg(trialdx));
        end
    end
else
    return
end

%% Section 6 - Sensitivity

filename = 'Datainput.xlsx';
sheet = 1;
xlRange = 'E21';
```

```
subsetA = xlsread(filename,sheet,xlRange);
GF = subsetA;

filename = 'Datainput.xlsx';
sheet = 1;
xlRange = 'E22';
subsetA = xlsread(filename,sheet,xlRange);
Rg = subsetA;

filename = 'Datainput.xlsx';
sheet = 1;
xlRange = 'E23';
subsetA = xlsread(filename,sheet,xlRange);
V_wb = subsetA;

Del1R = Region1avg*GF*Rg*(10^-6)*1000;
Del2R = Region2avg*GF*Rg*(10^-6)*1000;

if (sheetx == 4)
   e1t = Region1avg(1);
   e1c = Region2avg(1);
   e2t = Region1avg(2);
   e2c = Region2avg(2);
   e3t = Region1avg(3);
   e3c = Region2avg(3);
   e4t = Region1avg(4);
   e4c = Region2avg(4);
end

Sens = (GF/(4*1000))*(e1t-e2c+e3t-e4c);
disp('Sensitivity in mV/V is ');
disp(Sens);
disp('We have reached the end of the program');
```

APPENDIX 2- Matlab script - senslong.m

```
clc
close all
clear all
format compact

%% Matlab Code senslong
%% Section 1 - Parameter Input

filename = 'Datainput.xlsx';
sheet = 1;
xlRange = 'E5';
subsetA = xlsread(filename,sheet,xlRange);
D = subsetA;
filename = 'Datainput.xlsx';
sheet = 1;
xlRange = 'E6';
subsetA = xlsread(filename,sheet,xlRange);
d = subsetA;
filename = 'Datainput.xlsx';
sheet = 1;
xlRange = 'E7';
subsetA = xlsread(filename,sheet,xlRange);
L = subsetA;
filename = 'Datainput.xlsx';
sheet = 1;
xlRange = 'E8';
subsetA = xlsread(filename,sheet,xlRange);
xoffset = subsetA;
filename = 'Datainput.xlsx';
sheet = 1;
xlRange = 'E9';
subsetA = xlsread(filename,sheet,xlRange);
yoffset = subsetA;
filename = 'Datainput.xlsx';
sheet = 1;
xlRange = 'E10';
subsetA = xlsread(filename,sheet,xlRange);
inclin = subsetA;
filename = 'Datainput.xlsx';
sheet = 1;
xlRange = 'E12';
subsetA = xlsread(filename,sheet,xlRange);
T = subsetA;
filename = 'Datainput.xlsx';
sheet = 1;
xlRange = 'E13';
subsetA = xlsread(filename,sheet,xlRange);
W = subsetA;
filename = 'Datainput.xlsx';
sheet = 1;
xlRange = 'E14';
subsetA = xlsread(filename,sheet,xlRange);
P = subsetA;
filename = 'Datainput.xlsx';
sheet = 1;
xlRange = 'E16';
subsetA = xlsread(filename,sheet,xlRange);
E = subsetA;
filename = 'Datainput.xlsx';
sheet = 1;
xlRange = 'E17';
subsetA = xlsread(filename,sheet,xlRange);
v = subsetA;
filename = 'Datainput.xlsx';
sheet = 1;
xlRange = 'E18';
subsetA = xlsread(filename,sheet,xlRange);
ldiff = subsetA;
```

```
filename = 'Datainput.xlsx';
sheet = 1;
xlRange = 'E19';
subsetA = xlsread(filename,sheet,xlRange);
thetasplit_t = subsetA;
filename = 'Datainput.xlsx';
sheet = 1;
xlRange = 'E25';
subsetA = xlsread(filename,sheet,xlRange);
userang1 = subsetA;
filename = 'Datainput.xlsx';
sheet = 1;
xlRange = 'E26';
subsetA = xlsread(filename,sheet,xlRange);
userang2 = subsetA;

% dl = L-l later;
% ey = yoffset + (dl*tand(m));

%% Section 2 - Torsion Computation

R =D/2;
r = d/2;
lambda = 1;


thetafirstinp_tor=0;
thetalastinp_tor=359;
lfirst=0;
llast = L;
lvals = lfirst:ldiff:llast;

if (xoffset == 0 && yoffset == 0)
   thetaj = thetafirstinp_tor:thetasplit_t:thetalastinp_tor;
   for ldx = 1:length(lvals);
      l = lvals(ldx);
      for thetadx = 1:length(thetaj)
         theta = thetaj(thetadx);
         tautorsion(thetadx,ldx) = (16*T*D)/(pi()*((D*D*D*D)-(d*d*d*d)));
      end
   end
else
   exab = abs(xoffset);
   eyab = abs(yoffset);
   etor = sqrt(exab^2+eyab^2);
   ator = d/2;                                                    % inner circle radius
   btor = D/2;                                                    % outer circle radius
   mtor = d/D;
   ltor = etor/D;
   ptor = etor/ator;
   if(xoffset>=0 && yoffset>=0)
      theta_eccentricity = atand(eyab/exab);
      thetadiff_tor = 180-theta_eccentricity;
      thetafirst_tor=thetadiff_tor-thetafirstinp_tor;
      thetalast_tor=thetadiff_tor-thetalastinp_tor;
   else
      if(xoffset<0 && yoffset>=0)
         theta_eccentricity = 180-(atand(eyab/exab));
         thetadiff_tor = 180-theta_eccentricity;
         thetafirst_tor=thetadiff_tor-thetafirstinp_tor;
         thetalast_tor=thetadiff_tor-thetalastinp_tor;
      else
         if(xoffset<=0 && yoffset<0)
            theta_eccentricity = 180+(atand(eyab/exab));
            thetadiff_tor = 180-theta_eccentricity;
            thetafirst_tor=thetadiff_tor-thetafirstinp_tor;
            thetalast_tor=thetadiff_tor-thetalastinp_tor;
         else
            if(xoffset>0 && yoffset<0)
               theta_eccentricity = 360-(atand(eyab/exab));
```

```
                thetadiff_tor = 180-theta_eccentricity;
                thetafirst_tor=thetadiff_tor-thetafirstinp_tor;
                thetalast_tor=thetadiff_tor-thetalastinp_tor;
            else
                disp('Error ')
            end
        end
    end
end

if(thetafirst_tor<=0)
    thetafirstnew=thetafirst_tor;
else
    thetafirstnew=thetafirst_tor;
end

if(thetalast_tor<=0)
    thetalastnew=720+thetalast_tor;
else
    thetalastnew=thetalast_tor;
end

thetaj = thetafirstnew:thetasplit_t:thetalastnew;
disp('------------------------------------------------------------------------');

for ldx = 1:length(lvals);
    l = lvals(ldx);
    for thetadx = 1:length(thetaj);
        theta = thetaj(thetadx);
        a11_t = (ator^3)/(btor^2-ator^2);
        a12_t = 0;
        a13_t = (2*(ator^7)*(btor^2))/(((btor^2-ator^2)^2)*(btor^4-ator^4));
        a14_t = 0;
        a22_t = -(ator^4*btor^2)/((btor^2-ator^2)*(btor^4-ator^4));
        a23_t = 0;
        a24_t = (ator^8*btor^2)*((3*ator^8)-(ator^6*btor^2)+(3*ator^4*btor^4)-(3*ator^2*btor^6)-(2*btor^8))/(((btor^2-
ator^2)^2)*((btor^4-ator^4)^2)*(btor^6-ator^6));
        a33_t = ((ator^5*btor^2)*(ator^4+btor^4))/((btor^2-ator^2)*(btor^4-ator^4)*(btor^6-ator^6));
        a34_t = 0;
        a44_t = -(btor^2*ator^6)*((btor^10)+(2*ator^6*btor^6)+(2*ator^6*btor^4)+(ator^10))/((btor^2-ator^2)*(btor^4-
ator^4)*(btor^6-ator^6)*(btor^8-ator^8));
        a1_t = (a11_t*ptor)+(a12_t*ptor*ptor)+(a13_t*ptor*ptor*ptor)+(a14_t*ptor*ptor*ptor*ptor);
        a2_t = (a22_t*ptor*ptor)+(a23_t*ptor*ptor*ptor)+(a24_t*ptor*ptor*ptor*ptor);
        a3_t = (a33_t*ptor*ptor*ptor)+(a34_t*ptor*ptor*ptor*ptor);
        a4_t = (a44_t*ptor*ptor*ptor*ptor);
        be1_t = -a1_t*(btor^2);
        be2_t = -a2_t*(btor^4);
        be3_t = -a3_t*(btor^6);
        be4_t = -a4_t*(btor^8);
        Q_t = 1+((ltor^2)*(16*(mtor)^2/((1-mtor^2)*(1-mtor^4))))+((ltor^4)*(384*mtor^4/(((1-mtor^2)^2)*((1-mtor^4)^2))));
        phio_t = 32*T/(pi()*E*(D^4-d^4));
        phi_t = Q_t*phio_t;
        f1_t = -btor;
        f2_t = (a1_t-(be1_t/(btor*btor)))*cosd(theta);
        f3_t = 2*((a2_t*btor)-(be2_t/(btor*btor*btor)))*cosd(2*theta);
        f4_t = 3*((a3_t*btor*btor)-(be3_t/(btor*btor*btor*btor)))*cosd(3*theta);
        f5_t = 4*((a4_t*btor*btor*btor)-(be4_t/(btor*btor*btor*btor*btor)))*cosd(4*theta);
        fsum_t = f1_t+f2_t+f3_t+f4_t+f5_t;
        tautorsion(thetadx,ldx) = -E*phi_t*fsum_t;
    end
end
end

thetaindividual_tor = 0:thetasplit_t:359;

Areaentire = pi*(D^2-d^2)/4;
G = E/(2+(2*v));

if (thetasplit_t == 1)
    tautor = tautorsion(1:360,1:ldx);
```

```
else
   if (thetasplit_t == 0.5)
      tautor = tautorsion(1:719,1:ldx);
   else
      if (thetasplit_t == 0.1)
         tautor = tautorsion(1:3591,1:ldx);
      end
   end
end
gamma_strain = tautor/G*1000000;

%% Section 3 - Bending, Axial and Combination Computation

thetafirst = thetafirstinp_tor;
thetalast = thetalastinp_tor;
thetadiff = thetasplit_t;
lfirst=0;
llast = L;

thetavals = thetafirst:thetadiff:thetalast;
lvals = lfirst:ldiff:llast;
% disp('-----------------------------------------------------------------------');

for lidx = 1:length(lvals);
   l = lvals(lidx);
   % %    disp('From the load end at a distance of ');
   % %    disp(l);
   % %    disp('Results associated with different angles are as follows ');
   % %    disp('----------------------------------');
   for thetaidx = 1:length(thetavals);
      theta = thetavals(thetaidx);
      % %    disp('At an angle of ');
      % %    disp(theta);
      %%  Section 3.1 Centroid, Neutral Axis and Intended Point

      x_outer_circle = 0;                                  % xbar of outer circle is zero
      y_outer_circle = 0;                                  % ybar of outer circle is zero
      x_inner_circle = xoffset;                            % xbar of inner circle is equal to offset in x direction
      y_inner_circle = yoffset;                            % ybar of inner circle is equal to offset in y direction
      area_outer = pi*D*D/4;
      area_inner = pi*d*d/4;
      area_shaft = area_outer-area_inner;
      Xbar = ((x_outer_circle*area_outer)-(x_inner_circle*area_inner))/area_shaft;      % xbar of hollow circle
      Ybar = ((y_outer_circle*area_outer)-(y_inner_circle*area_inner))/area_shaft;      % ybar of hollow circle
      % disp('Centroid X Bar of the Entire Shaft in mm is ');
      % disp(Xbar);
      % disp('Centroid Y Bar of the Entire Shaft in mm is ');
      % disp(Ybar);
      % disp('Neutral Axis is along the Centroid ')
      ynaoff = yoffset-Ybar;                               % ynaoff is the distance between y centroid and y bar of inner
circle
      xnaoff = xoffset-Xbar;                               % xnaoff is the distance between x centroid and x bar of inner
circle
      e1 = sqrt((xoffset^2)+(yoffset^2));                  % eccentricity resultant between centre point and centroid
of inner
      e2 = sqrt((xnaoff^2)+(ynaoff^2));                    % eccentricity resultant between centroid of hollow and
centroid of inner
      if(0<=theta<=90)                                     % finding x1 and y1 of the interested point for the given input
angle
         x1 = R*cosd(theta);
         y1 = R*sind(theta);
         % disp('First Quadrant ');
      else if(theta<=180)
            x1 = R*cosd(180-theta);
            y1 = R*sind(180-theta);
            % disp('Second Quadrant ');
         else if (theta<=270)
               x1 = R*cosd(180-theta);
               y1 = R*sind(180-theta);
               % disp('Third Quadrant ');
```

```
            else if(theta<360)
                x1 = R*cosd(180-theta);
                y1 = R*sind(180-theta);
                % disp('Fourth Quadrant ');
            end
        end
    end
end

% disp('X Coordinate of the interested point is ');
% disp(x1);
% disp('Y Coordinate of the interested point is ');
% disp(y1);

%% Section 3.2 - Inner Diameter (w.r.t Neutral Axis) Conditions and Parameters

if (abs(ynaoff) >= r && ynaoff <0)                          % third or fourth quadrant and y is greater than r (No
Segment formation)
    % disp('X bar above the Neutral Axis is Not Intended to be used ');
    % disp('Y bar above the Neutral Axis is Not Intended to be used ');
    % disp('No Area for the inner circle above the Neutral Axis is to be considered ');
    x_bar_inner_section = 0;
    y_bar_inner_section = 0;
    Area_inner_section = 0;
else
    if(ynaoff >= r && ynaoff >0)                             % first or second quadrant and y is greater than r (No
Segment formation)
        x_bar_inner_section = xoffset;
        y_bar_inner_section = yoffset;
        area_inner = pi*r*r;
        Area_inner_section = area_inner;
    else
        yoff_abs = abs(ynaoff);                              % one of the four quadrants and y is less than r (Segment
formation)
        a = sqrt(((d)^2)-((2*yoff_abs)^2));
        thetaseg = 2*(asind(a/d));
        area_inner_segment = (r*r/2)*((pi*thetaseg/180)-sind(thetaseg));
        area_inner_remaining = area_inner - area_inner_segment;
        x_inner_seg = xoffset;
        Aseg = (2/3)*r*r*r*sind(thetaseg/2)*sind(thetaseg/2)*sind(thetaseg/2);
        Bseg = area_inner_segment;
        if(ynaoff <0)
            y_inner_seg = (Aseg/Bseg)+yoffset;
        else
            y_inner_seg = -(Aseg/Bseg)+yoffset;
        end
        x_inner_rem = xoffset;
        y_inner_rem = ((y_inner_circle*area_inner)-(y_inner_seg*area_inner_segment))/area_inner_remaining;
        if(ynaoff>0 && ynaoff <r)                            % first and second quadrant and y is less than r
(Remaining formation)
            x_bar_inner_section = x_inner_rem;
            y_bar_inner_section = y_inner_rem;
            Area_inner_section = area_inner_remaining;
        else
            if(ynaoff<0 && ynaoff<r)                         % third and fourth quadrant and y is less than r (Segment
formation)
                x_bar_inner_section = x_inner_seg;
                y_bar_inner_section = y_inner_seg;
                Area_inner_section = area_inner_segment;
            else
                x_bar_inner_section = xoffset;               % y is zero
                y_bar_inner_section = 4*r/(3*pi);
                Area_inner_section = area_inner/2;
            end
        end
    end
end
% disp ('Area of the Inner Portion above the Neutral Axis in mm is ');
% disp (Area_inner_section);
% disp ('X bar of the Inner Portion above the Neutral Axis in mm is ');
```

```
% disp (x_bar_inner_section );
% disp ('Y bar of the Inner Portion above the Neutral Axis in mm is ');
% disp (y_bar_inner_section );

%% Section 3.3 - Outer Diameter (w.r.t Neutral Axis) Conditions and Parameters

A = sqrt(((D)^2)-((2*Ybar)^2));
thetaseg_outer = 2*(asind(A/D));
area_outer_segment = (R*R/2)*((pi*thetaseg_outer/180)-sind(thetaseg_outer));
area_outer_remaining = area_outer - area_outer_segment;
x_outer_seg = 0;
Aseg_outer = (2/3)*R*R*R*sind(thetaseg_outer/2)*sind(thetaseg_outer/2)*sind(thetaseg_outer/2);
Bseg_outer = area_outer_segment;
y_outer_seg = (Aseg_outer/Bseg_outer);
x_outer_rem = 0;
y_outer_rem = ((y_outer_circle*area_outer)-(y_outer_seg*area_outer_segment))/area_outer_remaining;
if (Ybar < 0)
  Area_outer_section = area_outer_remaining;
  x_bar_outer_section = 0;
  y_bar_outer_section =  -y_outer_rem;
else
  if (Ybar > 0)
    Area_outer_section = area_outer_segment;
    x_bar_outer_section = 0;
    y_bar_outer_section = y_outer_seg;
  else
    Area_outer_section = area_outer/2;
    x_bar_outer_section = 0;
    y_bar_outer_section = 4*D/(3*2*pi);
  end
end
% disp ('Area of the Outer Portion above the Neutral Axis in mm is ');
% disp (Area_outer_section);
% disp ('X bar of the Outer Portion above the Neutral Axis in mm is ');
% disp (x_bar_outer_section);
% disp ('Y bar of the Outer Portion above the Neutral Axis in mm is ');
% disp (y_bar_outer_section);

%% Section 3.4 - First Moment of Inertia w.r.t. Neutral Axis

Area_Effective_Neutralaxis = Area_outer_section - Area_inner_section;
Ybar_nueutralaxis = ((y_bar_outer_section * Area_outer_section)- (y_bar_inner_section *
Area_inner_section))/Area_Effective_Neutralaxis;
% disp('Effective Centroid for the Remaining Portion cut horizontally along the neutral axis is ');
% disp(Ybar_nueutralaxis);
ydiffna = abs(Ybar_nueutralaxis - Ybar);
Q = ydiffna*Area_Effective_Neutralaxis;
% disp('Q which is the first Moment of the solid area above the Neutral Axis in mm^3 is ');
% disp(Q);

%% Section 3.5 - Polar Moment of Inertia w.r.t. Neutral Axis

Ona = sqrt((Xbar^2)+(Ybar^2));
Ina = sqrt(((Xbar-x_inner_circle)^2)+((Ybar-y_inner_circle)^2));
Jo = pi*(D^4)/32;
Ji = (pi*(d^4)/32)+((pi*(d^2)/4)*e1^2);
J = Jo-Ji;
% disp('The polar moment of inertia for an angular hollow shaft (for the corresponding eccentricity) along the Centre of the
Outer Diameter at the point of interest in mm^4 is');
% disp(J);
Jon = (pi*(D^4)/32)+((pi*(D^2)/4)*Ona^2);
Jin = (pi*(d^4)/32)+((pi*(d^2)/4)*Ina^2);
Jn = Jon - Jin;
% disp('The polar moment of inertia for an angular hollow shaft (for the corresponding eccentricity) along the Neutal Axis
at the point of interest in mm^4 is');
% disp(Jn);

%% Section 3.6 - Second Moment of Inertia w.r.t. Neutral Axis

if (abs(xoffset) <= abs(yoffset))
```

```
        Ixx_o = (pi*(D^4)/64)+((pi*(D^2)/4)*(Ona)^2);
        Ixx_i = (pi*(d^4)/64)+((pi*(d^2)/4)*(Ina)^2);
        Ixx = Ixx_o - Ixx_i;
        % disp('Moment of Inertia at the Centroid of the entire shaft along XX Axis in mm^4 is ')
        % disp(Ixx);
        Iyy = Jn - Ixx;
        % disp('Moment of Inertia at the Centroid of the entire shaft along YY Axis in mm^4 is ')
        % disp(Iyy);
    else
        Iyy_o = (pi*(D^4)/64)+((pi*(D^2)/4)*(Ona)^2);
        Iyy_i = (pi*(d^4)/64)+((pi*(d^2)/4)*(Ina)^2);
        Iyy = Iyy_o - Iyy_i;
        Ixx = Jn - Iyy;
        % disp('Moment of Inertia at the Centroid of the entire shaft along XX Axis in mm^4 is ')
        % disp(Ixx);
        % disp('Moment of Inertia at the Centroid of the entire shaft along YY Axis in mm^4 is ')
        % disp(Iyy);
    end
    b = 2*(D-d);

    % disp('-----------------------------');
    % disp('-----------------------------');
    % disp('Neutral Axis Portion Over ');
    % disp('Now Individual Points along the Outer Diameter');

    %% Section 3.7 - Points - Inner Diameter First & Second Quadrant

    if (y1>= 0 && y1>yoffset && (y1-yoffset)>=r)                        % ID 1Q or 2Q      y1 1Q or 2Q
Completely Below
        % disp('Inner X bar above the Point is Not Intended to be used ');
        % disp('Inner Y bar above the Point is Not Intended to be used ');
        % disp('No Area for the inner circle above the Point is to be considered ');
        x_bar_point_inner_section = 0;
        y_bar_point_inner_section = 0;
        Area_point_inner_section = 0;
        ain = 0;
    else
        if(yoffset>=0 && y1>=0 && y1>=yoffset && (y1-yoffset)<r)                  % ID 1Q or 2Q      y1 1Q or
2Q    Segment Formation
            a1 = sqrt(((d)^2)-((2*(y1-yoffset))^2));
            ain = a1;
            thetaseg_point_inner = 2*(asind(a1/d));
            area_point_inner_segment = (r*r/2)*((pi*thetaseg_point_inner/180)-sind(thetaseg_point_inner));
            area_inner_full = pi*r*r;
            area_point_inner_remaining = area_inner_full - area_point_inner_segment;
            x_point_inner_seg = xoffset;
            x_point_inner_rem = xoffset;
            Aseg_point_inner =
(2/3)*r*r*r*sind(thetaseg_point_inner/2)*sind(thetaseg_point_inner/2)*sind(thetaseg_point_inner/2);
            Bseg_point_inner = area_point_inner_segment;
            y_point_inner_seg = yoffset+(Aseg_point_inner/Bseg_point_inner);
            y_point_inner_rem = ((y_inner_circle*area_inner_full)-
(y_point_inner_seg*area_point_inner_segment))/area_point_inner_remaining;
            x_bar_point_inner_section = x_point_inner_seg;
            y_bar_point_inner_section = y_point_inner_seg;
            Area_point_inner_section =  area_point_inner_segment;
        else
            if(yoffset>=0 && y1>=0 &&  yoffset>=y1 && (yoffset-y1)<r)                 % ID 1Q or 2Q      y1 1Q or
2Q    Remaining Formation
                a1 = sqrt(((d)^2)-((2*(y1-yoffset))^2));
                ain = a1;
                thetaseg_point_inner = 2*(asind(a1/d));
                area_point_inner_segment = (r*r/2)*((pi*thetaseg_point_inner/180)-sind(thetaseg_point_inner));
                area_point_inner_remaining = area_inner - area_point_inner_segment;
                x_point_inner_seg = xoffset;
                Aseg_point_inner =
(2/3)*r*r*r*sind(thetaseg_point_inner/2)*sind(thetaseg_point_inner/2)*sind(thetaseg_point_inner/2);
                Bseg_point_inner = area_point_inner_segment;
                y_point_inner_seg = yoffset-(Aseg_point_inner/Bseg_point_inner);
                x_point_inner_rem = xoffset;
```

```
                y_point_inner_rem = ((y_inner_circle*area_inner)-
(y_point_inner_seg*area_point_inner_segment))/area_point_inner_remaining;
                x_bar_point_inner_section = x_point_inner_rem;
                y_bar_point_inner_section = y_point_inner_rem;
                Area_point_inner_section =  area_point_inner_remaining;
            else
                if(yoffset>=0 && y1<=0 && yoffset>=y1 && (yoffset-y1)<r)                    % ID 1Q or 2Q       y1 3Q or
4Q     Remaining Formation
                    a1 = sqrt(((d)^2)-((2*(y1-yoffset))^2));
                    ain = a1;
                    thetaseg_point_inner = 2*(asind(a1/d));
                    area_point_inner_segment = (r*r/2)*((pi*thetaseg_point_inner/180)-sind(thetaseg_point_inner));
                    area_point_inner_remaining = area_inner - area_point_inner_segment;
                    x_point_inner_seg = xoffset;
                    Aseg_point_inner =
(2/3)*r*r*r*sind(thetaseg_point_inner/2)*sind(thetaseg_point_inner/2)*sind(thetaseg_point_inner/2);
                    Bseg_point_inner = area_point_inner_segment;
                    y_point_inner_seg = yoffset-(Aseg_point_inner/Bseg_point_inner);
                    x_point_inner_rem = xoffset;
                    y_point_inner_rem = ((y_inner_circle*area_inner)-
(y_point_inner_seg*area_point_inner_segment))/area_point_inner_remaining;
                    x_bar_point_inner_section = x_point_inner_rem;
                    y_bar_point_inner_section = y_point_inner_rem;
                    Area_point_inner_section =  area_point_inner_remaining;
                else
                    if(yoffset>=0 && y1<=0 && y1>=yoffset && (y1-yoffset)<r)                % ID 1Q or 2Q       y1 3Q or
4Q     Segment Formation
                        a1 = sqrt(((d)^2)-((2*(y1-yoffset))^2));
                        ain = a1;
                        thetaseg_point_inner = 2*(asind(a1/d));
                        area_point_inner_segment = (r*r/2)*((pi*thetaseg_point_inner/180)-sind(thetaseg_point_inner));
                        area_inner_full = pi*r*r;
                        area_point_inner_remaining = area_inner_full - area_point_inner_segment;
                        x_point_inner_seg = xoffset;
                        x_point_inner_rem = xoffset;
                        Aseg_point_inner =
(2/3)*r*r*r*sind(thetaseg_point_inner/2)*sind(thetaseg_point_inner/2)*sind(thetaseg_point_inner/2);
                        Bseg_point_inner = area_point_inner_segment;
                        y_point_inner_seg = yoffset+(Aseg_point_inner/Bseg_point_inner);
                        y_point_inner_rem = ((y_inner_circle*area_inner_full)-
(y_point_inner_seg*area_point_inner_segment))/area_point_inner_remaining;
                        x_bar_point_inner_section = x_point_inner_seg;
                        y_bar_point_inner_section = y_point_inner_seg;
                        Area_point_inner_section =  area_point_inner_segment;
                    else
                        if(y1>0 && yoffset>y1 && (yoffset-y1)>=r)                    % ID 1Q or 2Q       y1 1Q or 2Q
Completely Above
                            x_bar_point_inner_section = xoffset;
                            y_bar_point_inner_section = yoffset;
                            area_inner = pi*r*r;
                            Area_point_inner_section = area_inner;
                            ain = a1;
                        else
                            % disp('The inner diameter point is not in the first and second quadrant ');    % Execute Next
                        end
                    end
                end
            end
        end
    end

    %% Section 3.8 - Points - Inner Diameter Third & Fourth Quadrant

    if ((y1<= 0 && y1<yoffset && (y1-yoffset)<=-r))                              % ID 3Q or 4Q       y1 3Q or 4Q
Completely Above
        x_bar_point_inner_section = xoffset;
        y_bar_point_inner_section = yoffset;
        area_inner = pi*r*r;
        Area_point_inner_section = area_inner;
        ain = 0;
```

```
       else
          if(yoffset<=0 && y1<=0 && y1<=yoffset && (y1-yoffset)>-r)                                   % ID 3Q or 4Q       y1 3Q or
4Q     Remaining Formation
             a2 = sqrt(((d)^2)-((2*(y1-yoffset))^2));
             ain = a2;
             thetaseg_point_inner = 2*(asind(a2/d));
             area_point_inner_segment = (r*r/2)*((pi*thetaseg_point_inner/180)-sind(thetaseg_point_inner));
             area_inner_full = pi*r*r;
             area_point_inner_remaining = area_inner_full - area_point_inner_segment;
             x_point_inner_seg = xoffset;
             x_point_inner_rem = xoffset;
             Aseg_point_inner =
(2/3)*r*r*r*sind(thetaseg_point_inner/2)*sind(thetaseg_point_inner/2)*sind(thetaseg_point_inner/2);
             Bseg_point_inner = area_point_inner_segment;
             y_point_inner_seg = yoffset-(Aseg_point_inner/Bseg_point_inner);
             y_point_inner_rem = ((y_inner_circle*area_inner_full)-
(y_point_inner_seg*area_point_inner_segment))/area_point_inner_remaining;
             x_bar_point_inner_section = x_point_inner_rem;
             y_bar_point_inner_section = y_point_inner_rem;
             Area_point_inner_section =  area_point_inner_remaining;
          else
             if(yoffset<=0 && y1<=0 && yoffset<=y1 && (yoffset-y1)>-r)                                % ID 3Q or 4Q       y1 3Q or
4Q     Segment Formation
                a2 = sqrt(((d)^2)-((2*(y1-yoffset))^2));
                ain = a2;
                thetaseg_point_inner = 2*(asind(a2/d));
                area_point_inner_segment = (r*r/2)*((pi*thetaseg_point_inner/180)-sind(thetaseg_point_inner));
                area_point_inner_remaining = area_inner - area_point_inner_segment;
                x_point_inner_seg = xoffset;
                Aseg_point_inner =
(2/3)*r*r*r*sind(thetaseg_point_inner/2)*sind(thetaseg_point_inner/2)*sind(thetaseg_point_inner/2);
                Bseg_point_inner = area_point_inner_segment;
                y_point_inner_seg = yoffset+(Aseg_point_inner/Bseg_point_inner);
                x_point_inner_rem = xoffset;
                y_point_inner_rem = ((y_inner_circle*area_inner)-
(y_point_inner_seg*area_point_inner_segment))/area_point_inner_remaining;
                x_bar_point_inner_section = x_point_inner_seg;
                y_bar_point_inner_section = y_point_inner_seg;
                Area_point_inner_section =  area_point_inner_segment;
             else
                if(yoffset<=0 && y1>=0 && yoffset<=y1 && (yoffset-y1)>-r)                             % ID 3Q or 4Q       y1 1Q or
2Q     Segment Formation
                   a2 = sqrt(((d)^2)-((2*(y1-yoffset))^2));
                   ain = a2;
                   thetaseg_point_inner = 2*(asind(a2/d));
                   area_point_inner_segment = (r*r/2)*((pi*thetaseg_point_inner/180)-sind(thetaseg_point_inner));
                   area_point_inner_remaining = area_inner - area_point_inner_segment;
                   x_point_inner_seg = xoffset;
                   Aseg_point_inner =
(2/3)*r*r*r*sind(thetaseg_point_inner/2)*sind(thetaseg_point_inner/2)*sind(thetaseg_point_inner/2);
                   Bseg_point_inner = area_point_inner_segment;
                   y_point_inner_seg = yoffset+(Aseg_point_inner/Bseg_point_inner);
                   x_point_inner_rem = xoffset;
                   y_point_inner_rem = ((y_inner_circle*area_inner)-
(y_point_inner_seg*area_point_inner_segment))/area_point_inner_remaining;
                   x_bar_point_inner_section = x_point_inner_seg;
                   y_bar_point_inner_section = y_point_inner_seg;
                   Area_point_inner_section =  area_point_inner_segment;
                else
                   if(yoffset<=0 && y1>=0 && y1<=yoffset && (y1-yoffset)>-r)                          % ID 3Q or 4Q       y1 1Q or
2Q     Remaining Formation
                      a2 = sqrt(((d)^2)-((2*(y1-yoffset))^2));
                      ain = a2;
                      thetaseg_point_inner = 2*(asind(a2/d));
                      area_point_inner_segment = (r*r/2)*((pi*thetaseg_point_inner/180)-sind(thetaseg_point_inner));
                      area_inner_full = pi*r*r;
                      area_point_inner_remaining = area_inner_full - area_point_inner_segment;
                      x_point_inner_seg = xoffset;
                      x_point_inner_rem = xoffset;
```

```
                Aseg_point_inner =
(2/3)*r*r*r*sind(thetaseg_point_inner/2)*sind(thetaseg_point_inner/2)*sind(thetaseg_point_inner/2);
                Bseg_point_inner = area_point_inner_segment;
                y_point_inner_seg = yoffset-(Aseg_point_inner/Bseg_point_inner);
                y_point_inner_rem = ((y_inner_circle*area_inner_full)-
(y_point_inner_seg*area_point_inner_segment))/area_point_inner_remaining;
                x_bar_point_inner_section = x_point_inner_rem;
                y_bar_point_inner_section = y_point_inner_rem;
                Area_point_inner_section =  area_point_inner_remaining;
              else
                if(y1<=0 && yoffset<y1 && (yoffset-y1)<=-r)                % ID 3Q or 4Q      y1 3Q or 4Q
Completely below
                  % disp('Inner X bar above the Point is Not Intended to be used ');
                  % disp('Inner Y bar above the Point is Not Intended to be used ');
                  % disp('No Area for the inner circle above the Point is to be considered ');
                  x_bar_point_inner_section = 0;
                  y_bar_point_inner_section = 0;
                  Area_point_inner_section = 0;
                  ain = a2;
                else
                  % disp('The inner diameter point is not in the third and fourth quadrant and its a CATASTROPHE ');       %
Catastrophe
                  ain = 0;
                  x_bar_point_inner_section = 0;
                  y_bar_point_inner_section = 0;
                  Area_point_inner_section = 0;
                end
              end
            end
          end
        end
      end

      % disp ('Area of the Inner Portion above the Point in mm is ');
      % disp (Area_point_inner_section);
      % disp ('X bar of the Inner Portion above the Point in mm is ');
      % disp (x_bar_point_inner_section );
      % disp ('Y bar of the Innner Portion above the Point in mm is ');
      % disp (y_bar_point_inner_section );

      %% Section 3.9 - Points - Outer Diameter

      A1 = sqrt(((D)^2)-((2*y1)^2));
      thetaseg_point_outer = 2*(asind(A1/D));
      area_point_outer_segment = (R*R/2)*((pi*thetaseg_point_outer/180)-sind(thetaseg_point_outer));
      area_point_outer_remaining = area_outer - area_point_outer_segment;
      x_point_outer_seg = 0;
      x_point_outer_rem = 0;
      Aseg_point_outer = (2/3)*R*R*R*sind(thetaseg_point_outer/2)*sind(thetaseg_point_outer/2)*sind(thetaseg_point_outer/2);
      Bseg_point_outer = area_point_outer_segment;
      y_point_outer_seg = (Aseg_point_outer/Bseg_point_outer);
      y_point_outer_rem = ((y_outer_circle*area_outer)-
(y_point_outer_seg*area_point_outer_segment))/area_point_outer_remaining;
      if (y1>0 && y1>=R)                                            % y1 lies on the farthest point in
Positive Y axis
        Area_point_outer_section = 0;
        x_bar_point_outer_section = 0;
        y_bar_point_outer_section = 0;
      else
        if (y1<0 && y1<=-R)                                          % y1 lies on the farthest point in
Negative Y axis
          Area_point_outer_section = area_point_outer_segment;
          x_bar_point_outer_section = 0;
          y_bar_point_outer_section = y_bar_outer_section;
        else
          if (0<y1<R)                                                % y1 3Q or 4Q    Remaining
Formation
            Area_point_outer_section = area_point_outer_remaining;
            x_bar_point_outer_section = 0;
            y_bar_point_outer_section =  -y_point_outer_rem;
```

```
            else
               if (y1<0<R)
                  Area_point_outer_section = area_point_outer_segment;                                     % y1 1Q or 2Q
Segment Formation
                  x_bar_point_outer_section = 0;
                  y_bar_point_outer_section = y_point_outer_seg;
               else
                  Area_point_outer_section = area_outer/2;                                            % y1 lies on the Horizontal
Axis
                  x_bar_point_outer_section = 0;
                  y_bar_point_outer_section = 4*D/(3*2*pi);
               end
            end
         end
      end

      % % disp ('Area of the Outer Portion above the Point in mm is ');
      % disp (Area_point_outer_section);
      % disp ('X bar of the Outer Portion above the Point in mm is ');
      % disp (x_bar_point_outer_section );
      % disp ('Y bar of the Outer Portion above the Point in mm is ');
      % disp (y_bar_point_outer_section );

      %% Section 3.10 - First Moment of Inertia w.r.t. Points

      Area_aboveNA_point_effective = (Area_point_outer_section-Area_point_inner_section);
      % disp('Effective Area above the Horizontal Axis along the point in mm2 is ');
      % disp(Area_aboveNA_point_effective);
      if (Area_aboveNA_point_effective~=0)
         y_centroid_point = ((y_bar_point_outer_section * Area_point_outer_section)-(y_bar_point_inner_section *
Area_point_inner_section))/Area_aboveNA_point_effective;
      else
         y_centroid_point =0;
      end
      % disp('Effective Centroid for the hollow portion above the horizontal axis about the point in mm ');
      % disp(y_centroid_point);
      ydiff = abs(Ybar - y_centroid_point);
      Qp = (ydiff) * Area_aboveNA_point_effective;
      % disp('Effective First Moment of Area for the hollow portion above the horizontal axis about the point in mm3 ');
      % disp(Qp);
      y_bendingload = Ybar-y1;
      % disp('Vertical Distance (to be used in bending load) of the point from the neutral axis is ');
      % disp(y_bendingload);
      % disp('First Moment of the solid area above the Neutral Axis in mm^3 is ');
      % disp(Q);
      % disp('First Moment of Area above the horizontal axis about the point in mm3 ');
      % disp(Qp);
      % disp('Moment of Inertia at the Centroid of the entire shaft along YY Axis in mm^4 is ')
      % disp(Iyy);
      % disp('Polar Moment of Inertia along the Neutal Axis in mm^4 is');
      % disp(Jn);

      %% Section 4 - Parameters
      Mz = W*l;
      if (y1>Ybar)
         yver = (y1-Ybar);              % Bending Load acts on A or above YBar, resulting in Tensile Stress, hence yver>0
      else
         if (y1<Ybar)
            yver = -(Ybar-y1);        % Bending Load acts on B or below YBar, resulting in Compressive Stress, hence yver<0
         else
            yver = 0;                % On the Centroidal Neutral Axis, yver is zero and hence My/I is also zero, hence no normal
bending stresses
         end
      end
      Izz = Iyy;
      Qzz = Qp;
      aou = abs(2*R*sind((180-(2*theta))/2));
      th =(aou - ain);
      % disp('Thickness at the point of interest in mm is ');
      % disp(th);
```

```
%% Section 5 - Stress and Strain Computations

sigma_axial = P/Areaentire;

indicesref = 1/thetasplit_t;
indiv_angles = (indicesref*theta)+1;
indiv_angle = fix(indiv_angles);
tau_torsion = tautorsion(indiv_angle);

sigma_bending_z = -Mz*yver/Izz;

if(th~=0)
   tau_bending = -(W*Qzz)/(th*Izz);
else
   tau_bending = 0;
end
sigmaYa(thetaidx,lidx) = 0;

if (x1>Xbar && y1>Ybar)
   sigmaXa(thetaidx,lidx) = -sigma_bending_z;
   tauXYa(thetaidx,lidx)  = tau_bending-tau_torsion;
else
   if (x1<Xbar && y1>Ybar)
      sigmaXa(thetaidx,lidx) = -sigma_bending_z;
      tauXYa(thetaidx,lidx)  = tau_bending + tau_torsion;
   else
      if(x1<Xbar && y1<Ybar)
         sigmaXa(thetaidx,lidx) = sigma_bending_z;
         tauXYa(thetaidx,lidx)  = -tau_bending-tau_torsion;
      else
         if(x1>Xbar && y1<Ybar)
            sigmaXa(thetaidx,lidx) = sigma_bending_z;
            tauXYa(thetaidx,lidx)  = -tau_bending+tau_torsion;

         else
            if(x1==Xbar && y1<Ybar)
                  sigmaXa(thetaidx,lidx) = sigma_bending_z;
                  tauXYa(thetaidx,lidx)  = -tau_torsion;
            else
               if(x1==Xbar && y1>Ybar)
                  sigmaXa(thetaidx,lidx) = -sigma_bending_z;
                  tauXYa(thetaidx,lidx)  = tau_torsion;
               else
                  if(y1==Ybar && x1<Xbar)
                     sigmaXa(thetaidx,lidx) = 0;
                     tauXYa(thetaidx,lidx)  = tau_bending + tau_torsion;
                  else
                     if(y1==Ybar && x1>Xbar)
                        sigmaXa(thetaidx,lidx) = 0;
                        tauXYa(thetaidx,lidx)  = tau_bending-tau_torsion;
                     end
                  end
               end
            end
         end
      end
   end
end

Mz_total(thetaidx,lidx) = Mz;
yvertotal(thetaidx,lidx) = yver;
W_total(thetaidx,lidx)=W;
Qzz_total(thetaidx,lidx)=abs(Qzz);
th_total(thetaidx,lidx)=th;
Izz_total(thetaidx,lidx)=Izz;

sigma_phi = -0.3*(sigma_bending_z+sigma_axial);
strain_zz(thetaidx,lidx)= (sigma_bending_z+sigma_axial)*1000000/E;
strain_tt(thetaidx,lidx) = sigma_phi*1000000/E;
```

```
    end
end

strain_45Max = ((strain_zz + strain_tt)/2)+(gamma_strain/2);
strain_ua1 = ((strain_zz + strain_tt)/2)-(((strain_zz - strain_tt)/2)*cosd(2*userang1))+((gamma_strain/2)*sind(2*userang1));
strain_ua2 = ((strain_zz + strain_tt)/2)-(((strain_zz - strain_tt)/2)*cosd(2*userang2))+((gamma_strain/2)*sind(2*userang2));

%% Section 6 - Plots

figure(1);
plot(thetaindividual_tor,tautor);
set(gca,'XTick',0:30:359 );
xlim([0 359]);
xlabel('Angle in degrees ')
ylabel('Shear Stress due to Torsion in MPa ')
title('Angle vs Shear Stress due to Torsion alone ');
grid on;

figure(2);
plot(thetavals,strain_zz);
title('Theta vs Strain zz at 45 degrees ');
xlabel('Angle in degrees ');
ylabel('Strain zz values in uE ');
xlim([0 359]);
set(gca,'XTick',0:30:359 );
grid on

figure(3);
plot(thetavals,strain_tt);
title('Theta vs Strain tt at 45 degrees ');
xlabel('Angle in degrees ');
ylabel('Strain tt values in uE ');
xlim([0 359]);
set(gca,'XTick',0:30:359 );
grid on

figure(4);
plot(thetavals,strain_45Max);
title('Theta vs Strain at 45 degrees ');
xlabel('Angle in degrees ');
ylabel('Strain values in uE ');
xlim([0 359]);
set(gca,'XTick',0:30:359 );
grid on

figure(5);
plot(thetavals,strain_ua1,thetavals,strain_ua2);
title('Theta vs Strain at userangle comparisons degrees ');
xlabel('Angle in degrees ');
ylabel('Strain values at 45 and userangles 1 and 2 in uE ');
xlim([0 359]);
set(gca,'XTick',0:30:359 );
grid on

%% Section 7 - Region Strain

option = menu('Choose the Number of Regions for Strain Gauges',...
    '0 Region ',...
    '1 Region ',...
    '2 Regions ',...
    '3 Regions ',...
    '4 Regions ',...
    '8 Regions');

switch option
    case 1
        nooftimes = 0;

    case 2
        nooftimes = 1:1;
```

```
        sheetx = 1;

    case 3
        nooftimes = 1:2;
        sheetx = 2;

    case 4
        nooftimes = 1:3;
        sheetx = 3;

    case 5
        nooftimes = 1:4;
        sheetx = 4;

    case 6
        nooftimes = 1:8;
        sheetx = 5;
end

if nooftimes ~= 0
    for trialdx = 1:length(nooftimes);
        trial = nooftimes(trialdx);

        filename = 'Datainput.xlsx';
        sheet = trialdx;
        xlRange = 'E32';
        subsetA = xlsread(filename,sheet,xlRange);
        x1u = -subsetA;

        filename = 'Datainput.xlsx';
        sheet = trialdx;
        xlRange = 'E33';
        subsetA = xlsread(filename,sheet,xlRange);
        sheet = sheetx;
        y1u = -subsetA;

        filename = 'Datainput.xlsx';
        sheet = trialdx;
        xlRange = 'E34';
        subsetA = xlsread(filename,sheet,xlRange);
        x2u = -subsetA;

        filename = 'Datainput.xlsx';
        sheet = trialdx;
        xlRange = 'E35';
        subsetA = xlsread(filename,sheet,xlRange);
        y2u = -subsetA;

        filename = 'Datainput.xlsx';
        sheet = trialdx;
        xlRange = 'E36';
        subsetA = xlsread(filename,sheet,xlRange);
        x3u = -subsetA;

        filename = 'Datainput.xlsx';
        sheet = trialdx;
        xlRange = 'E37';
        subsetA = xlsread(filename,sheet,xlRange);
        y3u = -subsetA;

        filename = 'Datainput.xlsx';
        sheet = trialdx;
        xlRange = 'E38';
        subsetA = xlsread(filename,sheet,xlRange);
        x4u = -subsetA;

        filename = 'Datainput.xlsx';
        sheet = trialdx;
        xlRange = 'E39';
        subsetA = xlsread(filename,sheet,xlRange);
```

```
    y4u = -subsetA;

    % inters = input('Enter 0 if the region intersect or pass through 0 degree in the first quadrant (Else enter any positive value)');

    xvals = 0:-ldiff:-L;
    yvals = 0:-thetasplit_t:-359;

    if (y1u < y4u || y2u < y3u )
        inters = 0;
    else
        inters = 1;
    end

    if (inters==~0)
        m1_u=(y2u-y1u)/(x2u-x1u);
        if (abs(m1_u) == Inf)
            m1u = 0;
        else
            m1u = m1_u;
        end
        c1u=y1u-(m1u*x1u);
        m2_u=(x3u-x2u)/(y3u-y2u);
        if (abs(m2_u) == Inf)
            m2u = 0;
        else
            m2u = m2_u;
        end
        c2u=x2u-(m2u*y2u);
        m3_u=(y4u-y3u)/(x4u-x3u);
        if (abs(m3_u) == Inf)
            m3u = 0;
        else
            m3u = m3_u;
        end
        c3u=y3u-(m3u*x3u);
        m4_u=(x1u-x4u)/(y1u-y4u);
        if (abs(m4_u) == Inf)
            m4u = 0;
        else
            m4u = m4_u;
        end
        c4u=x4u-(m4u*y4u);

        for xdx = 1:length(xvals);
            x = xvals(xdx);
            for ydx = 1:length(yvals);
                y = yvals(ydx);
                d1(ydx,xdx) = y-(m1u*x);
                d2(ydx,xdx) = x-(m2u*y);
                d3(ydx,xdx) = y-(m3u*x);
                d4(ydx,xdx) = x-(m4u*y);
                if(d1(ydx,xdx)<=c1u && d2(ydx,xdx)<=c2u && d3(ydx,xdx)>=c3u && d4(ydx,xdx)>=c4u);
                    Apoints(ydx,xdx)=1;
                else
                    Apoints(ydx,xdx)=0;
                end
            end
        end

        Region1 = strain_ua1.*Apoints;
        Region2 = strain_ua2.*Apoints;
        Region1avg(trialdx) = mean(nonzeros(Region1));
        Region2avg(trialdx) = mean(nonzeros(Region2));
        disp('Region Average in uE is ');
        disp(Region1avg(trialdx));
        disp(Region2avg(trialdx));
    else
        x1_u = x4u;
        y1_u = y4u;
```

```
            x2_u = x3u;
            y2_u = y3u;
            x3_u = x2u;
            y3_u = y2u;
            x4_u = x1u;
            y4_u = y1u;
            m1_u=(y2_u-y1_u)/(x2_u-x1_u);
            if (abs(m1_u) == Inf)
                m1u = 0;
            else
                m1u = m1_u;
            end
            c1u=y1_u-(m1u*x1_u);
            m2_u=(x3_u-x2_u)/(y3_u-y2_u);
            if (abs(m2_u) == Inf)
                m2u = 0;
            else
                m2u = m2_u;
            end
            c2u=x2_u-(m2u*y2_u);
            m3_u=(y4_u-y3_u)/(x4_u-x3_u);
            if (abs(m3_u) == Inf)
                m3u = 0;
            else
                m3u = m3_u;
            end
            c3u=y3_u-(m3u*x3_u);
            m4_u=(x1_u-x4_u)/(y1_u-y4_u);
            if (abs(m4_u) == Inf)
                m4u = 0;
            else
                m4u = m4_u;
            end
            c4u=x4_u-(m4u*y4_u);

            for xdx = 1:length(xvals);
                x = xvals(xdx);
                for ydx = 1:length(yvals);
                    y = yvals(ydx);
                    d1(ydx,xdx) = y-(m1u*x);
                    d2(ydx,xdx) = x-(m2u*y);
                    d3(ydx,xdx) = y-(m3u*x);
                    d4(ydx,xdx) = x-(m4u*y);
                    if(d1(ydx,xdx)<c1u && d2(ydx,xdx)<=c2u && d3(ydx,xdx)>c3u && d4(ydx,xdx)>=c4u);
                        Apoints1(ydx,xdx)=0;
                    else
                        Apoints1(ydx,xdx)=1;
                    end
                    if(d2(ydx,xdx)<=c2u && d4(ydx,xdx)>=c4u);
                        Apoints2(ydx,xdx)=1;
                    else
                        Apoints2(ydx,xdx)=0;
                    end
                    Apoints3 = Apoints1.*Apoints2;
                end
            end
            Region1 = strain_ua1.*Apoints3;
            Region2 = strain_ua2.*Apoints3;
            Region1avg(trialdx) = mean(nonzeros(Region1));
            Region2avg(trialdx) = mean(nonzeros(Region2));
            disp('Region Average in uE is ');
            disp(Region1avg(trialdx));
            disp(Region2avg(trialdx));
        end
    end
else
    return
end

%% Section 8 - Sensitivity
```

```
filename = 'Datainput.xlsx';
sheet = 1;
xlRange = 'E21';
subsetA = xlsread(filename,sheet,xlRange);
GF = subsetA;

filename = 'Datainput.xlsx';
sheet = 1;
xlRange = 'E22';
subsetA = xlsread(filename,sheet,xlRange);
Rg = subsetA;

filename = 'Datainput.xlsx';
sheet = 1;
xlRange = 'E23';
subsetA = xlsread(filename,sheet,xlRange);
V_wb = subsetA;

Del1R = Region1avg*GF*Rg*(10^-6)*1000;
Del2R = Region2avg*GF*Rg*(10^-6)*1000;

if (sheetx == 4)
    e1t = Region1avg(1);
    e1c = Region2avg(1);
    e2t = Region1avg(2);
    e2c = Region2avg(2);
    e3t = Region1avg(3);
    e3c = Region2avg(3);
    e4t = Region1avg(4);
    e4c = Region2avg(4);
end
% e1wb = input('Enter the Strain computed for Strain Gauge 1 (or type e1t for tension / e1c for compression - if 4 SG are
present) ');
% e2wb = input('Enter the Strain computed for Strain Gauge 2 (or type e2t for tension / e2c for compression) ');
% e3wb = input('Enter the Strain computed for Strain Gauge 3 (or type e3t for tension / e3c for compression) ');
% e4wb = input('Enter the Strain computed for Strain Gauge 4 (or type e4t for tension / e4c for compression) ');

Sens = (GF/(4*1000))*(e1t-e2c+e3t-e4c);
disp('Sensitivity in mV/V is ');
disp(Sens);
disp('We have reached the end of the program');
```

APPENDIX 3 - Excel sheet – Datainput.xlsx

Note: If the user is interested in running the Matlab script, then

- *Let an Excel sheet (with name Datainput) similar to the one below be created.*
- *The text values are to be entered accordingly (Text in similar cell identity).*
- *The user can change the parameters in column E and column M.*
- *The values will be read as an input by the Matlab Script.*
- *Let the content between cells 30 and 40 be copied (First Strain Gauge Region 1) and pasted in Sheets 2, Sheets 3 and Sheets 4 of the excel sheet in the same cell range (Cells 30 and 40).*
- *In sheet 2, the Cell value in C30-31 is to be named as Second Strain Gauge – Region .2*
- *In sheet 3, the Cell value in C30-31 is to be named as Third Strain Gauge – Region 2.*
- *In sheet 4, the Cell value in C30-31 is to be named as Fourth Strain Gauge – Region 2.*
- *Suitable values for the region coordinates in E32 to E39 of Sheets 1, 2, 3 and 4 are to be entered.*
- *When watched from load end and when Bending load acts on Top most point of the shaft*
  - *Sheet 1 corresponds to a strain gauge placed in left or top left location,*
  - *Sheet 2 corresponds to a strain gauge placed on top or upper right location,*
  - *Sheet 3 corresponds to a strain gauge placed on right or lower right location and*
  - *Sheet 4 corresponds to a strain gauge placed on bottom or lower left location.*
- *The Excel file should be saved in the same working directory as that of the script before the script is run in Matlab!*

MATLAB DATA INPUT - SHEET 1 TO SHEET 8

Input parameters for a shaft	*Diameter and Geometrical Parameters*		
	Outer Diameter of the shaft	12	mm
	Inner Diameter of the shaft	11,2	mm
	Total Length of the shaft	30	mm
	Offset in x direction	0,01	mm
	Offset in y direction	0,01	mm
	Inclination of the inner diameter axis at centrum	0	degrees
	Load Parameters		
	Torque	2100	Nmm
	Bending Load at the Load End	2,5	N
	Axial Load	5	N
	Material Properties and Grid Size		
	Young's Modulus	68947,6	MPa
	Poisson's Ratio	0,3000000	(no unit)
	Length Resolution	1	mm
	Angle Resolution	1	degrees
	Bridge Conncetion Details		
	Gauge Factor	2	(no unit)
	Resistance in the grid at no load	1000	ohm
	Supply Voltage for the Wheatstone Bridge	5	V
	Mounting Angle		
	Mounting Angle corresponding to S.G. 1	45	degrees
	Mounting Angle corresponding to S.G. 2	-45	degrees
	**Please don't forget to save this document after changes*		

210000	or	68947,6
Steel		*Aluminium*

If interested in Axis inclination, then	
Angle inclination	0,3
Which length to be seen	0
Corresponding ey is	0
Enter the above ey value in Cell E9 and length value in E7	

First Strain Gauge - Region 1

SG 1 on Left or Upper Left	Length Coordinate of Point A	26,7	mm
	Theta Coordinate of Point A	171,4	degrees
	Length Coordinate of Point B	23,3	mm
	Theta Coordinate of Point B	171,4	degrees
	Length Coordinate of Point C	23,3	mm
	Theta Coordinate of Point C	188,6	degrees
	Length Coordinate of Point D	26,7	mm
	Theta Coordinate of Point D	188,6	degrees

**Note*	** Length Coordinate is from the Load end*
	** Theta Coordinate is from the X axis shown below*